Sparse estimation of high-dimensional covariance matrices

by

Adam J. Rothman

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Statistics)
in The University of Michigan
2010

Doctoral Committee:

Associate Professor Elizaveta Levina, Co-Chair
Associate Professor Ji Zhu, Co-Chair
Associate Professor Bin Nan
Associate Professor Kerby A. Shedden

To my father

ACKNOWLEDGEMENTS

I am forever grateful to my thesis advisors Liza Levina and Ji Zhu for everything they have done to help me with this thesis and with my development as a researcher. Words cannot describe the incredible impact they have made on my life. I also thank my doctoral committee members Kerby Shedden and Bin Nan. I am grateful to my family, who have been very supportive, and I am especially grateful to my father Ed Rothman, for his outstanding guidance and encouragement. I also thank my fiance Dawn Zoch for her support.

Chapter II is joint work with Peter J. Bickel (University of California, Berkeley).

TABLE OF CONTENTS

DEDICATION ii

ACKNOWLEDGEMENTS iii

LIST OF FIGURES vi

LIST OF TABLES viii

CHAPTER

I. Introduction 1

II. Sparse permutation invariant covariance estimation 6

2.1 Introduction 6
2.2 Analysis of the SPICE method 7
2.3 The Cholesky-based SPICE algorithm 16
2.3.1 Algorithm convergence 19
2.3.2 Computational complexity 20
2.3.3 Choice of tuning parameter 21
2.4 Numerical Results 22
2.4.1 Simulations 22
2.4.2 Colon tumor classification example 26
2.5 Discussion 28
2.6 Derivation of the SPICE Algorithm 29

III. Generalized thresholding of large covariance matrices 31

3.1 Introduction 31
3.2 Examples of generalized thresholding 32
3.3 Consistency and sparsity of generalized thresholding 35
3.4 Simulation results 41
3.4.1 Simulation settings 41

3.4.2 Performance Evaluation 42
3.4.3 Summary of results 43
3.5 Example: Gene clustering via correlations 49

IV. A new approach to Cholesky-based covariance regularization in high dimensions 53

4.1 Introduction 53
4.2 Modified Cholesky decomposition of the covariance matrix . . 54
4.3 Regularized estimation of the Cholesky factor L 56
4.3.1 Banding the Cholesky factor 56
4.3.2 Connection to constrained maximum likelihood . . . 57
4.3.3 The penalized regression approach 61
4.4 Numerical results 62
4.4.1 Simulation Settings 62
4.4.2 Results 64
4.5 Sonar data example 68
4.6 Discussion 70

V. Sparse multivariate regression with covariance estimation . . 72

5.1 Introduction 72
5.2 Joint estimation of B and Ω via penalized normal likelihood . 76
5.2.1 The MRCE method 76
5.2.2 Computational algorithms 78
5.3 Simulation study 81
5.3.1 Estimators 81
5.3.2 Models 81
5.3.3 Performance evaluation 83
5.3.4 Results 83
5.4 Example: Predicting Asset Returns 86
5.5 Summary and discussion 89
5.6 Derivation of Algorithm 1 91

BIBLIOGRAPHY 93

LIST OF FIGURES

Figure

2.1 Computing time in seconds vs p (log-log scale) for SPICE and glasso 21

2.2 Heatmaps of zeros identified in the concentration matrix out of 50 replications. White color is 50/50 zeros identified, black is 0/50. . . 25

3.1 Generalized thresholding functions for $\lambda = 1$, $a = 3.7$, $\eta = 1$. 35

3.2 TPR vs. FPR for Model 2. The points correspond to 50 different realizations, with each method selecting its own threshold using validation data. The solid line is obtained by varying the threshold over the whole range (all methods have the same TPR and FPR for a fixed threshold). 45

3.3 TPR vs. FPR for Model 3. The points correspond to 50 different realizations, with each method selecting its own threshold using validation data. The solid line is obtained by varying the threshold over the whole range (all methods have the same value of TPR and FPR for a fixed threshold). 47

3.4 Average $K(q)$ versus q with $p = 200$. 48

3.5 (a) Heatmap of the absolute values of sample correlations of the top 40 genes; (b) Heatmap of the gene expression data, with rows (genes) sorted by hierarchical clustering and columns sorted by tissue class. 50

3.6 Heatmaps of the absolute values of estimated correlations. The 40 genes with the largest F-statistic are marked with stars. The genes are ordered by hierarchical clustering using estimated correlations. The percentage of off-diagonal elements estimated as zero is given in parentheses for each method. 51

4.1 Scree plots for the sample covariance (gray dashes), Ledoit–Wolf (dots), banding the sample covariance (dash-dot), Cholesky banding (black dashes), and the truth (solid) for $p = 1000$, averaged over 200 replications. 67

4.2 Heatmaps of the absolute values of entries in the correlation matrix estimates, where a correlation of magnitude 0 is white and a correlation of magnitude 1 is black. The top row is for metal spectra and the bottom row is for rock spectra. 68

4.3 Scree plots of the sample covariance (solid), sample banding (dots), and Cholesky banding (dashes) for the metal spectra in panel (a) and for the rock spectra in panel (b). 69

LIST OF TABLES

Table

2.1 Simulations: Average (SE) Kullback-Leibler loss over 50 replications. 24

2.2 Percentage of correctly estimated non-zeros (TP %) and correctly estimated zeros (TN %) in the concentration matrix (average and SE over 50 replications) for SPICE. 26

2.3 Averages and SEs of classification errors in % over 100 splits. Tuning parameter for SPICE chosen by (A): 5-fold CV on the training data maximizing the likelihood; (B): 5-fold CV on the training data minimizing the classification error; (C): minimizing the classification error on the test data. 28

3.1 Average(SE) operator norm loss for Model 1. 44

3.2 Average(SE) operator norm loss and true and false positive rates for Model 2. 44

3.3 Average(SE) operator norm loss and true and false positive rates for Model 3 ($k = p/2$). 46

4.1 Averages and standard errors of the operator norm loss for the sample covariance, Ledoit–Wolf's estimator, the banded sample covariance, and regularization of Cholesky factor of the covariance by banding, lasso, and nested lasso. 65

4.2 Averages and standard errors of true positive/true negative percentages for Σ_2, based on 200 replications. 66

4.3 Percentage of positive definite banded sample realizations 68

5.1 Model error for the AR(1) error covariance models of low dimension. Averages and standard errors in parenthesis are based on 50 replications with $n = 50$. Tuning parameters were selected using a 10^x resolution. 84

5.2 Model error for the AR(1) error covariance models of high dimension. Averages and standard errors in parenthesis are based on 50 replications with $n = 50$. Tuning parameters were selected using a 10^x resolution. 85

5.3 Model error for the FGN error covariance models of low dimension. Averages and standard errors in parenthesis are based on 50 replications with $n = 50$. Tuning parameters were selected using a 10^x resolution. 85

5.4 Model error for the FGN error covariance models of high dimension. Averages and standard errors in parenthesis are based on 50 replications with $n = 50$. Tuning parameters were selected using a 10^x resolution. 86

5.5 True Positive Rate / True Negative Rate for the AR(1) error covariance models, averaged over 50 replications; $n = 50$. Standard errors are omitted, the largest standard error is 0.04 and most are less than 0.01. Tuning parameters were selected using a 10^x resolution. . . . 87

5.6 True Positive Rate / True Negative Rate for the FGN error covariance models averaged over 50 replications; $n = 50$. Standard errors are omitted, the largest standard error is 0.04 and most are less than 0.01. Tuning parameters were selected using a 10^x resolution. . . . 87

5.7 Average testing squared error for each output (company) $\times 1000$, based on 26 testing points. Standard errors are reported in parenthesis. The results for the FES method where copied from Table 3 in Yuan et al. (2007). 89

5.8 Estimated coefficient matrix B for approximate MRCE. Results are rounded to the nearest tenth, and coefficients that are exactly zero are denoted by "0". 90

5.9 Signs of the inverse error covariance estimate for MRCE 90

CHAPTER I

Introduction

Estimation of large covariance matrices, particularly in situations where the data dimension p is comparable to or larger than the sample size n, has attracted a lot of attention recently. The abundance of high-dimensional data is one reason for the interest in the problem: gene arrays, fMRI, various kinds of spectroscopy, climate studies, and many other applications often generate very high dimensions and moderate sample sizes. Another reason is the ubiquity of the covariance matrix in data analysis tools. Principal component analysis (PCA), linear and quadratic discriminant analysis (LDA and QDA), inference about the means of the components, and analysis of independence and conditional independence in graphical models all require an estimate of the covariance matrix or its inverse, also known as the precision or concentration matrix. Finally, recent advances in random matrix theory – see Johnstone (2001) for a review, and also Paul (2007) – allowed in-depth theoretical studies of the traditional estimator, the sample (empirical) covariance matrix, and showed that without regularization the sample covariance performs poorly in high dimensions. These results helped stimulate research on alternative estimators in high dimensions.

The existing literature on covariance estimation can be loosely divided into two categories. One large class of methods covers the situation where variables have a natural ordering or there is a notion of distance between variables, as in longitudinal

data, time series, spatial data, or spectroscopy. There are, however, many applications where an ordering of the variables is not available, such as genetics, social, financial and economic data. Methods that are invariant to variable permutations (like the covariance matrix itself) are necessary in such applications.

Naturally ordered variables

A large class of covariance estimation methods covers the situation where variables have a natural ordering. The implicit regularizing assumption underlying these methods is that variables far apart in the ordering have small correlations (or partial correlations, if the object of regularization is the concentration matrix), and estimators that take advantage of this have been proposed by Wu and Pourahmadi (2003), Bickel and Levina (2004), Huang et al. (2006), Furrer and Bengtsson (2007), Bickel and Levina (2008b), Levina et al. (2008), and others. When the inverse of the covariance matrix is the primary goal and the variables are ordered, regularization is usually introduced via the modified Cholesky decomposition,

$$\Sigma^{-1} = L^T D^{-1} L.$$

Here L is a lower triangular matrix with $l_{jj} = 1$ and $l_{jj'} = -\phi_{jj'}$, where $\phi_{jj'}$, $j' < j$ is the coefficient of $X_{j'}$ in the population regression of X_j on $X_1, \ldots, X_{j-1}$, and D is a diagonal matrix with residual variances of these regressions on the diagonal. Several approaches to regularizing the Cholesky factor L have been proposed, mostly based on its regression interpretation. A k-banded estimator of L can be obtained by regressing each variable only on its closest k predecessors; Wu and Pourahmadi (2003) proposed this estimator and an estimation approach involving nonparametric methods for smoothing the sub-diagonals of L where they chose k via an AIC penalty. Bickel and Levina (2008b) showed that banding the Cholesky factor produces a consistent

estimator in the operator norm under weak conditions on the covariance matrix, and proposed a cross-validation scheme for picking k. Huang et al. (2006) proposed adding either an l_2 (ridge) or an l_1 (lasso) penalty on the elements of L to the normal likelihood. The lasso penalty creates zeros in L in arbitrary locations, which is more flexible than banding, but (unlike in the case of banding) the resulting estimate of the inverse may not have any zeros at all. Levina et al. (2008) proposed adaptive banding, which, by using a nested lasso penalty, allows a different k for each regression, and hence is more flexible than banding while also retaining some sparsity in the inverse. Bayesian approaches to the problem introduce zeros via priors, either in the Cholesky factor (Smith and Kohn, 2002) or in the inverse itself (Wong et al., 2003).

In Chapter IV (Rothman et al., 2010), we propose a new regression interpretation of the Cholesky factor of the covariance matrix, as opposed to this well known regression interpretation of the Cholesky factor of the inverse covariance, which leads to a new class of regularized covariance estimators suitable for high-dimensional problems.

Unordered variables

There are many applications where an ordering of the variables is not available. Some early work of Dempster (1972) proposed setting elements in the concentration matrix to zero as a means for regularization; however, this work did not address positive definiteness nor models with many variables. Regularizing large covariance matrices by Steinian shrinkage of eigenvalues has been proposed early on (Haff, 1980; Dey and Srinivasan, 1985). More recently, Ledoit and Wolf (2003) proposed a way to compute an optimal linear combination of the sample covariance with the identity matrix, which also results in shrinkage of eigenvalues. Shrinkage estimators are invariant to variable permutations but they do not affect the eigenvectors of the covariance, only the eigenvalues, and it has been shown that the sample eigenvectors are also not consistent when p is large (Johnstone and Lu, 2004). Shrinking eigenvalues also does

not create sparsity in any sense. Sometimes alternative estimators are available in the context of a specific application – e.g., for a factor analysis model with known factors Fan et al. (2008a) develop regularized estimators for both the covariance and its inverse.

Sparse concentration matrices are widely studied in the graphical models literature, since zero partial correlations imply no edge exists between vertices in an undirected graph structure. The classical graphical models approach, however, is different from covariance estimation, since it normally focuses on just finding the zeros. For example, Drton and Perlman (2008) develop a multiple testing procedure for simultaneously testing hypotheses of zeros in the concentration matrix. There are also more algorithmic approaches to finding zeros in the concentration matrix, such as running a lasso regression of each variable on all the other variables (Meinshausen and Bühlmann, 2006), or the PC-algorithm (Kalisch and Bühlmann, 2007). Both have been shown to be consistent in high-dimensional settings, but none of these methods supply an estimator of the covariance matrix. In principle, once the zeros are found, a constrained maximum likelihood estimator of the covariance can be computed (Chaudhuri et al., 2007), but it is not clear what the properties of such a two-step procedure would be.

Two recent papers, d'Aspremont et al. (2008) and Yuan and Lin (2007), take a penalized likelihood approach by applying an l_1 penalty to the entries of the concentration matrix. This results in a permutation-invariant loss function that tends to produce a sparse estimate of the inverse. Yuan and Lin (2007) used the max-det algorithm to compute the estimator, which limited their numerical results to values of $p \leq 10$, and derived a fixed p, large n convergence result. d'Aspremont et al. (2008) proposed a much faster semi-definite programming algorithm based on Nesterov's method for interior point optimization. A new very fast algorithm for the same problem was proposed by Friedman et al. (2008), which is based on the coordi-

nate descent algorithm for the lasso (Friedman et al., 2007). In Chapter II (Rothman et al., 2008), we offer the first large p asymptotic analysis of this estimator and offer a fast algorithm to compute this estimator. Our work has since been extended to more general penalties on the concentration matrix by Lam and Fan (2009) and Fan et al. (2009).

A simple alternative to penalized likelihood is thresholding the sample covariance matrix, which has been analyzed by Bickel and Levina (2008a) and El Karoui (2008). Thresholding carries essentially no computational burden, except for cross-validation for the tuning parameter (which is also necessary for penalized likelihood) and is thus an attractive option for problems in very high dimensions and real-time applications. However, in regression and wavelet shrinkage contexts (see, e.g., Donoho et al. (1995), Fan and Li (2001)), hard thresholding tends to do worse than more flexible estimators that combine thresholding with shrinkage, for example, soft thresholding or SCAD (Fan and Li, 2001). The estimates resulting from such shrinkage typically are continuous functions of the "naive" estimates, a desirable feature not shared by hard thresholding. We introduce a new class of generalized thresholding operators and offer consistency and sparsity analysis in Chapter III (Rothman et al., 2009).

Estimating the covariance matrix or its inverse is usually a means to an end and not the ultimate goal. We may ultimately be interested in prediction or classification but need an estimate of the covariance matrix or its inverse along the way. In Chapter V, we propose a procedure for constructing a sparse estimator of a multivariate regression coefficient matrix that accounts for correlation of the response variables. This method, which we call multivariate regression with covariance estimation (MRCE), involves penalized likelihood with simultaneous estimation of the regression coefficients and the covariance structure.

CHAPTER II

Sparse permutation invariant covariance estimation

2.1 Introduction

This chapter proposes a method for constructing a sparse estimator for the inverse covariance (concentration) matrix in high-dimensional settings. We call this estimator SPICE, an acronym standing for sparse permutation invariant covariance estimator. SPICE is formed by adding a lasso-type penalty to the negative normal log-likelihood. The lasso-type penalty encourages sparsity in the concentration matrix primarily because the convex penalty function is non-differentiable at points (matrices) where elements in the off-diagonal of the concentration matrix are exactly equal to zero. Aside from having a favorable convergence rate for sparse models as we will show, the SPICE estimator also yields a pattern of zeros and non-zeros in the concentration matrix implying an undirected graph structure.

We first establish a rate of convergence in the Frobenius norm as both data dimension p and sample size n are allowed to grow, and show that the rate depends explicitly on how sparse the true concentration matrix is. We also show that a correlation-based version of the method exhibits better rates in the operator norm. We illustrate these theoretical results with simulation examples.

We additionally derive a fast iterative algorithm for computing the estimator and

argue that it converges to the unique global minimizer of the convex problem. The algorithm relies on the popular Cholesky decomposition of the inverse covariance matrix but produces a permutation-invariant estimator. The method is compared to other estimators on simulated data and on a real data example of tumor tissue classification using gene expression data.

This chapter is organized as follows: Section 2.2 summarizes the SPICE approach in general, and presents consistency results. The Cholesky-based computational algorithm, along with a discussion of optimization issues, is presented in Section 2.3. Section 2.4 presents numerical results for SPICE and a number of other methods, for simulated data and a real example on classification of colon tumors using gene expression data. Section 2.5 concludes with discussion.

2.2 Analysis of the SPICE method

We assume throughout that we observe $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$, i.i.d. p-variate normal random variables with mean $\mathbf{0}$ and covariance matrix Σ_0, and write $\boldsymbol{X}_i = (X_{i1}, \ldots, X_{ip})^T$. Let $\Sigma_0 = [\sigma_{0ij}]$, and $\Omega_0 = \Sigma_0^{-1}$ be the inverse of the true covariance matrix. For any matrix $M = [m_{ij}]$, we write $|M|$ for the determinant of M, $\mathrm{tr}(M)$ for the trace of M, and $\varphi_{\max}(M)$ and $\varphi_{\min}(M)$ for the largest and smallest eigenvalues, respectively. We write $M^+ = \mathrm{diag}(M)$ for a diagonal matrix with the same diagonal as M, and $M^- = M - M^+$. In the asymptotic analysis, we will use the Frobenius matrix norm $\|M\|_F^2 = \sum_{i,j} m_{ij}^2$, and the operator norm (also known as matrix 2-norm), $\|M\|^2 = \varphi_{\max}(MM^T)$. We will also write $|\cdot|_1$ for the l_1 norm of a vector or matrix vectorized, i.e., for a matrix $|M|_1 = \sum_{i,j} |m_{ij}|$.

It is easy to see that under the normal assumption the negative log-likelihood, up to a constant, can be written in terms of the concentration matrix as

$$\ell(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n; \Omega) = \mathrm{tr}(\Omega\hat{\Sigma}) - \log|\Omega|,$$

where

$$\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{X}_i - \bar{\boldsymbol{X}}\right)\left(\boldsymbol{X}_i - \bar{\boldsymbol{X}}\right)^T$$

is the sample covariance matrix.

We define the SPICE estimator $\hat{\Omega}_\lambda$ of the inverse covariance matrix as the minimizer of the penalized negative log-likelihood,

$$\hat{\Omega}_\lambda = \arg\min_{\Omega \succ 0}\left\{\mathrm{tr}(\Omega\hat{\Sigma}) - \log|\Omega| + \lambda|\Omega^-|_1\right\} \tag{2.1}$$

where λ is a non-negative tuning parameter, and the minimization is taken over symmetric positive definite matrices.

SPICE is identical to the lasso-type estimator proposed by Yuan and Lin (2007), and very similar to the estimator of d'Aspremont et al. (2008) (they used $|\Omega|_1$ rather than $|\Omega^-|_1$ in the penalty). The loss function is invariant to permutations of variables and should encourage sparsity in $\hat{\Omega}$ due to the l_1 penalty applied to its off-diagonal elements.

We make the following assumptions about the true model:

A1: Let the set $S = \{(i,j) : \Omega_{0ij} \neq 0,\ i \neq j\}$. Then $\mathrm{card}(S) \leq s$.

A2: $\varphi_{\min}(\Sigma_0) \geq \underline{k} > 0$, or equivalently $\varphi_{\max}(\Omega_0) \leq 1/\underline{k}$.

A3: $\varphi_{\max}(\Sigma_0) \leq \overline{k}$.

Note that assumption A2 guarantees that Ω_0 exists. Assumption A1 is more of a definition, since it does not stipulate anything about s ($s = p(p-1)/2$ would give a full matrix).

Theorem II.1. *Let $\hat{\Omega}_\lambda$ be the minimizer defined by* (2.1)*. Under A1, A2, A3, if* $\lambda \asymp \sqrt{\frac{\log p}{n}}$,

$$\|\hat{\Omega}_\lambda - \Omega_0\|_F = O_P\left(\sqrt{\frac{(p+s)\log p}{n}}\right)\ . \tag{2.2}$$

The theorem can be restated, more suggestively, as

$$\frac{\|\hat{\Omega}_\lambda - \Omega_0\|_F^2}{p} = O_P\left(\left(1 + \frac{s}{p}\right)\frac{\log p}{n}\right) \ . \tag{2.3}$$

The reason for the second formulation (2.3) is the relation of the Frobenius norm to the operator norm, $\|M\|_F^2/p \leq \|M\|^2 \leq \|M\|_F^2$.

Before proceeding with the proof of Theorem II.1, we discuss a modification to SPICE based on using the correlation matrix. An inspection of the proof reveals that the worst part of the rate, $\sqrt{p \log p/n}$, comes from estimating the diagonal. This suggests that if we were to use the correlation matrix rather than the covariance matrix, we should be able to get the rate of $\sqrt{s \log p/n}$. Indeed, let $\Sigma_0 = W\Gamma W$, where Γ is the true correlation matrix, and W is the diagonal matrix of true standard deviations. Let $\hat{W}$ and $\hat{\Gamma}$ be the sample estimates of W and Γ, i.e., $\hat{W}^2 = \hat{\Sigma}^+$, $\hat{\Gamma} = \hat{W}^{-1}\hat{\Sigma}\hat{W}^{-1}$. Let $K = \Gamma^{-1}$. Define a SPICE estimate of K by

$$\hat{K}_\lambda = \arg\min_{\Omega \succ 0} \left\{\mathrm{tr}(\Omega\hat{\Gamma}) - \log|\Omega| + \lambda|\Omega^-|_1\right\} \tag{2.4}$$

Then we can define a modified correlation-based estimator of the concentration matrix by

$$\tilde{\Omega}_\lambda = \hat{W}^{-1}\hat{K}_\lambda\hat{W}^{-1}. \tag{2.5}$$

It turns out that in the Frobenius norm $\tilde{\Omega}$ has the same rate as $\hat{\Omega}$, but for $\tilde{\Omega}$ we can get a convergence rate in the operator norm (matrix 2-norm). As discussed previously by Bickel and Levina (2008b), El Karoui (2008) and others, the operator norm is more appropriate than the Frobenius norm for spectral analysis, e.g., PCA. It also allows for a direct comparison with banding rates obtained in Bickel and Levina (2008b) and thresholding rates in Bickel and Levina (2008a).

Theorem II.2. *Under assumptions of Theorem II.1,*

$$\|\tilde{\Omega}_\lambda - \Omega_0\| = O_P\left(\sqrt{\frac{(s+1)\log p}{n}}\right) .$$

Note. This rate is very similar to the rate for thresholding the covariance matrix obtained by Bickel and Levina (2008a). They showed that under the assumption $\max_i \sum_j |\sigma_{ij}|^q \leq c_0(p)$ for $0 \leq q < 1$, if the sample covariance entries are set to 0 when their absolute values fall below the threshold $\lambda = M\sqrt{\frac{\log p}{n}}$, then the resulting estimator converges to the truth in operator norm at the rate no worse than $O_P\Big(c_0(p)\big(\frac{\log p}{n}\big)^{(1-q)/2}\Big)$. Since the truly sparse case corresponds to $q = 0$, and $c_0(p)$ is a bound on the number of non-zero elements in each row, and thus $\sqrt{s} \asymp c_0(p)$, this rate coincides with ours, even though the estimator and the method of proof are very different. However, Lemma II.3 below is the basis of the proof in both cases, and ultimately it is the bound (2.6) that gives rise to the same rate. A similar rate has been obtained for banding the covariance matrix in Bickel and Levina (2008b), under an additional assumption that depends on the ordering of the variables and is not applicable here (see Bickel and Levina (2008a) for a comparison between banding and thresholding rates).

In the proof, we will need a lemma of Bickel and Levina (2008b) (Lemma 3) which is based on a large deviation result of Saulis and Statulevičius (1991). We state the result here for completeness.

Lemma II.3. *Let Z_i be i.i.d. $\mathcal{N}(\mathbf{0}, \Sigma_p)$ and $\varphi_{\max}(\Sigma_p) \leq \overline{k} < \infty$. Then, if $\Sigma_p = [\sigma_{ab}]$,*

$$P\big[|\sum_{i=1}^{n}(Z_{ij}Z_{ik} - \sigma_{jk})| \geq n\nu\big] \leq c_1 \exp(-c_2 n\nu^2) \quad \textit{for } |\nu| \leq \delta \tag{2.6}$$

where c_1, c_2 and δ depend on $\overline{k}$ only.

Proof of Theorem II.1.

Let

$$\begin{aligned} Q(\Omega) =& \mathrm{tr}(\Omega\hat{\Sigma}) - \log|\Omega| + \lambda|\Omega^-|_1 - \mathrm{tr}(\Omega_0\hat{\Sigma}) + \log|\Omega_0| - \lambda|\Omega_0^-|_1 \\ =& \mathrm{tr}\big[(\Omega-\Omega_0)(\hat{\Sigma}-\Sigma_0)\big] - (\log|\Omega| - \log|\Omega_0|) \\ &+ \mathrm{tr}\big[(\Omega-\Omega_0)\Sigma_0\big] + \lambda(|\Omega^-|_1 - |\Omega_0^-|_1) \end{aligned} \tag{2.7}$$

Our estimate $\hat{\Omega}$ minimizes $Q(\Omega)$, or equivalently $\hat{\Delta} = \hat{\Omega} - \Omega_0$ minimizes $G(\Delta) \equiv Q(\Omega_0 + \Delta)$. Note that we suppress the dependence on λ in $\hat{\Omega}$ and $\hat{\Delta}$.

The main idea of the proof is as follows. Consider the set

$$\Theta_n(M) = \{\Delta : \Delta = \Delta^T \, , \|\Delta\|_F = Mr_n\},$$

where

$$r_n = \sqrt{\frac{(p+s)\log p}{n}} \to 0 \; .$$

Note that $G(\Delta) = Q(\Omega_0 + \Delta)$ is a convex function, and

$$G(\hat{\Delta}) \leq G(0) = 0 \; .$$

Then, if we can show that

$$\inf\{G(\Delta) : \; \Delta \in \Theta_n(M)\} > 0 \; ,$$

the minimizer $\hat{\Delta}$ must be inside the sphere defined by $\Theta_n(M)$, and hence

$$\|\hat{\Delta}\|_F \leq Mr_n \; . \tag{2.8}$$

For the logarithm term in (2.7), doing the Taylor expansion of $f(t) = \log|\Omega + t\Delta|$ and

using the integral form of the remainder and the symmetry of Δ, Σ_0, and Ω_0 gives

$$\log|\Omega_0+\Delta|-\log|\Omega_0| = \mathrm{tr}(\Sigma_0\Delta)-\tilde{\Delta}^T\Big[\int_0^1 (1-v)(\Omega_0+v\Delta)^{-1}\otimes(\Omega_0+v\Delta)^{-1}dv\Big]\tilde{\Delta} \quad (2.9)$$

where $\otimes$ is the Kronecker product (if $A = [a_{ij}]_{p_1\times q_1}$, $B = [b_{kl}]_{p_2\times q_2}$, then $A\otimes B = [a_{ij}b_{kl}]_{p_1p_2\times q_1q_2}$), and $\tilde{\Delta}$ is Δ vectorized to match the dimensions of the Kronecker product.

Therefore, we may write (2.7) as,

$$\begin{aligned} G(\Delta) =& \mathrm{tr}\big(\Delta(\hat{\Sigma}-\Sigma_0)\big) + \tilde{\Delta}^T\Big[\int_0^1 (1-v)(\Omega_0+v\Delta)^{-1}\otimes(\Omega_0+v\Delta)^{-1}dv\Big]\tilde{\Delta} \\ &+ \lambda(|\Omega_0^- + \Delta^-|_1 - |\Omega_0^-|_1) \end{aligned} \quad (2.10)$$

For an index set A and a matrix $M = [m_{ij}]$, write $M_A \equiv [m_{ij}I((i,j)\in A)]$, where $I(\cdot)$ is an indicator function. Recall $S = \{(i,j):\ \Omega_{0ij}\neq 0,\ i\neq j\}$ and let $\overline{S}$ be its complement. Note that $|\Omega_0^- + \Delta^-|_1 = |\Omega_{0S}^- + \Delta_S^-|_1 + |\Delta_{\overline{S}}^-|_1$, and $|\Omega_0^-|_1 = |\Omega_{0S}^-|_1$. Then the triangular inequality implies

$$\lambda\big(|\Omega_0^- + \Delta^-|_1 - |\Omega_0^-|_1\big) \geq \lambda\big(|\Delta_{\overline{S}}^-|_1 - |\Delta_S^-|_1\big)\ . \quad (2.11)$$

Now, using symmetry again, we write

$$|\mathrm{tr}\big(\Delta(\hat{\Sigma}-\Sigma_0)\big)| \leq \Big|\sum_{i\neq j}(\hat{\sigma}_{ij}-\sigma_{0ij})\Delta_{ij}\Big| + \Big|\sum_i(\hat{\sigma}_{ii}-\sigma_{0ii})\Delta_{ii}\Big| = \mathrm{I} + \mathrm{II}. \quad (2.12)$$

To bound term I, note that the union sum inequality and Lemma II.3 imply that, with probability tending to 1,

$$\max_{i\neq j}|\hat{\sigma}_{ij}-\sigma_{0ij}| \leq C_1\sqrt{\frac{\log p}{n}}$$

and hence term I is bounded by

$$\mathrm{I} \leq C_1 \sqrt{\frac{\log p}{n}} |\Delta^-|_1. \tag{2.13}$$

The second bound comes from the Cauchy-Schwartz inequality and Lemma II.3:

$$\begin{aligned} \mathrm{II} &\leq \left[\sum_{i=1}^{p} \left(\hat{\sigma}_{ii} - \sigma_{0ii} \right)^2 \right]^{1/2} \|\Delta^+\|_F \leq \sqrt{p} \max_{1 \leq i \leq p} |\hat{\sigma}_{ii} - \sigma_{0ii}| \, \|\Delta^+\|_F \\ &\leq C_2 \sqrt{\frac{p \log p}{n}} \|\Delta^+\|_F \leq C_2 \sqrt{\frac{(p+s) \log p}{n}} \|\Delta^+\|_F, \end{aligned} \tag{2.14}$$

also with probability tending to 1.

Now, take

$$\lambda = \frac{C_1}{\varepsilon} \sqrt{\frac{\log p}{n}} \ . \tag{2.15}$$

By (2.10),

$$\begin{aligned} G(\Delta) &\geq \frac{1}{4} \underline{k}^2 \|\Delta\|_F^2 - C_1 \sqrt{\frac{\log p}{n}} |\Delta^-|_1 - C_2 \sqrt{\frac{(p+s) \log p}{n}} \|\Delta^+\|_F + \lambda \left(|\Delta^-_{\overline{S}}|_1 - |\Delta^-_S|_1 \right) \\ &= \frac{1}{4} \underline{k}^2 \|\Delta\|_F^2 - C_1 \sqrt{\frac{\log p}{n}} \left(1 - \frac{1}{\varepsilon} \right) |\Delta^-_{\overline{S}}|_1 - C_1 \sqrt{\frac{\log p}{n}} \left(1 + \frac{1}{\varepsilon} \right) |\Delta^-_S|_1 \\ &\quad - C_2 \sqrt{\frac{(p+s) \log p}{n}} \|\Delta^+\|_F \end{aligned} \tag{2.16}$$

The first term comes from a bound on the integral which we will argue separately below. The second term is always positive, and hence we may omit it for the lower bound. Now, note that

$$|\Delta^-_S|_1 \leq \sqrt{s} \|\Delta^-_S\|_F \leq \sqrt{s} \|\Delta^-\|_F \leq \sqrt{p+s} \|\Delta^-\|_F \ .$$

Thus we have

$$\begin{aligned} G(\Delta) &\geq \|\Delta^-\|_F^2 \left[\frac{1}{4}\underline{k}^2 - C_1\sqrt{\frac{(p+s)\log p}{n}}\left(1+\frac{1}{\varepsilon}\right)\|\Delta^-\|_F^{-1}\right] \\ &\quad + \|\Delta^+\|_F^2 \left[\frac{1}{4}\underline{k}^2 - C_2\sqrt{\frac{(p+s)\log p}{n}}\|\Delta^+\|_F^{-1}\right] \\ &= \|\Delta^-\|_F^2 \left[\frac{1}{4}\underline{k}^2 - \frac{C_1(1+\varepsilon)}{\varepsilon M}\right] + \|\Delta^+\|_F^2 \left[\frac{1}{4}\underline{k}^2 - \frac{C_2}{M}\right] > 0 \end{aligned} \tag{2.17}$$

for M sufficiently large.

It only remains to check the bound on the integral term in (2.10). Recall that $\varphi_{\min}(M) = \min_{\|x\|=1} x^T M x$. After factoring out the norm of $\tilde{\Delta}$, we have, for $\Delta \in \Theta_n(M)$,

$$\begin{aligned} &\varphi_{\min}\Big(\int_0^1 (1-v)(\Omega_0+v\Delta)^{-1}\otimes(\Omega_0+v\Delta)^{-1}dv\Big) \\ &\quad \geq \int_0^1 (1-v)\varphi_{\min}^2(\Omega_0+v\Delta)^{-1}dv \geq \frac{1}{2}\min_{0\leq v\leq 1}\varphi_{\min}^2(\Omega_0+v\Delta)^{-1} \\ &\quad \geq \frac{1}{2}\min\left\{\varphi_{\min}^2(\Omega_0+\Delta)^{-1} :\ \|\Delta\|_F \leq Mr_n\right\}. \end{aligned}$$

The first inequality uses the fact that the eigenvalues of Kronecker products of symmetric matrices are the products of the eigenvalues of their factors. Now

$$\varphi_{\min}^2(\Omega_0+\Delta)^{-1} = \varphi_{\max}^{-2}(\Omega_0+\Delta) \geq (\|\Omega_0\|+\|\Delta\|)^{-2} \geq \frac{1}{2}\underline{k}^2 \tag{2.18}$$

with probability tending to 1, since $\|\Delta\| \leq \|\Delta\|_F = o(1)$. This establishes the theorem. □

As noted above, an inspection of the proof shows that $\sqrt{p\log p/n}$ in the rate comes from estimating the diagonal. If we focus on the correlation matrix estimate $\hat{K}_\lambda$ in (2.4) instead, we can immediately obtain

Corollary 1. Under assumptions of Theorem II.1,

$$\|\hat{K}_\lambda - K\|_F = O_P\left(\sqrt{\frac{s\log p}{n}}\right) .$$

Now we can use Corollary 1 to prove Theorem II.2, the operator norm bound.

Proof of Theorem II.2. Write

$$\begin{aligned}\|\tilde{\Omega}_\lambda - \Omega_0\| &= \|\hat{W}^{-1}\hat{K}_\lambda\hat{W}^{-1} - W^{-1}KW^{-1}\| \\ &\le \|\hat{W}^{-1} - W^{-1}\|\,\|\hat{K}_\lambda - K\|\,\|\hat{W}^{-1} - W^{-1}\| \\ &+ \|\hat{W}^{-1} - W^{-1}\|(\|\hat{K}_\lambda\|\,\|W^{-1}\| + \|\hat{W}^{-1}\|\,\|K\|) \\ &+ \|\hat{K}_\lambda - K\|\,\|\hat{W}^{-1}\|\,\|W^{-1}\|\end{aligned}$$

where we are using the sub-multiplicative norm property $\|AB\| \le \|A\|\,\|B\|$ (see, e.g., Golub and Van Loan (1989)). Now, $\|W^{-1}\|$ and $\|K\|$ are $O(1)$ by assumptions A2 and A3. Lemma II.3 implies that

$$\|\hat{W}^2 - W^2\| = O_P\left(\sqrt{\frac{\log p}{n}}\right), \tag{2.19}$$

and since $\|\hat{W}^{-1} - W^{-1}\| \overset{P}{\asymp} \|\hat{W}^2 - W^2\|$ (where by $A \overset{P}{\asymp} B$ we mean $A = O_P(B)$ and $B = O_P(A)$), we have the rate of $\sqrt{\log p/n}$ for $\|\hat{W}^{-1} - W^{-1}\|$. This together with Corollary 1 in turn implies that $\|\hat{W}^{-1}\|$ and $\|\hat{K}_\lambda\|$ are $O_P(1)$, and the theorem follows. □

Note that in the Frobenius norm, we only have $\|\hat{W}^2 - W^2\| = O_P(\sqrt{p\log p/n})$, and thus the Frobenius rate of $\tilde{\Omega}_\lambda$ is the same as that of $\hat{\Omega}_\lambda$.

2.3 The Cholesky-based SPICE algorithm

In this section, we develop an iterative algorithm for computing the SPICE estimator using the Cholesky decomposition; however, unlike other estimators that depend on the Cholesky decomposition, we minimize a permutation invariant objective function, and thus the estimator remains permutation invariant. We use the quadratic approximation to the absolute value, a standard tool in optimization which has been previously used in the statistics literature to handle lasso-type penalties, for example, by Fan and Li (2001) and Huang et al. (2006). In this our algorithm differs from the glasso algorithm of Friedman et al. (2008), which is based on a lasso algorithm and works directly on the absolute values. Both algorithms have computation complexity of $O(p^3)$, but we acquire another small constant factor (on the order of 10) due to the additional iterations required for the quadratic approximation to converge (see more on this in Section 2.4). However, using the quadratic approximation allows us to write down the algorithm explicitly in general terms for an l_q penalty $|w_{ij}|^q$ with $q \geq 1$, rather than only for $q = 1$. In particular, our algorithm is equally applicable for use with a ridge penalty ($q = 2$), although in that special case it simplifies even further, or with a bridge penalty ($1 < q < 2$) proposed by Fu (1998), which may work better for certain classes of covariances. It can also be used with SCAD (Fan and Li, 2001) or other more complicated non-convex penalties that are typically approximated by the local quadratic approximation. Even though we derive the algorithm with a general q, in this chapter we only present results for $q = 1$.

Our goal is to minimize the objective function,

$$f(\Omega) = \operatorname{tr}(\Omega\hat{\Sigma}) - \log|\Omega| + \lambda \sum_{j' \neq j} |\omega_{j'j}|^q, \tag{2.20}$$

where $q = 1$ corresponds to the computation of $\hat{\Omega}_\lambda$ in (2.1). For $q \geq 1$, the objective function is convex in the elements of Ω and has a global minimum $\hat{\Omega}$. Our strategy

is to re-parametrize the objective (2.20) using the Cholesky decomposition of Ω to enforce automatic positive definiteness. Rather than using the modified Cholesky decomposition with its regression interpretation, as has been standard in the literature, we simply write

$$\Omega = T^T T,$$

where $T = [t_{ij}]$ is a lower triangular matrix. We can still use the regression interpretation if needed, by writing

$$\begin{aligned} t_{jj'} &= -\frac{\phi_{jj'}}{\sqrt{d_{jj}}}, \quad j' < j \\ t_{jj} &= \frac{1}{\sqrt{d_{jj}}}, \end{aligned} \tag{2.21}$$

where $\phi_{jj'}$ is the coefficient of $X_{j'}$ in the regression of X_j on $X_1, \ldots, X_{j-1}$, and d_{jj} is the corresponding residual variance.

To minimize f in terms of T, we apply a cyclical coordinate descent approach and minimize f with respect to one element of T at a time. Further, we use a quadratic approximation to f, which allows to find the minimum of the univariate functions of each parameter in closed form. The algorithm is iterated until convergence. Here we outline the main steps of the algorithm, and leave the full derivation in Section 2.6.

In a slight abuse of notation, we write X for the $n \times p$ data matrix where each column has already been centered by its sample mean. The three terms in (2.20) can be expressed as a function of T as follows:

$$\operatorname{tr}(\Omega\hat{\Sigma}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{p}\big(\sum_{k=1}^{j} t_{jk}X_{ik}\big)^2 \tag{2.22}$$

$$\log|\Omega| = 2\sum_{j=1}^{p}\log t_{jj} \tag{2.23}$$

$$\sum_{j'\neq j}|\omega_{j'j}|^q = 2\sum_{j'>j}\big|\sum_{k=j'}^{p} t_{kj'}t_{kj}\big|^q \tag{2.24}$$

The quadratic approximation for $|u|^q$ is shown in (2.25). Since the algorithm is iterative, $u^{(k)}$ denotes the value of u from the previous iteration, and $u^{(k+1)}$ is the value at current iteration.

$$|u^{(k+1)}|^q \approx \frac{q}{2}\frac{(u^{(k+1)})^2}{|u^{(k)}|^{2-q}} + \left(1 - \frac{q}{2}\right)|u^{(k)}|^q \tag{2.25}$$

Hunter and Li (2005) suggest replacing $|u^{(k)}|$ in the denominator with $|u^{(k)}| + \epsilon$ to avoid division by zero, and refer to this as the ϵ-perturbed quadratic approximation. This quadratic approximation to f, which we denote $\tilde{f}_{\epsilon,k}$ at iteration k, allows us to easily take partial derivatives with respect to each parameter in T, and provides a closed form solution for the univariate minimizer for each coordinate.

The algorithm requires an initial value $\hat{T}^{(0)}$, which corresponds to $\hat{\Omega}^{(0)}$. If the sample covariance $\hat{\Sigma}$ is non-degenerate, which is generally the case for $p < n$, one could simply set $\hat{\Omega}^{(0)} = \hat{\Sigma}^{-1}$. More generally, we found the following simple strategy to work well: approximate $\phi_{jj'}$ in (2.21) by regressing X_j on $X_{j'}$ *alone*, for $j' = 1, \ldots, j-1$, and then compute $\hat{T}^{(0)}$ using (2.21). Yet another alternative is to start from the diagonal estimator.

The Algorithm:

Step 0. Initialize $\hat{T} = \hat{T}^{(0)}$ and $\hat{\Omega}^{(0)} = (\hat{T}^{(0)})^T\hat{T}^{(0)}$.

Step 1. For each parameter t_{lc}, $c = 1, \ldots, p, l = c, \ldots, p$, solve $\nabla_{t_{lc}}\tilde{f}_{\epsilon,k}(T) = 0$ to find new $\hat{t}_{lc}$.

Step 2. Repeat Step 1 until convergence of $\hat{T}$ and set $T^{(k+1)} = \hat{T}$.

Step 3. Set $\hat{\Omega}^{(k+1)} = (T^{(k+1)})^T T^{(k+1)}$ and repeat Steps 1-3 until convergence of $\hat{\Omega}$.

Steps 2 and 3 may seem redundant, but they are needed for two different reasons. Step 2 is needed because we only minimize with respect to one parameter at a time, holding all other parameters fixed; and Step 3 is needed because of the quadratic

approximation for $|u|^q$. After convergence, we replace entries in $\hat{\Omega}$ with smaller magnitude than ϵ with zero, using a fixed value of $\epsilon = 10^{-8}$. Another approach with virtually the same performance is to replace entries of $\hat{\Omega}^{(k)}$ with ϵ if their magnitude falls below ϵ in Step 3, and use (2.25) directly in the objective function in Step 1 instead of using $\tilde{f}_{\epsilon,k}$.

In practice, we found that working with the correlation matrix as described in Theorem II.2 is slightly better than working with the covariance matrix, although the differences are fairly small. Still, in all the numerical results we standardize the variables first and then rescale our estimate by the sample standard deviations of the variables.

2.3.1 Algorithm convergence

The convergence of the algorithm essentially follows from two standard results. For the inner loop cycling through individual parameters, the value of the objective function decreases at each iteration, and the objective function is differentiable everywhere. Thus the inner loop of the algorithm converges by a standard theorem on cyclical coordinate descent for smooth functions (see, e.g., Bazaraa et al. (2006), p. 367), to a stationary point $\nabla g(T) = 0$, where $g(T) = \tilde{f}_{\epsilon,k}(T^T T)$. The function $\tilde{f}_{\epsilon,k}$ is convex in the original parameters ω_{ij}, but since we reparametrized it in terms of T, the function g is not necessarily convex in T. In the next proposition we verify that this stationary point of g corresponds to the global minimum of the convex function $\tilde{f}_{\epsilon,k}$.

Proposition II.4. *Let $\tilde{f} \equiv \tilde{f}_{\epsilon,k}$ be the original convex function f approximated by the ϵ-perturbed local quadratic approximation at iteration k, let T be a $p \times p$ lower triangular matrix, and let $g(T) = \tilde{f}(T^T T)$. Let S_0 be the unique solution to $\nabla \tilde{f}(S) = 0$, and let T_0 be a solution to $\nabla g(T) = 0$. Then $S_0 = T_0^T T_0$.*

Proof of Proposition II.4. Let $h : T \to T^T T$. Note that h maps all of $\mathbb{R}^{p(p+1)/2}$

(all lower triangular matrices) into a convex subset of $\mathbb{R}^{p(p+1)/2}$ (non-negative definite symmetric matrices). Denote the differential of h in the direction $d \in \mathbb{R}^{p(p+1)/2}$ evaluated at $t_0 \in \mathbb{R}^{p(p+1)/2}$ by $\nabla h(t_0)[d]$. Then,

$$\nabla h(t_0)[d] = T_0^T D + D^T T_0 \ , \tag{2.26}$$

where T_0 and D are, respectively, t_0 and d written as $p \times p$ matrices. Now, using the chain rule and (2.26), we have

$$\nabla g(t_0)[d] = \nabla \tilde{f}(vec(T_0^T T_0)) \left(T_0^T D + D^T T_0\right) \ . \tag{2.27}$$

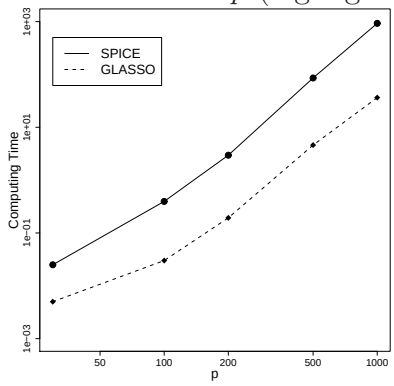
where we now think of $\tilde{f}$ as a function from $\mathbb{R}^{p(p+1)/2}$ to R. Since $\tilde{f}$ is convex and has a unique minimizer $s_0 = vec(S_0)$, $\nabla \tilde{f}(s)[d]$ vanishes iff $s = s_0$ or $d = 0$. Thus $\nabla g(t_0)[d] = 0$ vanishes iff $T_0^T T_0 = S_0$ or $T_0^T D + D^T T_0 = 0$, or $T_0^T D = -(T_0^T D)^T$. If any diagonal elements of T_0 are 0, then T_0 is singular, and so is $T_0^T T_0$, and thus $g(T_0) = \infty$, so a singular T_0 cannot be a stationary point of g. Since T_0 is lower triangular and all its diagonal elements must be non-zero, one can show by induction that $T_0^T D = -(T_0^T D)^T$ implies $D = 0$. □

For the outer loop iterating through the quadratic approximation, we can apply the argument of Hunter and Li (2005) for ϵ-perturbed local quadratic approximation obtained from general results for minorize-maximize algorithms, and conclude that as $k \to \infty$ and $\epsilon \to 0$ the algorithm converges to the global minimum of the original convex function f in (2.20). In practice, we have also observed that our algorithm and glasso converge to the same solution.

2.3.2 Computational complexity

The computational complexity of the algorithm in terms of p is $O(p^3)$, since each parameter update is at most $O(p)$ (see (2.32) in the Appendix), and there are $O(p^2)$

Figure 2.1: Computing time in seconds vs p (log-log scale) for SPICE and glasso



parameters. The only other algorithm for computing this estimator at the cost of $O(p^3)$ is glasso of Friedman et al. (2008); the algorithms of Yuan and Lin (2007) and d'Aspremont et al. (2008) have higher computational cost. For extensive timing comparisons of glasso and the algorithm of d'Aspremont et al. (2008), which showed convincingly that glasso is much faster, see Friedman et al. (2008). The exact timing also depends on the implementation, platform, etc (our algorithm is implemented in C and glasso in Fortran). Actual computing times we obtained for glasso and the SPICE algorithm are shown below in Figure 2.1, for model Ω_2 described in Section 2.4.1, with values of tuning parameters chosen as described in Section 2.3.3.

2.3.3 Choice of tuning parameter

Like any other penalty-based approach, SPICE requires selecting the tuning parameter λ. In simulations, we generate a separate validation dataset, and select λ by maximizing the normal likelihood on the validation data with $\hat{\Omega}_\lambda$ estimated from the training data. Alternatively, one can use 5-fold cross-validation, which we do for the real data analysis. There is some theoretical basis for selecting the tuning parameter in this way – see Bickel and Levina (2008a).

2.4 Numerical Results

In this section, we compare the performance of SPICE to the shrinkage estimator of Ledoit and Wolf (2003) and to the sample covariance matrix when applicable ($p < n$), using simulated and real data. We do not include any estimators that depend on variable ordering (such as banding of Bickel and Levina (2008b) or the Lasso penalty on the Cholesky factor of Huang et al. (2006)), nor estimators that focus on introducing sparsity in the covariance matrix itself rather than in its inverse (such as thresholding), as they would automatically be at a disadvantage on sparse concentration matrices. The Ledoit-Wolf estimator does not introduce sparsity in the inverse either, but we use it as a benchmark for cases when $p > n$, since the sample covariance is not invertible.

2.4.1 Simulations

In simulations, we focus on comparing performance on sparse concentration matrices, with varying levels of sparsity. We consider the following four covariance models.

1. Ω_1: AR(1), $\sigma_{j'j} = 0.7^{|j'-j|}$.

2. Ω_2: AR(4), $\omega_{j'j} = \mathbb{1}(|j'-j| = 0) + 0.4 \cdot \mathbb{1}(|j'-j| = 1)$
   $+ 0.2 \cdot \mathbb{1}(|j'-j| = 2) + 0.2 \cdot \mathbb{1}(|j'-j| = 3) + 0.1 \cdot \mathbb{1}(|j'-j| = 4)$.

3. $\Omega_3 = B + \delta I$, where each off-diagonal entry in B is generated independently and equals 0.5 with probability $\alpha = 0.1$ or 0 with probability $1 - \alpha = 0.9$. B has zeros on the diagonal, and δ is chosen so that the condition number of Ω_3 is p (keeping the diagonal constant across p would result in either loss of positive definiteness or convergence to identity for larger p).

4. Ω_4: Same as Ω_3 except $\alpha = 0.5$.

All models are sparse (see Figure 2.2), and are numbered in order of decreasing sparsity (or increasing s). Note that the number of non-zero entries in Ω_1 and Ω_2 is proportional to p, whereas Ω_3 and Ω_4 have the expected number of non-zero entries proportional to p^2.

For all models, we generated $n = 100$ multivariate normal training observations and a separate set of 100 validation observations. We considered five different values of p, $30, 100, 200, 500$ and 1000. The estimators were computed on the training data, with the tuning parameter for SPICE selected by minimizing the normal likelihood on the validation data. Using these values of the tuning parameters, we computed the estimated concentration matrix on the training data and compared it to the population concentration matrix.

We evaluate the concentration matrix estimation performance using the Kullback-Leibler loss,

$$\Delta_{KL}(\hat{\Omega}, \Omega) = \text{tr}\left(\Sigma\hat{\Omega}\right) - \log\left|\Sigma\hat{\Omega}\right| - p \ . \tag{2.28}$$

Note that this loss is based on $\hat{\Omega}$ and does not require inversion to compute $\hat{\Sigma}$, which is appropriate for a method estimating Ω. The Kullback-Leibler loss was used by Yuan and Lin (2007) and Levina et al. (2008) to assess performance of methods estimating Ω, and is obtained from the standard entropy loss of the covariance matrix (Lin and Perlman, 1985; Wu and Pourahmadi, 2003; Huang et al., 2006) by reversing the roles of Σ and Ω.

Results for the four covariance models are summarized in Table 2.1, which reports the average loss and the standard error over 50 replications. For Ω_1, Ω_2, and Ω_3, SPICE outperforms the Ledoit-Wolf estimator for all values of p. The sample covariance performs much worse than either estimator in all cases (for $p = 30$). For Ω_4, the least sparse of the four models, the Ledoit-Wolf estimator is about the same as SPICE (sometimes a little better, sometimes a little worse). This suggests, as we would expect from our bound on the rate of convergence, that SPICE provides the

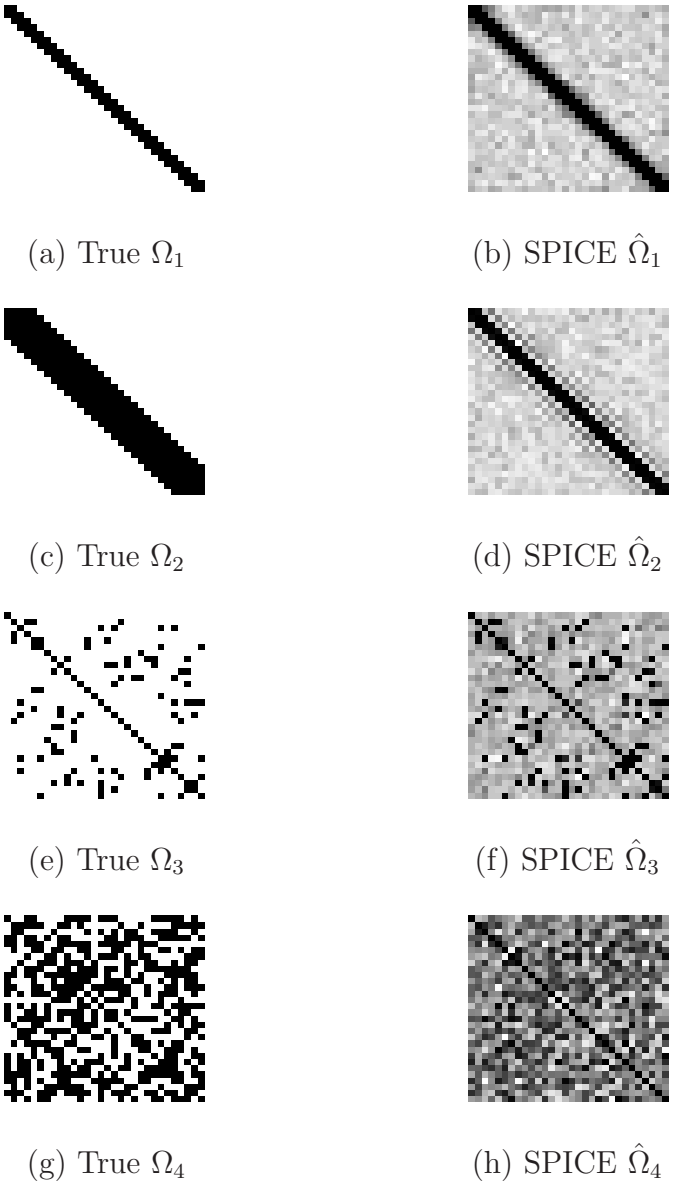
Table 2.1: Simulations: Average (SE) Kullback-Leibler loss over 50 replications.

p	Sample	Ledoit-Wolf	SPICE	Sample	Ledoit-Wolf	SPICE
	Ω_1			Ω_2		
30	8.52(0.14)	3.49(0.04)	1.61(0.03)	8.52(0.14)	2.77(0.02)	2.55(0.03)
100	NA	26.65(0.08)	8.83(0.05)	NA	12.96(0.02)	11.93(0.07)
200	NA	76.83(0.13)	21.23(0.09)	NA	28.16(0.01)	24.82(0.07)
500	NA	262.8(0.19)	78.26(0.26)	NA	74.37(0.02)	63.94(0.12)
1000	NA	594.0(0.13)	174.8(0.20)	NA	151.9(0.04)	133.7(0.20)
	Ω_3			Ω_4		
30	8.45(0.12)	3.50(0.05)	2.12(0.04)	8.45(0.12)	3.04(0.04)	3.77(0.04)
100	NA	29.25(0.44)	17.09(0.10)	NA	19.35(0.15)	21.33(0.06)
200	NA	86.93(1.64)	45.58(0.13)	NA	53.18(0.37)	51.93(0.13)
500	NA	240.3(3.24)	168.7(0.37)	NA	150.4(0.45)	176.6(0.33)
1000	NA	321.5(27.7)	277.3(23.5)	NA	269.8(18.1)	307.3(20.6)

biggest gains in sparse models.

To assess the performance of SPICE on recovering the sparsity structure in the inverse, we report percentages of non-zeros estimated as non-zero (TP %) and percentages of true zeros estimated as zero (TN %) in Table 2.2. We also plot heatmaps of the percentage of time each element was estimated as zero out of the 50 replications in Figure 2.2, for $p = 30$ for all four models. In general, recovering the sparsity structure is easier for smaller p and for sparser models.

Finally, some example computing times: the SPICE algorithm for Ω_2 takes about 2 seconds for $p = 200$, 1 minute for $p = 500$, and 15 minutes for $p = 1000$ on a regular PC. Glasso and SPICE both have complexity $O(p^3)$, but because of the quadratic approximation, SPICE tends to require more iterations to converge, and on average, we have observed a difference in computing times on the order of about 10 between glasso and SPICE. However, this factor does not grow with p, and SPICE computing times are still very reasonable even for large p.



(a) True Ω_1 (b) SPICE $\hat{\Omega}_1$

(c) True Ω_2 (d) SPICE $\hat{\Omega}_2$

(e) True Ω_3 (f) SPICE $\hat{\Omega}_3$

(g) True Ω_4 (h) SPICE $\hat{\Omega}_4$

Figure 2.2: Heatmaps of zeros identified in the concentration matrix out of 50 replications. White color is 50/50 zeros identified, black is 0/50.

Table 2.2: Percentage of correctly estimated non-zeros (TP %) and correctly estimated zeros (TN %) in the concentration matrix (average and SE over 50 replications) for SPICE.

p	TP %	TN %	TP %	TN %
	Ω_1		Ω_2	
30	100(0.00)	68.74(0.31)	50.18(1.44)	75.64(1.28)
100	100(0.00)	74.70(0.08)	49.96(1.10)	72.68(1.21)
200	100(0.00)	73.57(0.04)	27.62(0.12)	96.47(0.02)
500	100(0.00)	91.97(0.01)	22.48(0.09)	98.81(0.00)
1000	100(0.00)	98.95(0.00)	22.29(0.05)	98.82(0.00)
	Ω_3		Ω_4	
30	98.38(0.30)	63.85(1.28)	74.15(0.61)	44.50(0.84)
100	93.90(0.27)	54.01(0.61)	41.27(0.37)	63.07(0.36)
200	70.81(0.13)	69.82(0.05)	35.77(0.06)	66.08(0.06)
500	28.93(0.06)	89.28(0.02)	5.92(0.62)	94.27(0.61)
1000	4.73(0.40)	72.36(6.13)	2.07(0.14)	79.97(5.35)

2.4.2 Colon tumor classification example

In this section, we compare performance of covariance estimators for LDA classification of tumors using gene expression data from Alon et al. (1999). In this experiment, colon adenocarcinoma tissue samples were collected, 40 of which were tumor tissues and 22 non-tumor tissues. Tissue samples were analyzed using an Affymetrix oligonucleotide array. The data were processed, filtered, and reduced to a subset of 2,000 gene expression values with the largest minimal intensity over the 62 tissue samples. Additional information about the dataset and pre-processing can be found in Alon et al. (1999).

To assess the performance at different dimensions, we reduce the full dataset of 2,000 gene expression values by selecting p most significant genes as measured by the two-sample t-statistic, for $p = 50, 100, 200$. Then we use linear discriminant analysis (LDA) to classify these tissues as either tumorous or non-tumorous. We classify each

test observation $\boldsymbol{x}$ to either class $k = 0$ or $k = 1$ using the LDA rule

$$\delta_k(\boldsymbol{x}) = \arg\max_k \left\{ \boldsymbol{x}^T \hat{\Omega} \hat{\boldsymbol{\mu}}_k - \frac{1}{2} \hat{\boldsymbol{\mu}}_k^T \hat{\Omega} \hat{\boldsymbol{\mu}}_k + \log \hat{\pi}_k \right\}, \tag{2.29}$$

where $\hat{\pi}_k$ is the proportion of class k observations in the training data, $\hat{\boldsymbol{\mu}}_k$ is the sample mean for class k on the training data, and $\hat{\Omega}$ is an estimator of the inverse of the common covariance matrix on the training data computed by one of the methods under consideration. Detailed information on LDA can be found in Mardia et al. (1979).

To create training and test sets, we randomly split the data into a training set of size 42 and a testing set of size 20; following the approach used by Wang et al. (2007), we require the training set to have 27 tumor samples and 15 non-tumor samples. We repeat the split at random 100 times and measure the average classification error. The average errors with standard errors over the 100 splits are presented in Table 2.3. We omit the sample covariance because it is not invertible with such a small sample size, and include the naive Bayes classifier instead (where $\hat{\Sigma}$ is estimated by a diagonal matrix with sample variances on the diagonal). Naive Bayes has been shown to perform better than the sample covariance in high-dimensional settings (Bickel and Levina, 2004).

For an application such as classification, there are several possibilities for selecting the tuning parameter. Since we have no separate validation data available, we perform 5-fold cross-validation on the training data. One possibility (columns A in Table 2.3) is to continue using normal likelihood as a criterion for cross-validation, like we did in simulations. Another possibility (columns B in Table 2.3) is to use classification error as the cross-validation criterion, since that is the ultimate performance measure in this case. Table 2.3 shows that for SPICE both methods of tuning perform similarly. For reference, we also include the best error rate achievable on the test data, which

Table 2.3: Averages and SEs of classification errors in % over 100 splits. Tuning parameter for SPICE chosen by (A): 5-fold CV on the training data maximizing the likelihood; (B): 5-fold CV on the training data minimizing the classification error; (C): minimizing the classification error on the test data.

		$p = 50$	$p = 100$	$p = 200$
N. Bayes		15.8(0.77)	20.0(0.84)	23.1(0.96)
L-W		15.2(0.55)	16.3(0.71)	17.7(0.61)
SPICE	A	12.1(0.65)	18.7(0.84)	18.3(0.66)
SPICE	B	14.7(0.73)	16.9(0.85)	18.0(0.70)
SPICE	C	9.0(0.57)	9.1(0.51)	10.2(0.52)

is obtained by selecting the tuning parameter to minimize the classification error on the test data (columns C in Table 2.3). SPICE provides the best improvement over naive Bayes and Ledoit-Wolf for $p = 50$; for larger p, as less informative genes are added into the pool, the performance of all methods worsens.

2.5 Discussion

We have analyzed a penalized likelihood approach to estimating a sparse concentration matrix via a lasso-type penalty, and showed that its rate of convergence depends explicitly on how sparse the true matrix is. This is analogous to results for banding (Bickel and Levina, 2008b), where the rate of convergence depends on how quickly the off-diagonal elements of the true covariance decay, and for thresholding (Bickel and Levina, 2008a; El Karoui, 2008), where the rate also depends on how sparse the true covariance is by various definitions of sparsity. We conjecture that other structures can be similarly dealt with, and other types of penalties may show similar behavior when applied to the "right" type of structure – for example, a ridge, bridge, or other more complex penalty may work well for a model that is not truly sparse but has many small entries. A generalization of this work to other penalties has been recently completed by Lam and Fan (2009), who have also proved "sparsistency" of SPICE-type estimators.

While we assumed normality, it can be replaced by a tail condition, analogously to Bickel and Levina (2008b). The use of normal likelihood is, of course, less justifiable if we do not assume normality, but it was found empirically that it still works reasonably well as a loss function even if the true distribution is not normal (Levina et al., 2008).

The Cholesky decomposition of covariance was only considered appropriate when variables are ordered, and we have shown it to be a useful tool for enforcing positive definiteness of the estimator even when variables have no natural ordering. Our optimization algorithm has complexity of $O(p^3)$ and is equally applicable to general l_q penalties.

2.6 Derivation of the SPICE Algorithm

In this section we give a full derivation of the parameter update equations involved in the optimization algorithm. Recall that we have re-parametrized the objective function (2.20) using (2.22)–(2.24). We cycle through the parameters in T and for each t_{lc}, compute partial derivatives with respect to t_{lc} while holding all other parameters fixed, and solve the univariate linear equation corresponding to setting this partial derivative to 0.

For simplicity, we separate the likelihood and the penalty by writing $\tilde{f}(T) = \ell(T) + P(T)$. We also suppress the ϵ-perturbation in the denominator for simplicity of notation. For the likelihood part, taking the partial derivative with respect to t_{lc}, $1 \le c \le p,\ c \le l \le p$ gives

$$\begin{aligned}\frac{\partial}{\partial t_{\ell c}}\ell(T) &= -2\underbrace{\frac{\partial}{\partial t_{\ell c}}\sum_{j=1}^{p}\log t_{jj}}_{=0 \text{ if } j\neq c} + \frac{1}{n}\sum_{i=1}^{n}\underbrace{\frac{\partial}{\partial t_{\ell c}}\sum_{j=1}^{p}\left(\sum_{k=1}^{j} t_{jk}X_{ik}\right)^2}_{=0 \text{ if } j\neq l} \\ &= \frac{-2}{t_{cc}}\mathbb{1}\{l=c\} + t_{lc}\left[2\hat{\sigma}_{cc}\right] + 2\sum_{k=1,\ k\neq c}^{l} t_{lk}\hat{\sigma}_{kc}, \end{aligned} \tag{2.30}$$

For the penalty part, using the quadratic approximation (2.25) gives

$$\frac{\partial}{\partial t_{\ell c}} P(T) \approx \frac{\partial}{\partial t_{\ell c}} \sum_{j'>j} \frac{\lambda q}{|\omega^0_{j'j}|^{2-q}} \omega^2_{j'j} = \sum_{k=1,k\neq c}^{l} \frac{\lambda q}{|\omega^0_{ck}|^{2-q}} \frac{\partial}{\partial t_{\ell c}} \omega^2_{ck} \ , \tag{2.31}$$

since the only nonzero terms in (2.31) are those for which $j' \leq l$ and either $j' = c$ or $j = c$. For $1 \leq k \leq l$ such that $k \neq c$, we have $\frac{\partial}{\partial t_{\ell c}} \omega^2_{ck} = 2\omega_{ck} t_{lk}$, and collecting terms together we get

$$\frac{\partial}{\partial t_{\ell c}} P(T) = t_{lc} \left[2\lambda q \sum_{k=1,k\neq c}^{l} \frac{t^2_{lk}}{|\omega^0_{ck}|^{2-q}} \right] + 2\lambda q \sum_{k=1,k\neq c}^{l} \frac{(\omega_{ck} - t_{lc} t_{lk}) t_{lk}}{|\omega^0_{ck}|^{2-q}} \ . \tag{2.32}$$

Combining together (2.30) and (2.32), we have the parameter update equation for t_{lc} when $l \neq c$, is given by

$$\hat{t}_{lc} = \frac{-\sum_{k=1,\ k\neq c}^{l} t_{lk} \hat{\sigma}_{kc} - \lambda q \sum_{k=1,k\neq c}^{l} (\omega_{ck} - t_{lc} t_{lk}) t_{lk} |\omega^0_{ck}|^{q-2}}{\hat{\sigma}_{cc} + \lambda q \sum_{k=1,k\neq c}^{l} t^2_{lk} |\omega^0_{ck}|^{q-2}} \ .$$

If $l = c$, we solve $au^2 + bu - 1 = 0$ for u using the quadratic formula, where

$$\begin{aligned} a &= \hat{\sigma}_{cc} + \lambda q \sum_{k=1,k\neq c}^{l} t^2_{lk} |\omega^0_{ck}|^{q-2} \ , \\ b &= \sum_{k=1,\ k\neq c}^{l} t_{lk} \hat{\sigma}_{kc} + \lambda q \sum_{k=1,k\neq c}^{l} (\omega_{ck} - t_{lc} t_{lk}) t_{lk} |\omega^0_{ck}|^{q-2} \ , \end{aligned}$$

then take the positive solution $\hat{t}_{cc} = u^+$.

We also can quickly update the ω_{ck} involving t_{lc} via

$$\omega_{ck} = \omega^0_{ck} + t_{lk}(\hat{t}_{lc} - t_{lc}) \ .$$

CHAPTER III

Generalized thresholding of large covariance matrices

3.1 Introduction

In this chapter we propose a new class of generalized thresholding operators which combine thresholding with shrinkage, and study generalized thresholding of the sample covariance matrix in high dimensions. Bickel and Levina (2008a) and El Karoui (2008) showed favorable large p convergence rates for hard thresholding of the sample covariance matrix (i.e. setting elements in the sample covariance matrix to zero if their magnitude falls below a thresholding parameter λ). We specifically generalize the hard thresholding approach to covariance estimation to a whole class of estimators based on element-wise shrinkage and thresholding. For any $\lambda \geq 0$, define a generalized thresholding operator to be a function $s_\lambda : \mathbb{R} \to \mathbb{R}$ satisfying the following conditions for all $z \in \mathbb{R}$:

(i) $|s_\lambda(z)| \leq |z|$;

(ii) $s_\lambda(z) = 0$ for $|z| \leq \lambda$;

(iii) $|s_\lambda(z) - z| \leq \lambda$.

Condition (i) establishes shrinkage, condition (ii) enforces thresholding, and condition (iii) limits the amount of shrinkage to no more than λ. It is possible to have different

parameters λ_1 and λ_2 in (ii) and (iii); for simplicity, we keep them the same. For a related discussion of penalties that have such properties, see also Antoniadis and Fan (2001).

This chapter is organized as follows. To make our definition of generalized thresholding concrete, we start by giving examples in Section 3.2, and show that generalized thresholding covers many popular shrinkage/thresholding functions, including hard and soft thresholding, SCAD (Fan and Li, 2001), and adaptive lasso (Zou, 2006). In Section 3.3, we establish convergence rates for generalized thresholding of the sample covariance on a class of "approximately sparse" matrices, and show they are consistent as long as $\log p/n$ tends to 0. We also show that generalized thresholding is, in the terminology of Lam and Fan (2009), "sparsistent", meaning that in addition to being consistent it estimates true zeros as zeros with probability tending to 1, and, under an additional condition, estimates non-zero elements as non-zero, with the correct sign, with probability tending to 1. This property is sometimes referred to as sign consistency. Simulation results are given in Section 3.4, where we show that while all the estimators in this class are guaranteed the same bounds on convergence rates and have similar performance in terms of overall loss, the more flexible penalties like SCAD are substantially better at getting the true sparsity structure since in practice one must select the tuning parameter. Finally, Section 3.5 presents an application of the methods to gene expression data on small round blue-cell tumors (SRBC).

3.2 Examples of generalized thresholding

It turns out that conditions (i)–(iii) which define generalized thresholding are satisfied by a number of commonly used shrinkage/thresholding procedures. These procedures are commonly introduced as solutions to penalized quadratic loss problems with various penalties. Since in our case the procedure is applied to each element

separately, the optimization problems are univariate. Suppose $s_\lambda(z)$ is obtained as

$$s_\lambda(z) = \arg\min_\theta \{\frac{1}{2}(\theta - z)^2 + p_\lambda(\theta)\} \ , \tag{3.1}$$

where p_λ is a penalty function. Next, we check that several popular penalties and thresholding rules satisfy our conditions for generalized thresholding. For more details on the relationship between penalty functions and resulting thresholding rules, see Antoniadis and Fan (2001).

The simplest example of generalized thresholding is the hard thresholding rule,

$$s_\lambda^H(z) = z\mathbb{1}(|z| > \lambda) \ , \tag{3.2}$$

where $\mathbb{1}(\cdot)$ is the indicator function. Hard thresholding obviously satisfies conditions (i)–(iii).

Soft thresholding results from solving (3.1) with the lasso (ℓ_1) penalty function, $p_\lambda(\theta) = \lambda|\theta|$, and gives the rule

$$s_\lambda^S(z) = \text{sign}(z)(|z| - \lambda)_+ \ . \tag{3.3}$$

Soft thresholding has been studied in the context of wavelet shrinkage by Donoho and Johnstone (1994) and Donoho et al. (1995), and in the context of regression by Tibshirani (1996). The soft-thresholding operator s_λ^S obviously satisfies conditions (i) and (ii). To check (iii), note that $|s_\lambda^S(z) - z| = |z|$ when $|z| \leq \lambda$, and $|s_\lambda^S(z) - z| = \lambda$ when $|z| > \lambda$. Thus soft thresholding corresponds to the maximum amount of shrinkage allowed by condition (iii), whereas hard thresholding corresponds to no shrinkage.

The smoothly clipped absolute deviation (SCAD) penalty was proposed by Fan (1997) and Fan and Li (2001) as a compromise between hard and soft thresholding.

Like soft thresholding, it is continuous in z, but the amount of shrinkage decreases as $|z|$ increases and after a certain threshold there is no shrinkage, which results in less bias. The SCAD thresholding function is a linear interpolation between soft thresholding up to 2λ and hard thresholding after $a\lambda$ (see Figure 3.1). The value $a = 3.7$ was recommended by Fan and Li (2001), and we use it throughout the paper. See Fan and Li (2001) for the formulae of the SCAD thresholding function and the corresponding penalty function. The SCAD thresholding operator s_λ^{SC} satisfies conditions (i)–(iii): (ii) is immediate, and (i) and (iii) follow from $|s^S(|z|)| \leq |s^{SC}(|z|)| \leq |s^H(|z|)|$.

Another idea proposed to mitigate the bias of lasso for large regression coefficients is adaptive lasso (Zou, 2006). In regression context, the idea is to multiply each $|\beta_j|$ in the lasso penalty by a weight w_j, which is smaller for larger initial estimates $\hat{\beta}_j$. Thus large coefficients get penalized less. One choice of weights proposed was $w_j = |\hat{\beta}_j|^{-\eta}$, where $\hat{\beta}_j$ are ordinary least squares estimates. Note that in the context of regression, the special case $\eta = 1$ is closely related to the non-negative garrote (Breiman, 1995). In our context, an analogous weight would be $|\hat{\sigma}_{ij}|^{-\eta}$. We can rewrite this as a penalty function $p_\lambda(\theta) = \lambda w(z)|\theta|$, where w is taken to be $C|z|^{-\eta}$, $\eta \geq 0$. Zou (2006) have $C = 1$ (it is absorbed in λ), but for us it is convenient to set $C = \lambda^{-\eta}$, because then the resulting operator satisfies condition (ii), i.e., thresholds everything below λ to 0. The resulting thresholding rule corresponding to $C = \lambda^{-\eta}$, which we still call adaptive lasso for simplicity, is given by

$$s_\lambda^{AL}(z) = \mathrm{sgn}(z)(|z| - \lambda^{\eta+1}|z|^{-\eta})_+ \tag{3.4}$$

Conditions (i) and (ii) are obviously satisfied. To check (iii) for $|z| > \lambda$, note that $|s_\lambda^{AL}(z) - z| = \lambda^{\eta+1}|z|^{-\eta} \leq \lambda$.

As illustrated in Figure 3.1, both SCAD and adaptive lasso fall in between hard and soft thresholding; any other function sandwiched between hard and soft thresh-

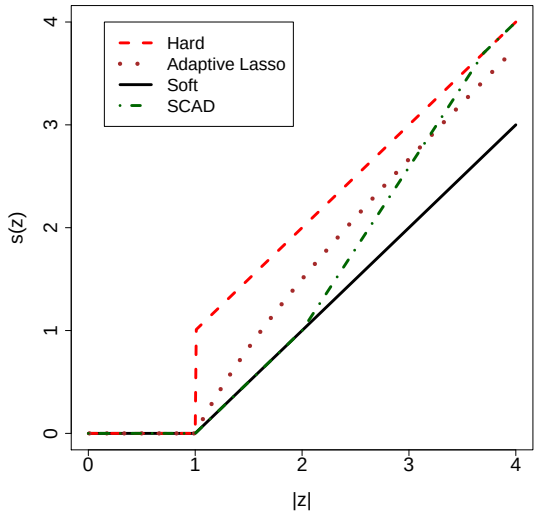


Figure 3.1: Generalized thresholding functions for $\lambda = 1$, $a = 3.7$, $\eta = 1$.

olding will satisfy conditions (i)–(iii), for example, the clipped L_1 penalty. For conditions on the penalty p_λ that imply the resulting operator is sandwiched between hard and soft thresholding, see Antoniadis and Fan (2001). In this paper, we focus on the operators themselves rather than the penalties, since the penalties are never used directly.

3.3 Consistency and sparsity of generalized thresholding

In this section, we derive theoretical properties of the generalized thresholding estimator in the high-dimensional setting, meaning that both the dimension and the sample size are allowed to grow. Let $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ denote i.i.d. p-dimensional random vectors sampled from a distribution F with $E\boldsymbol{X}_1 = 0$ (without loss of generality), and $E(\boldsymbol{X}_1 \boldsymbol{X}_1^T) = \Sigma$. The convention in the literature is to assume that F is Gaussian. However, the key result underlying this theory is the bound (3.11), and Bickel and Levina (2008b) noted that for this result the normal assumption can be replaced with

a tail condition on the marginal distributions, namely that for all $1 \leq j \leq p$,

$$E(e^{tX_{1j}^2}) < \infty \ , \tag{3.5}$$

for $t \in (-t_0, t_0)$, for some $t_0 > 0$.

Let $\hat{\Sigma}$ denote the sample covariance matrix,

$$\hat{\Sigma} = \frac{1}{n} \sum_{k=1}^{n} \left(\boldsymbol{X}_k - \bar{\boldsymbol{X}} \right) \left(\boldsymbol{X}_k - \bar{\boldsymbol{X}} \right)^T \ . \tag{3.6}$$

Let $s_\lambda(A) = [s_\lambda(a_{ij})]$ denote the matrix resulting from applying a generalized thresholding operator s_λ to each of the elements of a matrix A. Condition (ii) implies that $s_\lambda(A)$ is sparse for sufficiently large λ. Like with hard thresholding and banding of the covariance matrix, the estimator $s_\lambda(\hat{\Sigma})$ is not guaranteed to be positive definite, but instead we show that it converges to a positive definite limit with probability tending to 1.

We proceed to establish a bound on the convergence rate for $s_\lambda(\hat{\Sigma})$. The result is uniform on a class of "approximately sparse" covariance matrices which was introduced by Bickel and Levina (2008a):

$$\mathcal{U}_\tau(q, c_0(p), M) = \{ \Sigma : \sigma_{ii} \leq M, \ \max_i \sum_{j=1}^{p} |\sigma_{ij}|^q \leq c_0(p) \ \} \ , \tag{3.7}$$

for $0 \leq q < 1$. When $q = 0$, this is a class of truly sparse matrices. For example, a d-diagonal matrix satisfies this condition with any $0 \leq q < 1$ and $c_0(p) = M^q d$. Another example is the AR(1) covariance matrix, $\sigma_{ij} = \rho^{|i-j|}$, which satisfies the condition with $c_0(p) \equiv c_0$. Note that the condition of bounded variances, $\sigma_{ii} \leq M$, is weaker than the often assumed bounded eigenvalues condition, $\lambda_{\max}(\Sigma) \leq M$. Also note that the constant $c_0(p)$ is allowed to depend on p and is thus not an explicit restriction on sparsity. The convergence will be established in the matrix operator

norm (also known as spectral or l_2 matrix norm), $\|A\|^2 = \lambda_{\max}(AA^T)$.

Theorem III.1 (Consistency)**.** *Suppose s_λ satisfies conditions (i)–(iii) and F satisfies condition* (3.5)*. Then, uniformly on $\mathcal{U}_\tau\big(q, c_0(p), M\big)$, for sufficiently large M', if $\lambda = M'\sqrt{\frac{\log p}{n}} = o(1)$,*

$$\|s_\lambda(\hat{\Sigma}) - \Sigma\| = O_P\left(c_0(p)\left(\frac{\log p}{n}\right)^{\frac{1-q}{2}}\right) .$$

To prove Theorem III.1 we start from a Lemma summarizing several earlier results we will use. The proofs and/or further references for these can be found in Bickel and Levina (2008a).

Lemma III.2. *Under conditions of Theorem III.1,*

$$\max_i \sum_{j=1}^p |\hat{\sigma}_{ij}| \mathbb{1}(|\hat{\sigma}_{ij}| \geq \lambda,\ |\sigma_{ij}| < \lambda) = O_P\left(c_0(p)\lambda^{-q}\sqrt{\frac{\log p}{n}} + c_0(p)\lambda^{1-q}\right) \tag{3.8}$$

$$\max_i \sum_{j=1}^p |\sigma_{ij}|\ \mathbb{1}(|\hat{\sigma}_{ij}| < \lambda,\ |\sigma_{ij}| \geq \lambda) = O_P\left(c_0(p)\lambda^{-q}\sqrt{\frac{\log p}{n}} + c_0(p)\lambda^{1-q}\right) \tag{3.9}$$

$$\max_i \sum_{j=1}^p |\hat{\sigma}_{ij} - \sigma_{ij}|\ \mathbb{1}(|\hat{\sigma}_{ij}| \geq \lambda,\ |\sigma_{ij}| \geq \lambda) = O_P\left(c_0(p)\lambda^{-q}\sqrt{\frac{\log p}{n}}\right) \tag{3.10}$$

$$P(\max_{i,j} |\hat{\sigma}_{ij} - \sigma_{ij}| > t) \leq C_1 p^2 e^{-nC_2t^2} + C_3 p e^{-nC_4 t} \tag{3.11}$$

where $t = o(1)$ and C_1, C_2, C_3, C_4 depend only on M.

Proof of Theorem III.1. We start from the decomposition

$$\|s_\lambda(\hat{\Sigma}) - \Sigma\| \leq \|s_\lambda(\Sigma) - \Sigma\| + \|s_\lambda(\hat{\Sigma}) - s_\lambda(\Sigma)\| . \tag{3.12}$$

For symmetric matrices, the operator norm satisfies (see e.g., Golub and Van Loan

(1989)),

$$\|A\| \leq \max_i \sum_j |a_{ij}| \ . \tag{3.13}$$

That is, the operator norm is bounded by the matrix l_1 or l_∞ norm, which coincide for symmetric matrices. From this point on, we bound all the operator norms by (3.13). For the first term in (3.12), note that by assumptions (ii) and (iii),

$$\begin{aligned}\sum_{j=1}^p |s_\lambda(\sigma_{ij}) - \sigma_{ij}| &\leq \sum_{j=1}^p |\sigma_{ij}| \mathbb{1}(|\sigma_{ij}| \leq \lambda) + \sum_{j=1}^p \lambda \mathbb{1}(|\sigma_{ij}| > \lambda) \\ &= \sum_{j=1}^p |\sigma_{ij}|^q |\sigma_{ij}|^{1-q} \mathbb{1}(|\sigma_{ij}| \leq \lambda) + \sum_{j=1}^p \lambda^q \lambda^{1-q} \mathbb{1}(|\sigma_{ij}| > \lambda) \leq \lambda^{1-q} \sum_{j=1}^p |\sigma_{ij}|^q \ ,\end{aligned}$$

and therefore by (3.13) and the definition (3.7) the first term in (3.12) is bounded by $\lambda^{1-q} c_0(p)$.

For the second term in (3.12), note that by (i) and (ii),

$$\begin{aligned}|s_\lambda(\hat{\sigma}_{ij}) - s_\lambda(\sigma_{ij})| &\leq |\hat{\sigma}_{ij}| \mathbb{1}(|\hat{\sigma}_{ij}| \geq \lambda, \ |\sigma_{ij}| < \lambda) + |\sigma_{ij}| \mathbb{1}(|\hat{\sigma}_{ij}| < \lambda, \ |\sigma_{ij}| \geq \lambda) \\ &\quad + \big(|\hat{\sigma}_{ij} - \sigma_{ij}| + |s_\lambda(\hat{\sigma}_{ij}) - \hat{\sigma}_{ij}| + |s_\lambda(\sigma_{ij}) - \sigma_{ij}|\big) \mathbb{1}(|\hat{\sigma}_{ij}| \geq \lambda, \ |\sigma_{ij}| \geq \lambda)\end{aligned} \tag{3.14}$$

The first three terms in (3.14) are controlled by (3.8), (3.9), and (3.10), respectively. For the fourth term, applying (iii) we have

$$\begin{aligned}\max_i \sum_{j=1}^p |s_\lambda(\hat{\sigma}_{ij}) - \hat{\sigma}_{ij}| \mathbb{1}(|\hat{\sigma}_{ij}| \geq \lambda, \ |\sigma_{ij}| \geq \lambda) &\leq \max_i \sum_{j=1}^p \lambda^q \lambda^{1-q} \mathbb{1}(|\hat{\sigma}_{ij}| \geq \lambda, \ |\sigma_{ij}| \geq \lambda) \\ &\leq \lambda^{1-q} \max_i \sum_{j=1}^p |\sigma_{ij}|^q \mathbb{1}(|\sigma_{ij}| \geq \lambda) \leq \lambda^{1-q} c_0(p) \ .\end{aligned}$$

Similarly, for the last term in (3.14) we have,

$$\max_i \sum_{j=1}^p |s_\lambda(\sigma_{ij}) - \sigma_{ij}| \mathbb{1}(|\hat{\sigma}_{ij}| \geq \lambda, \ |\sigma_{ij}| \geq \lambda) \leq \lambda^{1-q} c_0(p) \ .$$

Collecting all the terms, we obtain

$$\|s_{\lambda_n}(\hat{\Sigma}) - \Sigma\| = O_P\left(c_0(p)\left(\lambda^{1-q} + \lambda^{-q}\sqrt{\frac{\log p}{n}}\right)\right)$$

and the theorem follows by substituting $\lambda = M'\sqrt{\frac{\log p}{n}}$. □

For the case of hard thresholding, this theorem was established in Bickel and Levina (2008a). Note that, through $c_0(p)$, the rate explicitly depends on how sparse the truth is. Also note that this rate is very similar to the rate of $\sqrt{\frac{s \log p}{n}}$ for a sparse estimator of the *inverse* covariance matrix established in Rothman et al. (2008), where s is the number of non-zero off-diagonal elements in the true inverse, even though the estimator is obtained by a completely different approach of adding a lasso penalty to the normal likelihood. However, the fundamental result underlying these different analyses is the bound (3.11), which ultimately gives rise to similar rates.

Next, we state a sparsity result, which, together with Theorem III.1, establishes the "sparsistency" property in the sense of Lam and Fan (2009).

Theorem III.3 (Sparsity)**.** *Suppose s_λ satisfies conditions (i)–(iii), F satisfies* (3.5)*, and $\sigma_{ii} \leq M$ for all i. Then, for sufficiently large M', if $\lambda = M'\sqrt{\frac{\log p}{n}} = o(1)$,*

$$s_\lambda(\hat{\sigma}_{ij}) = 0 \ \textit{for all} \ (i,j) \ \textit{such that} \ \sigma_{ij} = 0 \ , \tag{3.15}$$

with probability tending to 1. If we additionally assume that all non-zero elements of Σ satisfy $|\sigma_{ij}| > \tau$, where $\sqrt{n}(\tau - \lambda) \to \infty$, we also have, with probability tending to 1,

$$\operatorname{sgn}(s_\lambda(\hat{\sigma}_{ij}) \cdot \sigma_{ij}) = 1 \ \textit{for all} \ (i,j) \ \textit{such that} \ \sigma_{ij} \neq 0 \ . \tag{3.16}$$

Proof of Theorem III.3. To prove (3.15), apply (ii) to get

$$\{(i,j) : s_\lambda(\hat{\sigma}_{ij}) \neq 0, \sigma_{ij} = 0\} = \{(i,j) : |\hat{\sigma}_{ij}| > \lambda, \sigma_{ij} = 0\} \subseteq \{(i,j) : |\hat{\sigma}_{ij} - \sigma_{ij}| > \lambda\} \ .$$

Therefore

$$P\big(\sum_{i,j} \mathbb{1}(s_\lambda(\hat{\sigma}_{ij}) \neq 0, \sigma_{ij} = 0) > 0\big) \leq P(\max_{i,j} |\hat{\sigma}_{ij} - \sigma_{ij}| > \lambda) \ . \tag{3.17}$$

Now we apply (3.11). With the choice $\lambda = M'\sqrt{\frac{\log p}{n}}$, the first term dominates the second one, so we only need to make sure $C_1 p^2 e^{-nC_2\lambda^2} \to 0$. Since we can choose M' large enough so that $2 - C_2 M'^2 < 0$, the probability in (3.17) tends to 0.

Similarly, for (3.16) we have,

$$\{(i,j) : s_\lambda(\hat{\sigma}_{ij}) \leq 0, \sigma_{ij} > 0 \ \text{ or } \ s_\lambda(\hat{\sigma}_{ij}) \geq 0, \sigma_{ij} < 0\} \subseteq \{(i,j) : |\hat{\sigma}_{ij} - \sigma_{ij}| > \tau - \lambda\} \ ,$$

and applying the bound (3.11) and the additional condition $\sqrt{n}(\tau - \lambda) \to \infty$ gives

$$P\big(\sum_{i,j} \mathbb{1}(|\hat{\sigma}_{ij} - \sigma_{ij}| \geq \tau - \lambda) > 0\big) \leq C_1 p^2 e^{-nC_2(\tau-\lambda)^2} \to 0 \ .$$

□

Note that Theorem III.3 only requires that the true variances are bounded, and not the approximately sparse assumption. The additional condition on non-zero elements is analogous to the condition of El Karoui (2008) that non-zero elements are greater than $n^{-\alpha}$. If we assume the same, i.e., let $\tau = n^{-\alpha}$, the result holds under a slightly stronger condition $\log p / n^{1-2\alpha} \to 0$ instead of $\log p / n \to 0$. It may also be possible to develop further joint asymptotic normality results for non-zero elements along the lines of Fan and Peng (2004) or Lam and Fan (2009), but we do not pursue this further because of restrictive conditions required for the method of proof used there ($p^2/n \to 0$).

3.4 Simulation results

3.4.1 Simulation settings

To compare the performance of various generalized thresholding estimators, both in terms of the overall covariance estimation and recovering the sparsity pattern, we conducted a simulation study with the following three covariance models.

Model 1: AR(1), where $\sigma_{ij} = \rho^{|i-j|}$, for $\rho = 0.3$ and 0.7;

Model 2: MA(1), where $\sigma_{ij} = \rho \mathbb{1}(|i-j| = 1) + \mathbb{1}(i = j)$, for $\rho = 0.3$;

Model 3: "Triangular" covariance, $\sigma_{ij} = (1 - \frac{|i-j|}{k})_+$, for $k = \lfloor p/2 \rfloor$.

Models 1 and 2 are standard test cases in the literature. Note that even though these models come from time series, all estimators considered here are permutation invariant, and thus the order of the variables is irrelevant. Model 1 is "approximately sparse", because even though there are no true zeros, there are many very small entries away from the diagonal. Model 2 is a tri-diagonal covariance matrix and is the most sparse of the three models. Model 3 has a linear decay in covariances as one moves away from the diagonal and provides a simple way to generate a positive definite matrix with the level of sparsity controlled by the parameter k. With $k = p/2$, model 3 is effectively the least sparse of the three models we consider. This covariance structure was considered by Wagaman and Levina (2009).

For each model, we generated $n = 100$ independent and identically distributed p-variate normal random vectors with mean 0 and covariance Σ, for $p = 30, 100, 200,$ and 500. The number of replications was fixed at 50. The tuning parameter λ for each method was selected by minimizing the Frobenius norm of the difference between $s_\lambda(\hat{\Sigma})$ and the sample covariance matrix computed from 100 independently generated validation data observations. We note that the use of a validation set can be replaced with cross-validation without any significant change in results. We

selected the Frobenius norm ($\|A\|_F^2 = \sum_{i,j} a_{ij}^2$) for tuning because it had a slightly better performance than the operator norm or the matrix l_1 norm. Also, a theoretical justification for this choice for cross-validation has been provided by Bickel and Levina (2008a).

3.4.2 Performance Evaluation

Keeping consistent with theory in Section 3.3, we defined the loss function for the estimators by the expected operator norm of the difference between the true covariance and the estimator,

$$L(s_\lambda(\hat{\Sigma}), \Sigma) = E\|s_\lambda(\hat{\Sigma}) - \Sigma\| \ .$$

The ability to recover sparsity was evaluated via the true positive rate (TPR) in combination with the false positive rate (FPR), defined as

$$\text{TPR} = \frac{\#\{(i,j) : s_\lambda(\hat{\sigma}_{ij}) \neq 0 \quad \text{and} \quad \sigma_{ij} \neq 0\}}{\#\{(i,j) : \sigma_{ij} \neq 0\}} \ , \tag{3.18}$$

$$\text{FPR} = \frac{\#\{(i,j) : s_\lambda(\hat{\sigma}_{ij}) \neq 0 \quad \text{and} \quad \sigma_{ij} = 0\}}{\#\{(i,j) : \sigma_{ij} = 0\}} \ . \tag{3.19}$$

Note that the sample covariance has TPR = 1, and a diagonal estimator has FPR = 0.

In addition, we compute a measure of agreement of principal eigenspaces between the estimator and the truth, which is relevant for principal components analysis. The measure we use to compare the eigenspaces spanned by the first q eigenvectors was defined by Krzanowski (1979) as

$$K(q) = \sum_{i=1}^{q} \sum_{j=1}^{q} (\hat{\boldsymbol{e}}_{(i)}^T \boldsymbol{e}_{(j)})^2, \tag{3.20}$$

where $\hat{\boldsymbol{e}}_{(i)}$ denotes the estimated eigenvector corresponding to the i-th largest esti-

mated eigenvalue, and $\boldsymbol{e}_{(i)}$ is the true eigenvector corresponding to the true i-th largest eigenvalue. Computing cosines of angles between all possible pairs of eigenvectors removes the problem of similar eigenvectors estimated in a different order. Note that $K(0) \equiv 0$ and $K(p) = p$. For any $0 < q < p$, perfect agreement between the two eigenspaces will result in $K(q) = q$. A convenient way to evaluate this measure is to plot $K(q)$ against q. Alternative measures of eigenvector agreement are available; for example, Fan et al. (2008b) proposed using the measure

$$D(q) = 1 - \frac{1}{q} \sum_{i=1}^{q} \max_{1 \le j \le q} |\boldsymbol{e}_{(i)}^T \hat{\boldsymbol{e}}_j| \ ,$$

which shares many of the properties of the Krzanowski's measure, such as invariance to permutations of the eigenvector order.

3.4.3 Summary of results

Table 3.1 summarizes simulation results for the AR(1) model. Note that this model is not truly sparse, and thus true and false positive rates are not relevant. All generalized thresholding estimators improve over the sample covariance matrix under the operator norm loss. This improvement increases with dimension p. The thresholding rules are all quite similar for this model, with perhaps hard thresholding having a slight edge for $\rho = 0.7$ (more large entries) and being slightly worse than the others for $\rho = 0.3$.

Table 3.2 gives results for Model 2, the tri-diagonal sparse truth. We again see a drastic improvement in estimation performance of the thresholded estimates over the sample covariance matrix, which increases with dimension. This is expected since this is the sparsest model we consider. Under operator norm loss, the rules that combine thresholding with shrinkage all outperform hard thresholding, with soft thresholding performing slightly better than SCAD and adaptive lasso.

Table 3.1: Average(SE) operator norm loss for Model 1.

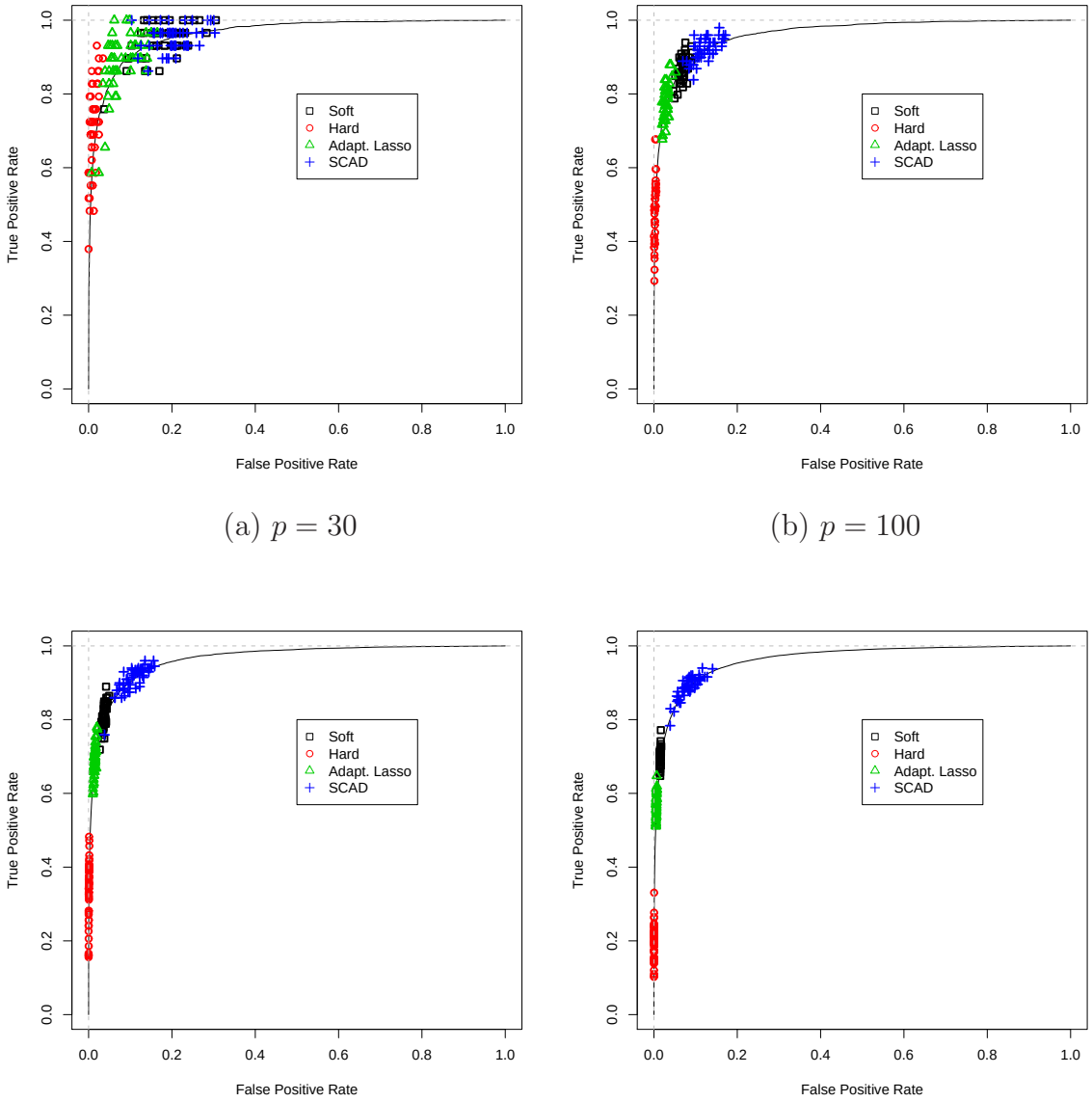
p	ρ	Sample	Hard	Soft	Adapt.lasso	SCAD
30	0.3	1.30(0.02)	0.75(0.01)	0.71(0.01)	0.71(0.01)	0.71(0.01)
30	0.7	1.75(0.04)	1.56(0.04)	1.59(0.05)	1.53(0.04)	1.47(0.04)
100	0.3	3.09(0.03)	0.93(0.01)	0.86(0.01)	0.86(0.01)	0.85(0.01)
100	0.7	4.10(0.07)	2.17(0.04)	2.49(0.03)	2.30(0.04)	2.16(0.04)
200	0.3	4.90(0.03)	0.98(0.01)	0.90(0.00)	0.91(0.01)	0.90(0.00)
200	0.7	6.63(0.08)	2.46(0.03)	2.86(0.02)	2.65(0.03)	2.52(0.03)
500	0.3	9.69(0.04)	1.06(0.01)	0.95(0.00)	0.96(0.00)	0.95(0.00)
500	0.7	12.54(0.08)	2.80(0.02)	3.23(0.02)	3.01(0.02)	2.97(0.02)

Table 3.2: Average(SE) operator norm loss and true and false positive rates for Model 2.

p	Sample	Hard	Soft	Adapt.lasso	SCAD
	Operator norm loss				
30	1.34(0.02)	0.69(0.01)	0.61(0.01)	0.62(0.01)	0.63(0.01)
100	2.99(0.02)	0.88(0.01)	0.70(0.01)	0.73(0.01)	0.72(0.01)
200	4.94(0.03)	0.94(0.02)	0.75(0.01)	0.78(0.01)	0.76(0.01)
500	9.65(0.04)	1.01(0.02)	0.81(0.01)	0.85(0.01)	0.81(0.01)
	TPR/FPR				
30	NA	0.70/0.01	0.94/0.18	0.88/0.08	0.95/0.21
100	NA	0.49/0.00	0.87/0.07	0.78/0.03	0.92/0.12
200	NA	0.33/0.00	0.81/0.04	0.69/0.01	0.91/0.11
500	NA	0.20/0.00	0.70/0.02	0.57/0.01	0.89/0.08

The 50 realizations of the values of TPR and FPR are also plotted in Figure 3.2, in addition to their average values given in Table 3.2. Here we see a big difference between the different thresholding rules. Hard thresholding tends to zero out too many elements, presumably due to its inability to shrink moderate values; thus it has a very low false positive rate, but also a lower true positive rate than the other methods, particularly for large p. Overall, Figure 3.2 suggests that the SCAD thresholding has the best performance on sparsity for this model, particularly for large values of p.

Table 3.3 gives results for the "triangular" model with $k = p/2$, the least sparse of the three models we consider. Here we see only a small improvement of thresholded estimates over the sample covariance in the operator norm loss. All methods miss a



(a) $p = 30$ (b) $p = 100$

(c) $p = 200$ (d) $p = 500$

Figure 3.2: TPR vs. FPR for Model 2. The points correspond to 50 different realizations, with each method selecting its own threshold using validation data. The solid line is obtained by varying the threshold over the whole range (all methods have the same TPR and FPR for a fixed threshold).

substantial fraction of true zeros, most likely because a large number of small non-zero true entries leads to a choice of threshold that is too low. In this case, hard thresholding does somewhat better on false positives, which we conjecture may in

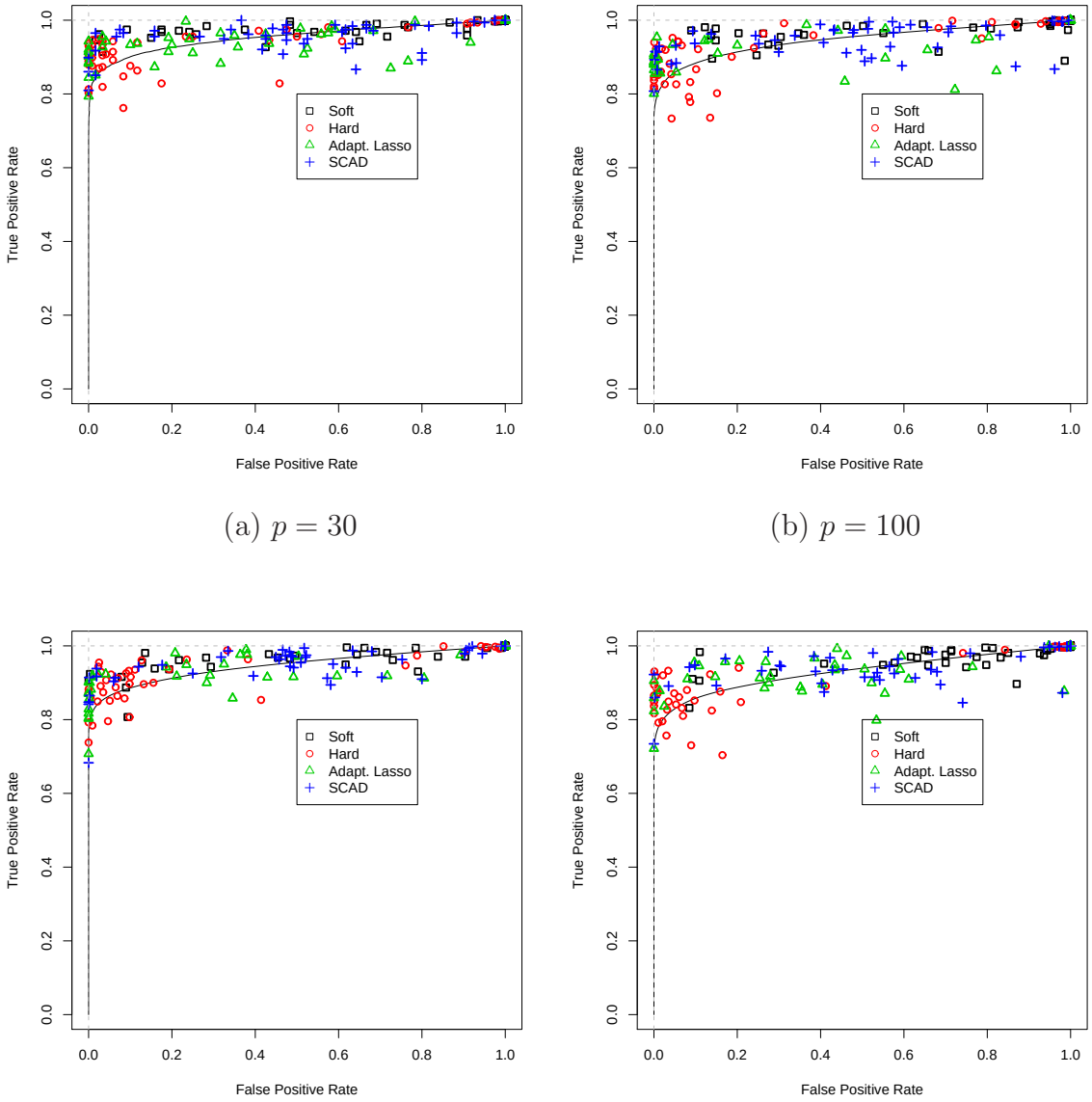
general be the case for less sparse models. However, the plot of realizations of TPR and FPR in Figure 3.3 shows that the variance is very high and there is no clear best choice for estimating the sparsity structure in this case.

Table 3.3: Average(SE) operator norm loss and true and false positive rates for Model 3 ($k = p/2$).

p	Sample	Hard	Soft	Adapt.lasso	SCAD
	Operator norm loss				
30	2.55(0.10)	2.40(0.10)	2.33(0.10)	2.34(0.09)	2.39(0.09)
100	8.67(0.37)	8.10(0.37)	8.05(0.39)	7.99(0.35)	8.11(0.36)
200	17.66(0.90)	16.81(0.85)	16.42(0.79)	16.21(0.75)	16.69(0.99)
500	43.71(2.01)	40.49(1.80)	42.75(1.87)	41.08(1.80)	40.60(1.79)
	TPR/FPR				
30	NA	0.92/0.26	0.98/0.69	0.94/0.45	0.95/0.51
100	NA	0.91/0.28	0.98/0.72	0.94/0.54	0.94/0.46
200	NA	0.92/0.35	0.97/0.69	0.94/0.49	0.95/0.51
500	NA	0.90/0.39	0.98/0.79	0.94/0.54	0.95/0.59

In Figure 3.4, we plot the average eigenspace agreement measure $K(q)$ defined in (3.20) versus q for $p = 200$ in all four models. For effectively sparser models AR(1) and MA(1), all thresholding methods improve on eigenspace estimation relative to the sample covariance, with SCAD and adaptive lasso showing the best performance. This effect is more pronounced for large p (plots not shown). For the less sparse triangular model, there is in fact no improvement relative to the covariance matrix, even though there is a slight improvement in overall operator norm loss. However, the eigenvalues corresponding to $q > 50$ here are very small, and thus the differences in eigenspaces are inconsequential. The biggest improvement in eigenspace estimation across models is for AR(1) with $\rho = 0.7$, which is consistent with our expectations that these methods perform best for models with many small or zero entries and few large entries well separated from 0.

Overall, the simulations show that in truly sparse models thresholding makes a big difference, and that penalties that combine the advantages of hard and soft thresholding, tend to perform best at recovering the true zeros. When the true model is not

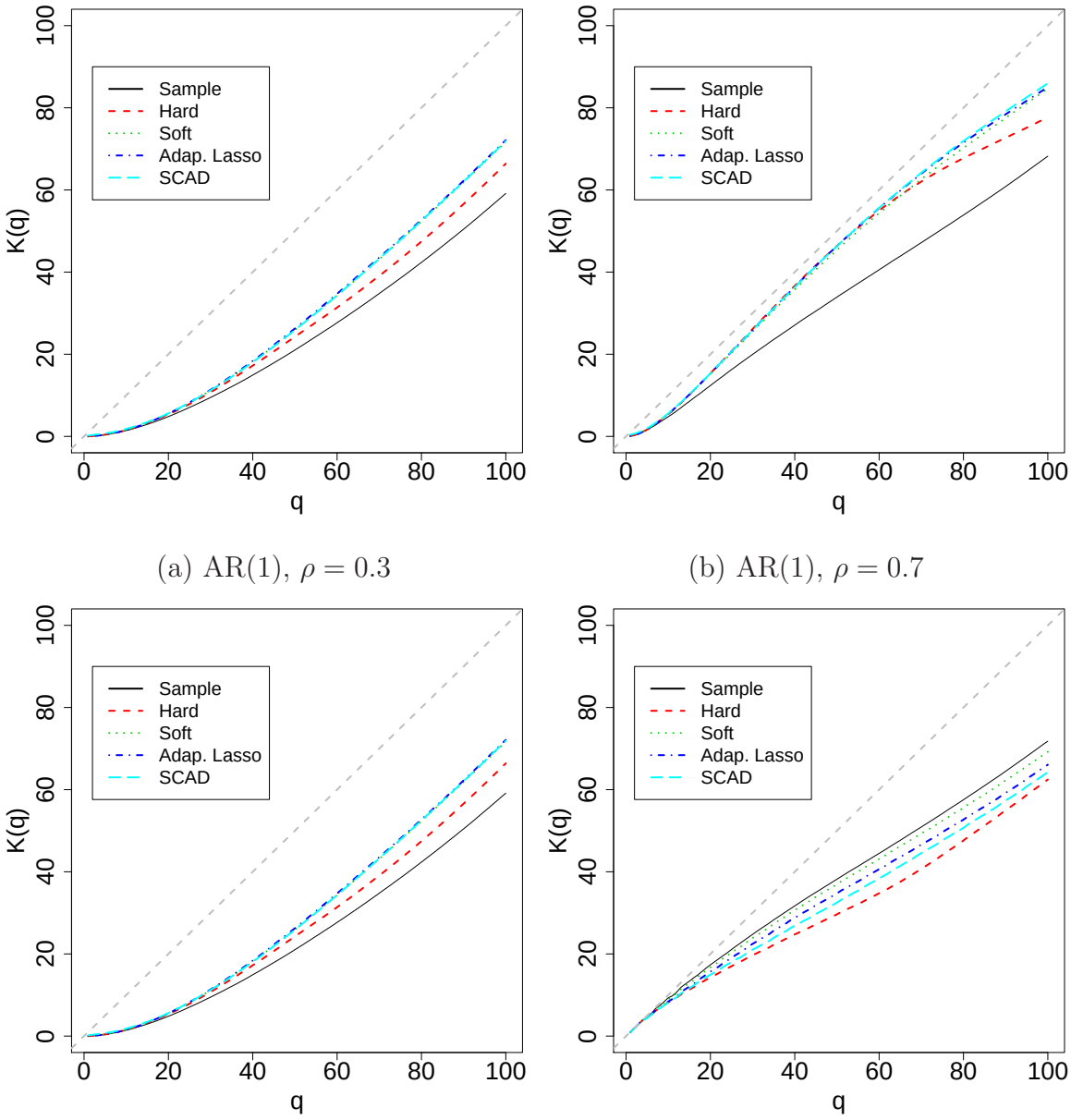


(a) $p = 30$ (b) $p = 100$

(c) $p = 200$ (d) $p = 500$

Figure 3.3: TPR vs. FPR for Model 3. The points correspond to 50 different realizations, with each method selecting its own threshold using validation data. The solid line is obtained by varying the threshold over the whole range (all methods have the same value of TPR and FPR for a fixed threshold).

sparse, the thresholded estimator does no worse than the sample covariance matrix, and thus in practice there does not seem to be any harm in applying thresholding even when there is little or no prior information about the degree of sparsity of the



(a) AR(1), $\rho = 0.3$

(b) AR(1), $\rho = 0.7$

(c) MA(1), $\rho = 0.3$

(d) Triangular, $k = p/2$

Figure 3.4: Average $K(q)$ versus q with $p = 200$.

true model.

3.5 Example: Gene clustering via correlations

Clustering genes using their correlations is a popular technique in gene expression data analysis (Eisen et al., 1998; Hastie et al., 2000). Here we investigate the effect of generalized thresholding on gene clustering using the data from a small round blue-cell tumors (SRBC) microarray experiment (Khan et al., 2001). The experiment had 64 training tissue samples, and 2308 gene expression values recorded for each sample. The original dataset included 6567 genes and was filtered down by requiring that each gene have a red intensity greater than 20 over all samples (for additional information, see Khan et al. (2001)). There are four types of tumors in the sample (EWS, BL-NHL, NB, and RMS).

First we ranked the genes by how much discriminative information they provide, using the F-statistic,

$$F = \frac{\frac{1}{k-1}\sum_{m=1}^{k} n_m(\bar{x}_m - \bar{x})^2}{\frac{1}{n-k}\sum_{m=1}^{k}(n_m - 1)\hat{\sigma}_m^2},$$

where $k = 4$ is the number of classes, $n = 64$ is the number of tissue samples, n_m is the number of tissue samples of class m, $\bar{x}_m$ and $\hat{\sigma}_m^2$ are the sample mean and variance of class m, and $\bar{x}$ is the overall mean. Then we selected top 40 and bottom 160 genes according to their F-statistics, so that we have both informative and non-informative genes. This selection was done to allow visualizing the correlation matrices via heatmaps.

We apply group average agglomerative clustering to genes using the estimated correlation in the dissimilarity measure,

$$d_{jj'} = 1 - |\hat{\rho}_{jj'}|, \tag{3.21}$$

where $\hat{\rho}_{jj'}$ is the estimated correlation between gene j and gene j'. We estimate the correlation matrix using hard, soft, adaptive lasso, and SCAD thresholding of the sample correlation matrix. The tuning parameter λ was selected via the resampling

scheme described in Bickel and Levina (2008b). The group-average agglomerative clustering is a bottom-up clustering method, which starts from treating all genes as singleton groups. Each step merges the two most similar groups, chosen to have the smallest average of pairwise dissimilarity between members of one group and the other. There are a total of $p - 1$ stages, and the last stage forms one group of size p. Figure 3.5 shows a heatmap of the data, with rows (genes) sorted by hierarchical clustering based on the sample correlations and columns (patients) sorted by tissue class for the 40 genes with the highest F-statistics, along with a heatmap of the sample correlations (absolute values) of the 40 genes ordered by hierarchical clustering. In all correlation heatmaps, we plot absolute values rather than the correlations themselves, because here we are interested in the strength of pairwise association between the genes regardless of its sign. It is clear that these 40 genes form strongly correlated blocks that correspond to different classes.

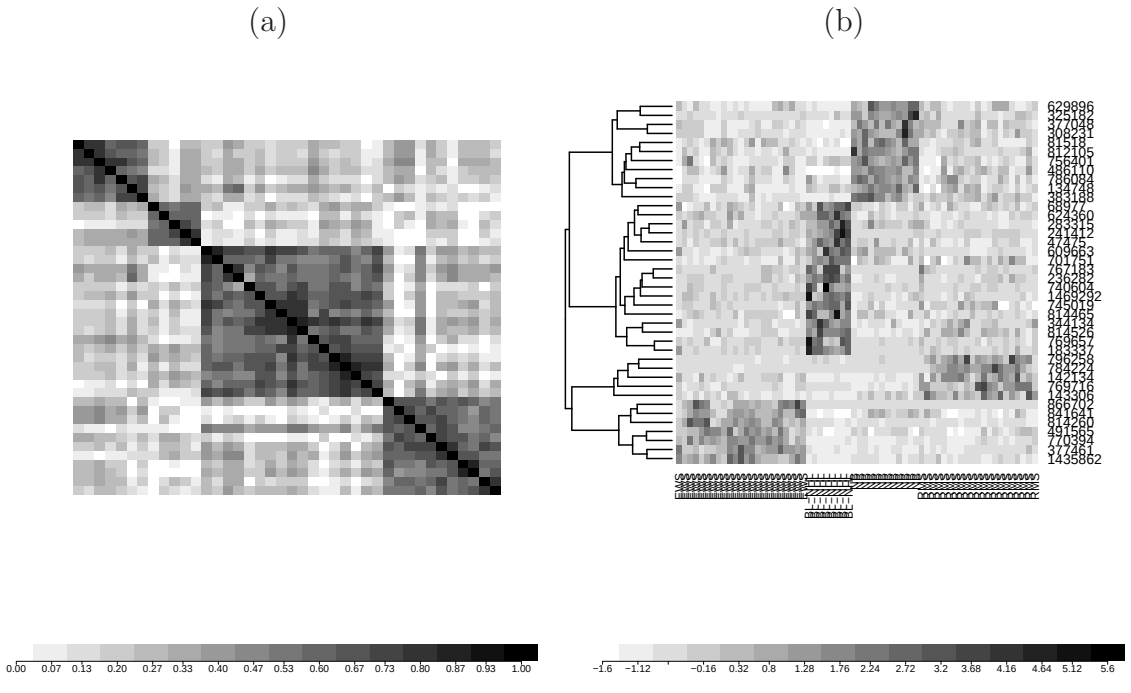


Figure 3.5: (a) Heatmap of the absolute values of sample correlations of the top 40 genes; (b) Heatmap of the gene expression data, with rows (genes) sorted by hierarchical clustering and columns sorted by tissue class.

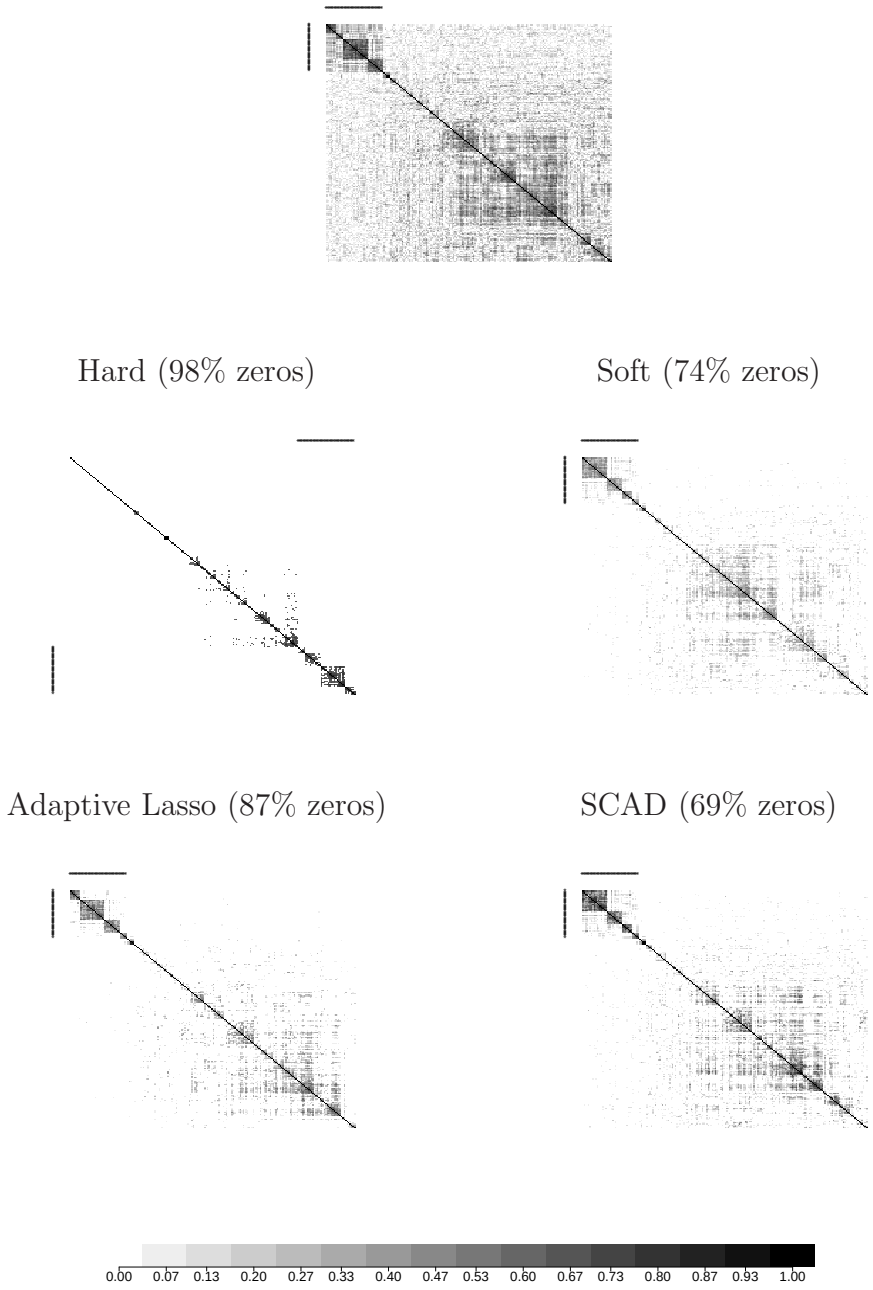


Figure 3.6: Heatmaps of the absolute values of estimated correlations. The 40 genes with the largest F-statistic are marked with stars. The genes are ordered by hierarchical clustering using estimated correlations. The percentage of off-diagonal elements estimated as zero is given in parentheses for each method.

The resulting heatmaps of the correlation matrix ordered by hierarchical clustering for each thresholding method are shown in Figure 3.6, along with the percentage of off-diagonal entries estimated as zero. Hard thresholding estimates many more zeros than other methods, resulting in a nearly diagonal estimator. This is consistent with hard thresholding results in simulations, where it tended to threshold too many entries, especially in higher dimensions. Also consistent with the simulation study is the performance of SCAD, which estimates the smallest number of zeros and appears to do a good job at cleaning up the signal without losing the block structure. As in simulations, adaptive lasso's performance is fairly similar to SCAD. This example confirms that using a combination of thresholding and shrinkage, which is more flexible than hard thresholding, results in a cleaner and more informative estimate of the sparsity structure.

CHAPTER IV

A new approach to Cholesky-based covariance regularization in high dimensions

4.1 Introduction

A large class of covariance estimators relies on the assumption that variables have a natural ordering, and those far apart in the ordering have small partial correlations. There are many applications that fall in this class, such as longitudinal data and spectroscopy, and exploiting the natural ordering present in the data in such cases leads to improved performance. The inverse estimators in this case usually rely on the modified Cholesky decomposition of the inverse covariance matrix, which is described in Section 4.2. This decomposition has a nice regression interpretation to which regularization can be applied more easily (Wu and Pourahmadi, 2003; Huang et al., 2006; Bickel and Levina, 2008b; Levina et al., 2008).

In this chapter we show that the modified Cholesky factor of the covariance matrix, rather than its inverse, also has a natural regression interpretation, and therefore all Cholesky-based regularization methods can be applied to the covariance matrix itself instead of its inverse to obtain a sparse estimator with guaranteed positive definiteness. As with all Cholesky-based regularization methods, this approach exploits the assumption of naturally ordered variables where variables far apart in the ordering tend to have small correlations. The simplest estimator in this new class is banding

the covariance Cholesky factor. Unlike banding the sample covariance matrix itself, it is guaranteed to be positive definite, but still has the same low computational complexity. We also derive some theoretical properties of banded estimators, connecting sparsity in a matrix to sparsity in its Cholesky factor and connecting banding Cholesky factors to constrained maximum likelihood.

4.2 Modified Cholesky decomposition of the covariance matrix

Throughout this chapter we assume that the data $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ are independent and identically distributed p-variate random vectors with population covariance matrix Σ and, without loss of generality, mean $\mathbf{0}$. Let $\hat{\Sigma}$ denote the sample covariance matrix, $\hat{\Sigma} = n^{-1}\sum_{i=1}^{n}(\boldsymbol{X}_i - \bar{\boldsymbol{X}})(\boldsymbol{X}_i - \bar{\boldsymbol{X}})^\intercal$. As a tool for regularizing the inverse covariance matrix, Pourahmadi (1999) and Wu and Pourahmadi (2003) suggested using the modified Cholesky factorization of Σ^{-1}. For a mean 0 random vector $X = (X^{(1)}, \ldots, X^{(p)})^\intercal$ with covariance matrix Σ, this factorization arises from regressing each variable $X^{(j)}$ on $X^{(j-1)}, \ldots, X^{(1)}$ that is, fitting regressions

$$X^{(j)} = \sum_{q=1}^{j-1}(-t_{jq})X^{(q)} + \epsilon^{(j)} = \hat{X}^{(j)} + \epsilon^{(j)} \ ,$$

where $\epsilon^{(j)}$ denotes the error term in regression j, $j = 2, \ldots, p$, and $\epsilon^{(1)} = X^{(1)}$. Let $\epsilon = (\epsilon^{(1)}, \ldots, \epsilon^{(p)})^\intercal$, let $D = \mathrm{var}(\boldsymbol{\epsilon})$ be the diagonal matrix of error variances, and let $T = (t_{jq})$ denote the lower-triangular matrix containing regression coefficients with the opposite sign, with ones on the diagonal. Then writing $\boldsymbol{\epsilon} = \boldsymbol{X} - \hat{\boldsymbol{X}} = T\boldsymbol{X}$ and using the fact that the errors are uncorrelated, $D = \mathrm{var}(\boldsymbol{\epsilon}) = \mathrm{var}(T\boldsymbol{X}) = T\Sigma T^\intercal$, and thus

$$\Sigma^{-1} = T^\intercal D^{-1} T. \tag{4.1}$$

This decomposition transforms inverse covariance matrix estimation into a regression problem, and hence regularization approaches for regression can be applied. If these regressions are not regularized, the resulting estimate is simply $\hat{\Sigma}^{-1}$. Banding the Cholesky factor of the inverse refers to regularizing by only including the immediate k predecessors in the regression, $X^{(j-k)}, \ldots, X^{(j-1)}$, for some fixed k (Wu and Pourahmadi, 2003; Bickel and Levina, 2008b).

The modified Cholesky factorization of Σ itself can be obtained from a latent variable regression model. Let $\Sigma = LDL^\intercal$ be the modified Cholesky decomposition of Σ, where D is diagonal and L is lower triangular with ones on the diagonal. Let $\boldsymbol{\epsilon}$ be a normal vector with independent components, $\boldsymbol{\epsilon} \sim N_p(\mathbf{0}, D)$. Then if we let $\boldsymbol{X} = L\boldsymbol{\epsilon}$, we have

$$\Sigma = \text{var}\ (L\boldsymbol{\epsilon}) = LDL^\intercal\ . \tag{4.2}$$

Our main interest here is in the regression interpretation. The vector $\boldsymbol{\epsilon}$ is unobserved, but because L is lower triangular, we can think of (4.2) as a sequence of regressions, where each variable $X^{(j)}$ is regressed on the previous regression errors $\epsilon^{(j-1)}, \ldots, \epsilon^{(1)}$. For $j = 2, \ldots, p$, we have

$$X^{(j)} = \sum_{q=1}^{j-1} l_{jq}\epsilon^{(q)} + \epsilon^{(j)} = \tilde{X}^{(j)} + \epsilon^{(j)}\ . \tag{4.3}$$

The decompositions above apply to the population matrices; Pourahmadi (2007) briefly mentions this decomposition for the population, but does not discuss any implications for estimation. Let $\mathcal{X}$ denote an n by p data matrix, where each column $\boldsymbol{x}_j \in \mathbb{R}^n$ is centered by its sample mean. For the first variable, we set $\boldsymbol{e}_1 = \boldsymbol{x}_1$. For $j = 2, \ldots, p$, let $\boldsymbol{l}_j = (l_{j1}, \ldots, l_{j,j-1})^\intercal$, $Z_j = (\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{j-1})$, and compute coefficients and the residual, respectively, as

$$\hat{\boldsymbol{l}}_j = \underset{\boldsymbol{l}_j}{\text{argmin}}\, \|\boldsymbol{x}_j - Z_j\boldsymbol{l}_j\|^2, \qquad\qquad \boldsymbol{e}_j = \boldsymbol{x}_j - Z_j\hat{\boldsymbol{l}}_j\ . \tag{4.4}$$

The variances are estimated as $\hat{d}_{jj} = n^{-1}\|\boldsymbol{e}_j\|^2$. Let Z denote the n by p matrix of residuals from carrying out the regressions in (4.3) sequentially. Here we assume that $p < n$ to ensure that all model matrices are of full column rank; Section 4.3 discusses the rank deficient case when $p \geq n$. Performing the regressions in (4.4) amounts to, for each $j = 2, \ldots, p$, orthogonally projecting the response $\boldsymbol{x}_j$ onto the span of $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{j-1}$ to estimate $\hat{\boldsymbol{l}}_j$. After the last projection we have an orthogonal basis $(\boldsymbol{e}_1, \ldots, \boldsymbol{e}_p)$, and the estimates $\hat{L}$ and $\hat{D}$. This algorithm is a scaled version of Gram–Schmidt orthogonalization of the data matrix $\mathcal{X}$ for computing its QR decomposition, where the upper triangular matrix R is restricted to have positive diagonal entries. The orthonormal matrix Q is the matrix Z with its column vectors scaled to have unit length and $R^\intercal = \hat{L}(n\hat{D})^{\frac{1}{2}}$. If all regressions are fitted by least squares, the resulting estimate recovers the sample covariance matrix: $\hat{\Sigma} = n^{-1}\mathcal{X}^\intercal\mathcal{X} = n^{-1}R^\intercal R = \hat{L}\hat{D}\hat{L}^\intercal$.

4.3 Regularized estimation of the Cholesky factor L

4.3.1 Banding the Cholesky factor

The simplest way to introduce sparsity in the Cholesky factor L is to estimate only the first k sub-diagonals of L and set the rest to zero. This approach for the inverse was proposed by Wu and Pourahmadi (2003) and Bickel and Levina (2008b). In our case, each variable $\boldsymbol{x}_j$ is regressed on the k previous residuals $\boldsymbol{e}_{j-k}, \ldots, \boldsymbol{e}_{j-1}$, for all $j = 2, \ldots, p$. The index $j - k$ is understood to mean $\max(1, j - k)$. Let $\boldsymbol{l}_j^{(k)} = (l_{j,j-k}, \ldots, l_{j,j-1})^\intercal$ and $Z_j^{(k)} = (\boldsymbol{e}_{j-k}, \ldots, \boldsymbol{e}_{j-1})$. Then we compute,

$$\hat{\boldsymbol{l}}_j^{(k)} = \underset{\boldsymbol{l}_j^{(k)}}{\operatorname{argmin}} \|\boldsymbol{x}_j - Z_j^{(k)}\boldsymbol{l}_j^{(k)}\|^2, \qquad \boldsymbol{e}_j = \boldsymbol{x}_j - Z_j^{(k)}\hat{\boldsymbol{l}}_j^{(k)} \ . \tag{4.5}$$

In each regression, the design matrix $Z_j^{(k)}$ has orthogonal columns, which allows (4.5) to be solved with at most k univariate regressions. Hence the computational cost of

banding the Cholesky factor in this manner is $O(kpn)$, the same order as banding the sample covariance matrix without the Cholesky decomposition. To ensure that design matrices are of full rank, the banding parameter k must be less than $\min(n-1, p)$. For sparse matrices, it is usually not necessary to search for values of $k \geq n$, since the optimal k is much smaller than n. We describe how to choose k in Section 4.4. If we do need to perform regressions when $k \geq n-1$, we use a generalized inverse of $Z_j^{(k)\intercal} Z_j^{(k)}$ for fitting ordinary least squares, in which case the resulting estimator is positive semi-definite.

Although each design matrix $Z_j^{(k)}$ has orthogonal columns, all of the residual vectors $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_p$ are not necessarily mutually orthogonal; $\boldsymbol{e}_j$ and $\boldsymbol{e}_{j'}$ are only guaranteed to be orthogonal if $|j - j'| \leq k$.

4.3.2 Connection to constrained maximum likelihood

Given that a Cholesky-based banded estimator is always positive definite, it is natural to ask whether it coincides with the maximum likelihood estimator under the banded constraint. Here we show that, somewhat surprisingly, banding the Cholesky factor of the inverse coincides with constrained maximum likelihood, and banding the Cholesky factor of the covariance matrix itself does not. First we establish some relationships between zero patterns in positive definite matrices and their Cholesky factors.

Proposition IV.1. *Let Σ and Ω be positive definite matrices with modified Cholesky decompositions $\Sigma = LDL^\intercal$ and $\Omega = T^\intercal D^{-1} T$, where L and T are both lower triangular. Then for any row i and $c(i) < i$, $\sigma_{i1} = \cdots = \sigma_{i,c(i)} = 0$ if and only if $l_{i1} = \cdots = l_{i,c(i)} = 0$; and for any column j and $r(j) > j$, $\omega_{p,j} = \cdots = \omega_{r(j),j} = 0$ if and only if $t_{p,j} = \cdots = t_{r(j),j} = 0$.*

Proof of Proposition IV.1. We prove the first claim only since the proof of the second one is very similar. From $\sigma_{ij} = \sum_{m=1}^{j} l_{im} l_{jm} d_{mm}$, it is obvious that $l_{i1} = \cdots = l_{i,c(i)} =$

0 implies $\sigma_{i1} = \cdots = \sigma_{i,c(i)} = 0$.

Now assume $\sigma_{i1} = \cdots = \sigma_{i,c(i)} = 0$ for some i. The formula for computing the modified Cholesky factorization L one column at a time, starting from the first column is given by, for $i > j$ (Watkins, 1991),

$$d_{ii} = \sigma_{ii} - \sum_{m=1}^{i-1} l_{im}^2 d_{mm}, \qquad l_{ij} = \frac{1}{d_{jj}} \left(\sigma_{ij} - \sum_{m=1}^{j-1} l_{im} l_{jm} d_{mm} \right) . \tag{4.6}$$

We proceed by induction: for the first column of L, $l_{i1} = \sigma_{i1}/\sigma_{11}$, hence $l_{i1} = 0$. Assuming that for a column $u < c(i)$ we have $l_{i1} = \cdots = l_{iu} = 0$, using (4.6) gives,

$$l_{i,u+1} = \frac{1}{d_{u+1,u+1}} \left(\sigma_{i,u+1} - \sum_{m=1}^{u} l_{im} l_{u+1,m} d_{u+1,u+1} \right) = \frac{\sigma_{i,u+1}}{d_{u+1,u+1}} ,$$

which implies $l_{i,u+1} = 0$. □

Proposition IV.1 is a simple matrix property, but we are not aware of a source to cite, so we give a proof in the Appendix for completeness. Proposition IV.1 implies that a covariance Cholesky factor with banded rows of arbitrary band lengths, not necessarily all the same, corresponds to a covariance matrix with banded rows of the same band lengths. On the other hand, the modified Cholesky factor of the inverse covariance matrix T with arbitrary column band lengths corresponds to an inverse covariance matrix Ω with the same column band lengths. In particular, the Cholesky factor of either the covariance matrix or the inverse is k-banded if and only if the corresponding matrix itself is k-banded.

Proposition IV.2. *Banding the modified Cholesky factor T of the inverse covariance matrix Ω maximizes the normal likelihood subject to the banded constraint, $\omega_{ij} = 0$ for $|i - j| > k$.*

Proof of Proposition IV.2. Let $\Omega_{(k)}$ be a symmetric positive definite matrix with k non-zero main sub-diagonals, $\omega_{(k)ij} = 0$ for $|i - j| > k$. The negative normal log-

likelihood up to a constant, as a function of the non-zero unique parameters in $\Omega_{(k)}$ is, $f(\Omega_{(k)}) = \mathrm{tr}(\hat{\Sigma}\Omega_{(k)}) - \log|\Omega_{(k)}|$. The k-banded constrained maximum likelihood estimator $\hat{\Omega}_{(k)}$ satisfies $\nabla f(\hat{\Omega}_{(k)}) = 0$. Let $T_{(k)}^{\intercal} D_{(k)}^{-1} T_{(k)} = \Omega_{(k)}$ be the modified Cholesky decomposition of $\Omega_{(k)}$. By Proposition IV.1, $t_{(k)ij} = 0$ for $|i-j| > k$. Let $g(T_{(k)}, D_{(k)}) \equiv f(T_{(k)}^{\intercal} D_{(k)}^{-1} T_{(k)})$, where g is a function of non-zero unique parameters in $(T_{(k)}, D_{(k)})$.

We continue by establishing that if $\nabla g(\hat{T}_{(k)}, \hat{D}_{(k)}) = 0$ then $\hat{T}_{(k)}^{\intercal} \hat{D}_{(k)}^{-1} \hat{T}_{(k)} = \hat{\Omega}_{(k)}$. Let $h(T_{(k)}, D_{(k)}) = T_{(k)}^{\intercal} D_{(k)}^{-1} T_{(k)}$. Denote the differential of h in the direction $u = (A_T, A_D)$ evaluated at $(T_{(k)}, D_{(k)})$, by $\nabla h(T_{(k)}, D_{(k)})[u]$. Then

$$\nabla h(T_{(k)}, D_{(k)})[u] = T_{(k)}^{\intercal} D_{(k)}^{-1} A_T + A_T^{\intercal} D_{(k)}^{-1} T_{(k)} - T_{(k)}^{\intercal} D_{(k)}^{-2} A_D T_{(k)} \ , \tag{4.7}$$

where A_T is written as a $p \times p$ matrix with non-zero entries in the same positions as the non-zero lower triangular entries in $T_{(k)}$, and A_D is written as a $p \times p$ diagonal matrix. Since the diagonal entries of $T_{(k)}$ are all equal to 1 and the diagonal entries of $D_{(k)}$ are positive, one can show by induction that $\nabla h(T_{(k)}, D_{(k)})[u] = 0$ implies $u = 0$. By the chain rule,$\nabla g(T_{(k)}, D_{(k)})[u] = \nabla f(T_{(k)}^{\intercal} D_{(k)}^{-1} T_{(k)})[u] \cdot \nabla h(T_{(k)}, D_{(k)})[u]$. Since f is convex with global minimizer $\hat{\Omega}_{(k)}$ it follows that $\nabla f(T_{(k)}^{\intercal} D_{(k)}^{-1} T_{(k)})[u] = 0$ if and only if $T_{(k)}^{\intercal} D_{(k)}^{-1} T_{(k)} = \hat{\Omega}_{(k)}$ unless $u = 0$. Hence we have that $\nabla g(T_{(k)}, D_{(k)})[u] = 0$ iff $\nabla f(T_{(k)}^{\intercal} D_{(k)}^{-1} T_{(k)})[u] = 0$ and $\hat{T}_{(k)}^{\intercal} \hat{D}_{(k)}^{-1} \hat{T}_{(k)} = \hat{\Omega}_{(k)}$.

Minimizing,

$$g(T_{(k)}, D_{(k)}) = \sum_{j=1}^{p} \left\{ n \log d_{(k)jj} + \sum_{i=1}^{n} \frac{1}{d_{(k)jj}} \left(x_{ij} + \sum_{v=j-k}^{j-1} t_{(k)jv} x_{iv} \right)^2 \right\},$$

where $\hat{\Sigma} = n^{-1} \mathcal{X}^{\intercal} \mathcal{X}$, is equivalent to minimizing,

$$g_j(t_{(k)j,j-k}, \ldots, t_{(k)j,j-1}, d_{(k)jj}) = n \log d_{(k)jj} + \sum_{i=1}^{n} \frac{1}{d_{(k)jj}} \left\{ x_{ij} - \sum_{v=j-k}^{j-1} (-t_{(k)jv}) x_{iv} \right\}^2 ,$$

for each row $j = 1, \ldots, p$. For row j, the solution to $\nabla g_j(\hat{t}_{(k)j,j-k}, \ldots, \hat{t}_{(k)j,j-1}, \hat{d}_{(k)jj}) = 0$, gives exactly the ordinary least squares regression coefficients with the opposite sign from regressing $\boldsymbol{x}_j$ on $\boldsymbol{x}_{j-k}, \ldots, \boldsymbol{x}_{j-1}$, and the sample variance of the n residuals from this fit. □

Proposition IV.3. *Banding the modified Cholesky factor L of the covariance matrix Σ does not maximize the normal likelihood under the constraint that $\sigma_{ij} = 0$ for $|i - j| > k$.*

Proof of Proposition IV.3. We show this by counter-example for $p = 3$. Let the function g be the negative normal log-likelihood parametrized by the inverse Cholesky factor $T = L^{-1}$ and D. Consider a 3×3 covariance matrix Σ with the banding constraint $\sigma_{31} = \sigma_{13} = 0$. This constraint is equivalent to $l_{31} = 0$ by Proposition IV.1. The unique parameters in the inverse Cholesky factor T in terms of the entries in the Cholesky factor L are: $t_{21} = -l_{21}$, $t_{31} = -l_{31} + l_{32}l_{21}$, and $t_{32} = -l_{32}$. Minimizing the negative log-likelihood subject to $l_{31} = 0$ is equivalent to minimizing the unconstrained function

$$\begin{aligned} b(l_{21}, l_{32}, D) =& n \sum_{j=1}^{3} \log d_{jj} + \frac{1}{d_{11}} \|\boldsymbol{x}_1\|^2 + \frac{1}{d_{22}} \|\boldsymbol{x}_2 - l_{21}\boldsymbol{x}_1\|^2 \\ &+ \frac{1}{d_{33}} \|\boldsymbol{x}_3 + l_{32}l_{21}\boldsymbol{x}_1 - l_{32}\boldsymbol{x}_2\|^2 \ . \end{aligned}$$

Since $\partial b(\hat{l}_{21}, \hat{l}_{32}, \hat{D})/\partial l_{21} = 2\hat{l}_{32}\boldsymbol{x}_1^\intercal \boldsymbol{x}_3 \hat{d}_{33}^{-1} \neq 0$ with probability 1, the Cholesky banding solution does not satisfy the first-order necessary condition for being an optimum of an unconstrained differentiable function b, and hence Cholesky banding does not maximize the constrained normal likelihood. □

Intuitively, the constrained maximum likelihood result holds for the inverse only because the inverse is the canonical parameter of the normal likelihood. The constrained maximum likelihood estimator of the covariance matrix can be computed by

the algorithm proposed by Chaudhuri et al. (2007), but this algorithm only works for $p < n$. We are not aware of suitable constrained maximum likelihood estimation algorithms for $p > n$, which makes banding the Cholesky factor a more attractive option for computing a positive definite estimator for large p. In Section 4.4, we briefly compare the numerical performance of banding the Cholesky factor of covariance to the constrained maximum likelihood estimator when $p < n$, and find that the two estimators are in practice very close, even though they differ theoretically.

4.3.3 The penalized regression approach

Once we have the regression interpretation (4.3), all penalty-based approaches proposed for regularizing the inverse become equally applicable to the covariance matrix itself. In general, we can estimate the Cholesky factor by,

$$\hat{\boldsymbol{l}}_j = \underset{\boldsymbol{l}_j}{\operatorname{argmin}}\{\|\boldsymbol{x}_j - Z_j\boldsymbol{l}_j\|^2 + P_\lambda(\boldsymbol{l}_j)\}. \tag{4.8}$$

Penalty functions P_λ that encourage sparsity in the coefficient vector $\boldsymbol{l}_j$ are of particular interest. Huang et al. (2006) applied the lasso penalty in the inverse covariance Cholesky estimation problem, and here we can analogously use

$$P_\lambda^L(\boldsymbol{l}_j) = \lambda \sum_{t=1}^{j-1} |l_{jt}|.$$

The lasso penalty function can result in zeros in arbitrary locations in the Cholesky factor, which may or may not lead to any zeros in the resulting covariance matrix. To impose additional structure, Levina et al. (2008) proposed the nested lasso penalty, which in our context is given by,

$$P_\lambda^{NL}(\boldsymbol{l}_j) = \lambda \left(|l_{j,j-1}| + \frac{|l_{j,j-2}|}{|l_{j,j-1}|} + \frac{|l_{j,j-3}|}{|l_{j,j-2}|} + \cdots + \frac{|l_{j,1}|}{|l_{j,2}|} \right) , \tag{4.9}$$

where $0/0$ is defined as 0. This penalty imposes the restriction that $l_{jt} = 0$ if $l_{j,t+1} = 0$. By Proposition IV.1, this means that all the zeros estimated in the Cholesky factor of covariance $\hat{L}$ will be preserved in $\hat{\Sigma}$. This is not the case in the inverse Cholesky decomposition for which this penalty was originally proposed by Levina et al. (2008), although some zeros are preserved in that case as well. In practice, Levina et al. (2008) recommend using a slightly modified version of (4.9), where the first term is divided by the univariate regression coefficient from regressing $\boldsymbol{x}_j$ on $\boldsymbol{e}_{j-1}$ alone, to address a potential difference of scales, which is the version we used in simulations. Both lasso and nested lasso have much higher computational cost than banding, and are not appropriate for very large p; however, the additional flexibility of the sparsity structure of nested lasso's variable band widths may work well in some cases.

4.4 Numerical results

4.4.1 Simulation Settings

Our simulation study compares the performance of all the covariance estimators discussed in Section 4.3, banding the sample covariance matrix directly (Bickel and Levina, 2008b), which is not positive definite, and, as a benchmark, the shrinkage estimator of Ledoit and Wolf (2003) which does not depend on the order of variables. The Ledoit–Wolf estimator is a linear combination of the identity matrix and the sample covariance matrix, with coefficients optimal in a certain sense; it does not introduce any sparsity.

We consider the following three covariance structures: Σ_1 has entries $\sigma_{ij} = \rho^{|i-j|}$ with $\rho = 0.7$, Σ_2 has entries $\sigma_{ij} = \mathbb{1}(i = j)+0.4\ \mathbb{1}(|i - j| = 1)+0.2\ \mathbb{1}(2 \leq |i - j| \leq 3)+0.1\ \mathbb{1}(|i - j| = 4)$, and Σ_3 has entries $\sigma_{ij} = \mathbb{1}(i = j)+ 0.5\ \mathbb{1}(i \neq j)$. The first order autoregressive model Σ_1 has a dense Cholesky factor, but its entries decay as one moves away from the diagonal. We only report results for $\rho = 0.7$, but the same

pattern is observed over the whole range of ρ. The fourth order moving average model Σ_2 is a banded matrix with $k = 4$, and therefore its Cholesky factor is also 4-banded. The model Σ_1 was considered by Bickel and Levina (2008b), and Σ_2 by Yuan and Lin (2007). Model Σ_3 is a full matrix, where introducing sparsity cannot improve estimation, and thus we expect the regularization methods to perform similarly to the covariance matrix.

We generate $n = 100$ training observations and another 100 independent validation observations from $N_p(\mathbf{0}, \Sigma)$, with $p = 30$, 100, 200, 500, and 1000. Lasso and nested lasso were not run for $p \geq 500$ due to their high computational cost. Tuning parameters were selected by minimizing the Frobenius norm, $\|M\|_F^2 = \sum_{i,j} m_{ij}^2$, of the difference between the regularized estimate computed with the training observations and the sample covariance computed with the validation observations. The results are not sensitive to the choice of loss, we have also tested matrix 1-norm and matrix 2-norm losses and obtained very similar results, and we selected the Frobenius norm because it had a very slight edge in simulations and because there are general theoretical results justifying cross-validation via Frobenius norm (Bickel and Levina, 2008a). The whole process was repeated 200 times.

To compare estimators, we used the operator norm, also known as the matrix 2-norm, $\|M\|^2 = \lambda_{\max}(MM^\intercal)$, of the difference between the covariance estimator and the truth, $\Delta(\hat{\Sigma}, \Sigma) = \|\hat{\Sigma} - \Sigma\|$. This loss is commonly used to assess covariance estimators because convergence in this norm implies convergence of all eigenvectors and eigenvalues. Other losses such as Frobenius norm, matrix 1-norm, and entropy loss are omitted to save space; they produce very similar results.

We also compute the true positive rate and true negative rate, defined respectively

as

$$\mathrm{TPR}(\hat{\Sigma}, \Sigma) = \frac{\#\{(i,j) : \hat{\sigma}_{ij} \neq 0 \quad \text{and} \quad \sigma_{ij} \neq 0\}}{\#\{(i,j) : \sigma_{ij} \neq 0\}}\,, \tag{4.10}$$

$$\mathrm{TNR}(\hat{\Sigma}, \Sigma) = \frac{\#\{(i,j) : \hat{\sigma}_{ij} = 0 \quad \text{and} \quad \sigma_{ij} = 0\}}{\#\{(i,j) : \sigma_{ij} = 0\}}\,. \tag{4.11}$$

The sample covariance has $\mathrm{TPR}(\hat{\Sigma}, \Sigma) = 1$, and a diagonal estimator has $\mathrm{TNR}(\hat{\Sigma}, \Sigma) = 1$.

4.4.2 Results

The averages and standard errors over 200 replications of the operator norm loss for the three models are given in Table 4.1. For models Σ_1 and Σ_2, where the true Cholesky factor is either banded or has entries decaying fast as one goes away from the diagonal, banding the Cholesky factor provides the best performance in every case. In particular, it outperforms banding the sample covariance directly, particularly in high dimensions, presumably due to its ability to enforce positive definiteness. Both banding methods outperform the Ledoit–Wolf estimator, which is not sparse at all, and lasso applied to the Cholesky factor, which cannot create a banded structure and loses sparsity in the matrix itself. The nested lasso does have the ability to create a banded structure in the Cholesky factor, but its extra flexibility, not needed for these models, leads to noisier estimates in this case. As expected, the margin by which sparse regularized estimators outperform non-sparse estimators, the sample and Ledoit–Wolf, is larger for the sparse population covariance Σ_2. For the full matrix Σ_3, introducing sparsity cannot help, and thus all sparse estimators, excluding nested lasso, perform similarly to the sample covariance. The Ledoit–Wolf estimator is very close to the sample covariance because one eigenvalue of Σ_3 is very large relative to others, which makes the coefficient of the sample covariance term very close to 1. Nested lasso has a large risk for $p = 200$ because it can only estimate up to $n - 1$

Table 4.1: Averages and standard errors of the operator norm loss for the sample covariance, Ledoit–Wolf's estimator, the banded sample covariance, and regularization of Cholesky factor of the covariance by banding, lasso, and nested lasso.

p	Samp.	Ledoit–Wolf	Samp. Band.	Chol. Band.	Lasso	Nes. Lasso
			Σ_1			
30	1.82	1.70	1.31	1.30	1.73	1.47
	(0.03)	(0.02)	(0.02)	(0.02)	(0.02)	(0.02)
100	4.10	3.10	1.61	1.61	3.53	1.83
	(0.04)	(0.01)	(0.02)	(0.02)	(0.01)	(0.02)
200	6.59	3.83	1.77	1.76	3.91	1.97
	(0.04)	(0.01)	(0.02)	(0.01)	(0.01)	(0.01)
500	12.47	4.43	1.96	1.91	–	–
	(0.04)	(0.00)	(0.02)	(0.01)		
1000	20.64	4.64	2.08	2.01	–	–
	(0.04)	(0.00)	(0.02)	(0.01)		
			Σ_2			
30	1.44	1.14	0.76	0.74	1.24	0.87
	(0.02)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)
100	3.27	1.63	0.92	0.89	1.63	1.03
	(0.02)	(0.00)	(0.01)	(0.01)	(0.00)	(0.01)
200	5.33	1.77	1.00	0.95	1.72	1.08
	(0.02)	(0.00)	(0.01)	(0.01)	(0.00)	(0.01)
500	10.37	1.84	1.09	1.06	–	–
	(0.03)	(0.00)	(0.01)	(0.01)		
1000	17.58	1.85	1.17	1.14	–	–
	(0.03)	(0.00)	(0.01)	(0.01)		
			Σ_3			
30	2.62	2.64	2.63	2.69	2.62	2.68
	(0.07)	(0.07)	(0.07)	(0.07)	(0.07)	(0.07)
100	8.83	8.86	8.83	8.88	8.82	8.86
	(0.22)	(0.23)	(0.22)	(0.23)	(0.22)	(0.23)
200	17.63	17.85	17.63	17.73	17.62	68.11
	(0.43)	(0.44)	(0.43)	(0.43)	(0.43)	(0.67)
500	44.58	44.77	44.58	44.62	–	–
	(1.25)	(1.29)	(1.25)	(1.25)		
1000	86.62	87.13	86.62	86.69	–	–
	(2.43)	(2.47)	(2.43)	(2.43)		

non-zeros in any row of the Cholesky factor in a band extending from the diagonal. The covariance Cholesky factor of Σ_3 is $l_{ij} = (j+1)^{-1} I(i > j) + I(i = j)$ and thus

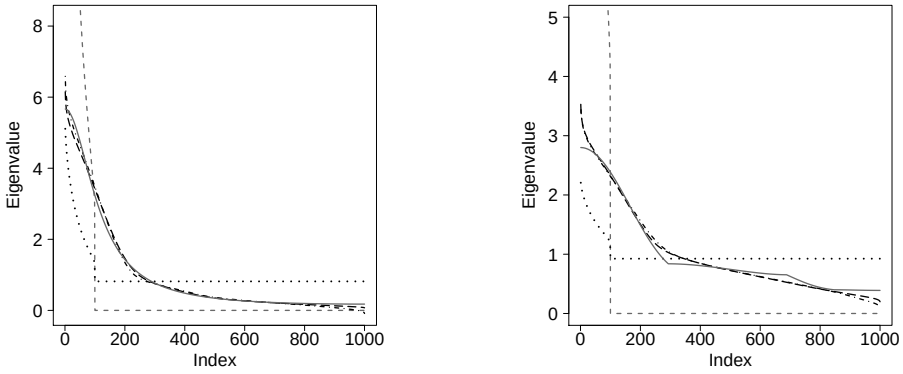
the true row coefficients are the smallest in a band extending from the diagonal. The lasso is also only able to estimate up to $n-1$ non-zeros in any row of the Cholesky factor; however, these non-zeros could be estimated in any location, and estimating most of the first $n-1$ coefficients in a row as non-zero is enough to come close to the sample covariance in this model.

The banded maximum likelihood estimator was also computed using the algorithm of Chaudhuri et al. (2007) for $p = 30$, since the algorithm is only applicable when $p < n$. Its loss values are 1.32(0.02) for Σ_1, 0.74(0.01) for Σ_2, and 2.77(0.07) for Σ_3, which are essentially the same as those for Cholesky banding for $p = 30$.

For the sparse matrix Σ_2, we also report true positive and true negative rates of estimating zeros in Table 4.2. Both Cholesky banding and sample covariance banding have nearly perfect true negative rates, but banding the Cholesky factor has a better true positive rate than for banding the sample, which means that banding the sample tends to set more sub-diagonals to zero than necessary. The lasso method has a low true negative rate because zeros in the Cholesky factor are not preserved in the matrix, and the nested lasso does reasonably well on both but not as well as Cholesky banding.

Table 4.2: Averages and standard errors of true positive/true negative percentages for Σ_2, based on 200 replications.

p	Sample Band.	Cholesky Band.	Lasso	Nested Lasso
30	87.47/100.00	90.19/99.69	99.71/3.90	94.07/89.06
	(0.84)/(0.00)	(0.83)/(0.12)	(0.05)/(0.27)	(0.32)/(0.46)
100	88.31/100.00	93.72/99.99	90.38/37.35	93.82/97.25
	(0.87)/(0.00)	(0.76)/(0.01)	(0.14)/(0.21)	(0.20)/(0.08)
200	87.22/100.00	93.92/100.00	90.59/34.93	94.11/98.69
	(0.88)/(0.00)	(0.76)/(0.00)	(0.10)/(0.14)	(0.14)/(0.03)
500	85.42/100.00	96.51/100.00	–	–
	(0.87)/(0.00)	(0.61)/(0.00)	–	–
1000	85.77/100.00	98.13/100.00	–	–
	(0.88)/(0.00)	(0.47)/(0.00)	–	–



Σ_1 Σ_2

Figure 4.1: Scree plots for the sample covariance (gray dashes), Ledoit–Wolf (dots), banding the sample covariance (dash-dot), Cholesky banding (black dashes), and the truth (solid) for $p = 1000$, averaged over 200 replications.

In Fig. 4.1 we plot the average estimated eigenvalues in descending order for sample banding, Cholesky banding, the sample covariance, and the Ledoit–Wolf estimator, as well as the true eigenvalues, for both models with $p = 1000$. Since $n = 100$, the sample covariance matrix only has 99 non-zero eigenvalues. Cholesky banding and sample banding perform similarly for both models, with Cholesky banding having a slight edge for the small eigenvalues. The banding methods outperform both the sample covariance and the Ledoit–Wolf estimator by a considerable amount, especially for larger eigenvalues. This is expected since the banding methods performed best under the operator norm loss, and the truth is banded or almost banded. For Σ_3, the plots are indistinguishable and are omitted to save space.

Since sample covariance banding does not necessarily produce a positive definite estimator, we also report the percentage of estimates that are positive definite in Table 4.3. It is clear that larger p and denser truth make it harder to keep positive definiteness.

Table 4.3: Percentage of positive definite banded sample realizations

p	30	100	200	500	1000
Σ_1	61.5	16.5	3.0	0.0	0.0
Σ_2	100.0	100.0	99.5	98	98.0
Σ_3	98.0	4.0	0.0	0.0	0.0

4.5 Sonar data example

In this section we illustrate the effects of Cholesky banding and sample covariance banding on SONAR data from the UCI machine learning data repository, available at http://www.ics.uci.edu/~mlearn/MLRepository.html. This dataset has 111 spectra from metal cylinders and 97 spectra from rocks, where each spectrum has 60 frequency band energy measurements. These spectra were measured at multiple angles for the same objects, but following previous analyses of the dataset we assume independence of the spectra.

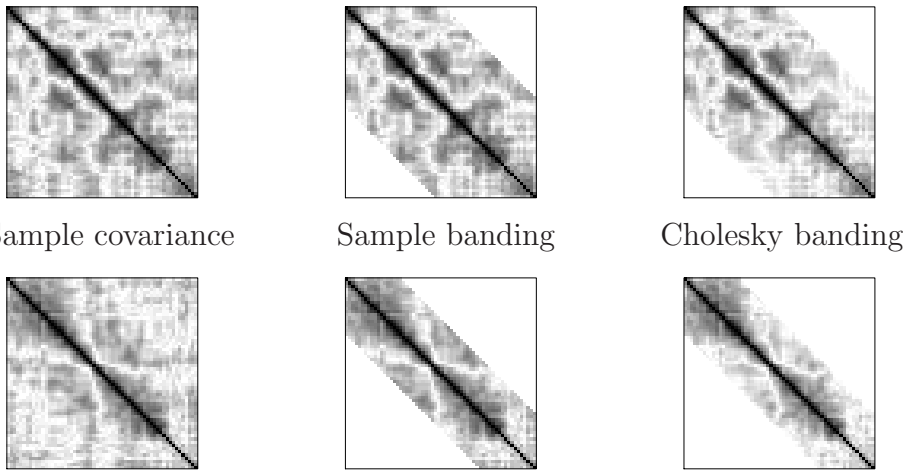
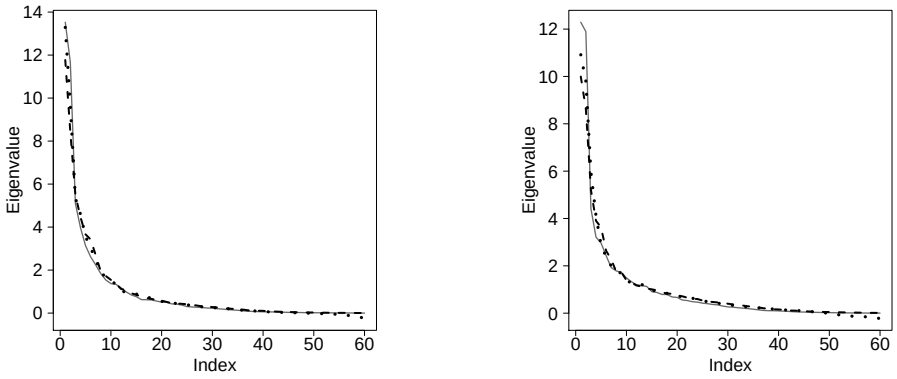


Figure 4.2: Heatmaps of the absolute values of entries in the correlation matrix estimates, where a correlation of magnitude 0 is white and a correlation of magnitude 1 is black. The top row is for metal spectra and the bottom row is for rock spectra.

The top panel of Fig. 4.2 shows heatmaps of the absolute values of the sample correlation matrices for metal and rock, where we standardize the variables first to facilitate comparison for metal and rock spectra, which are on different scales. Both



(a) (b)

Figure 4.3: Scree plots of the sample covariance (solid), sample banding (dots), and Cholesky banding (dashes) for the metal spectra in panel (a) and for the rock spectra in panel (b).

matrices show a general pattern of correlations decaying away from the diagonal, which makes banding a reasonable option.

The banding parameter k for both banding methods was selected using the random-splitting scheme of Bickel and Levina (2008b),

$$\hat{k} = \underset{k}{\operatorname{argmin}} N^{-1} \sum_{v=1}^{N} \|\hat{\Sigma}_{(k)}^{(v)} - \tilde{\Sigma}^{(v)}\|_F \ ,$$

where $\hat{\Sigma}_{(k)}^{(v)}$ is the banded estimator with k bands computed on the training data, and $\tilde{\Sigma}^{(v)}$ is the sample covariance of the validation data. To obtain these training and validation sets, the data was split at random $N = 100$ times, with 1/3 of the sample used for training. For metal, Cholesky banding and sample banding both chose $\hat{k} = 31$ sub-diagonals; for rock, Cholesky banding chose $\hat{k} = 17$ and sample banding chose $\hat{k} = 18$. Since these values are so close, for easier visual comparison we show both with $\hat{k} = 17$ for the rock spectra. The heatmaps of the absolute values of correlations from the banded estimators are shown in Fig. 4.2. We see that Cholesky banding shrinks the non-zero correlations whereas the sample banding does not, which is the property that allows Cholesky banding to achieve positive definiteness.

We also show eigenvalue plots for these estimators in Fig. 4.3(a) and (b). We see that the sample covariance has the most spread out eigenvalues, and the eigenvalues from Cholesky banding have the least spread, as we would expect.

We also compared the performance of the various estimators if they are used in quadratic discriminant analysis to discriminate between rock and metal. An observation $\boldsymbol{x}$ is classified as rock $j = 0$ or metal $j = 1$ using the rule, $G(\boldsymbol{x}) = \text{argmax}_j\{\log|\hat{\Omega}_j|/2 - (\boldsymbol{x} - \hat{\boldsymbol{\mu}}_j)^\intercal\hat{\Omega}_j(\boldsymbol{x} - \hat{\boldsymbol{\mu}}_j)/2 + \log\hat{\pi}_j\}$, where $\hat{\pi}_j$ is the proportion of class j observations, $\hat{\boldsymbol{\mu}}_j$ is the class j sample mean, and $\hat{\Omega}_j$ is the inverse covariance estimate for class j, all computed on the training data. More details can be found in Mardia et al. (1979). In addition to banding the Cholesky factor of covariance and of the inverse, we also added a diagonal estimator of the covariance matrix, which corresponds to the naive Bayes classifier. Banding the sample covariance was omitted because it is not invertible. Leave-one-out cross validation was used to estimate the testing error, and the banding parameters were selected with 10 random splits with 1/3 of the data used for training, using Frobenius loss for covariance Cholesky banding and the validation likelihood for the inverse covariance Cholesky banding. The test errors were 24.0% for the sample covariance, 32.7% for naive Bayes, 20.2% for covariance Cholesky banding, and 14.9% for inverse Cholesky banding. Both banding methods are substantially better than either estimating the whole dependency structure by the sample covariance or not estimating it at all with naive Bayes. We conjecture the inverse Cholesky banding does better because it introduces sparsity directly in the inverse.

4.6 Discussion

In terms of convergence rates, one would expect a convergence result analogous to the one for inverse Cholesky banding established by Bickel and Levina (2008b) to hold here as well, but this case presents substantial extra technical difficulties in

analysis, due to the fact that the errors used as predictors in the regressions required to compute the Cholesky factor are unobservable and have to be estimated by residuals. Nonetheless, we expect the method to be equally useful based on its good practical performance.

The regression representation of the covariance matrix and its inverse have obvious parallels with time series models for moving average and autoregressive processes, respectively. However, we do not fit a parametric model here, and do not assume stationarity, which would correspond to imposing a Toeplitz structure on the matrix, and thus fitting and model selection methods are very different from time series. As a rule of thumb in practice, if it is not clear from the problem whether it is preferable to regularize the covariance or the inverse, we would recommend fitting both and choosing the sparser estimate.

CHAPTER V

Sparse multivariate regression with covariance estimation

5.1 Introduction

Multivariate regression is a generalization of the classical regression model of regressing a single response on p predictors to regressing $q > 1$ responses on p predictors. Applications of this general model arise in chemometrics, econometrics, psychometrics, and other quantitative disciplines where one predicts multiple responses with a single set of prediction variables. For example, predicting several measures of quality of paper with a set of variables relating to its production, or predicting asset returns for several companies using the vector auto-regressive model (Reinsel, 1997), both result in multivariate regression problems.

Let $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})^T$ denote the predictors, let $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{iq})^T$ denote the responses, and let $\boldsymbol{\epsilon}_i = (\epsilon_1, \ldots, \epsilon_q)^T$ denote the errors, all for the ith sample. The multivariate regression model is given by,

$$\boldsymbol{y}_i = B^T \boldsymbol{x}_i + \boldsymbol{\epsilon}_i, \qquad \text{for} \quad i = 1, \ldots, n,$$

where B is a $p \times q$ regression coefficient matrix and n is the sample size. Column k of B is the regression coefficient vector from regressing the kth response on the predictors.

We make the standard assumption that $\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_n$ are i.i.d $N_q(0, \Sigma)$. Thus, given a realization of the predictor variables, the covariance matrix of the response variables is Σ. This assumption of correlated errors suggests that separately estimating each column of B by performing q separate regressions may be inferior to jointly estimating all columns of B, accounting for the correlated errors.

The model can be expressed in matrix notation. Let X denote the $n \times p$ predictor matrix where its ith row is $\boldsymbol{x}_i^T$, let Y denote the $n \times q$ random response matrix where its ith row is $\boldsymbol{y}_i^T$, and let E denote the $n \times q$ random error matrix where its ith row is $\boldsymbol{\epsilon}_i^T$, then the model is,

$$Y = XB + E.$$

Note that if $q = 1$, the model simplifies to the classical regression problem where B is a p dimensional regression coefficient vector. For simplicity of notation we assume that columns of X and Y have been centered and thus the intercept terms are omitted.

The negative log-likelihood function of (B, Ω), where $\Omega = \Sigma^{-1}$, can be expressed up to a constant as,

$$g(B, \Omega) = \operatorname{tr}\left[\frac{1}{n}(Y - XB)^T(Y - XB)\Omega\right] - \log|\Omega|. \tag{5.1}$$

The maximum likelihood estimator for B is simply $\hat{B}^{\text{OLS}} = (X^TX)^{-1}X^TY$, which amounts to performing separate ordinary least squares estimates for each of the q response variables and does not depend on Ω.

Prediction with the multivariate regression model requires the estimation of pq parameters which becomes challenging when there are many predictors and responses. Criterion-based model selection has been extended to multivariate regression by Fujikoshi and Satoh (1997) and Bedrick and Tsai (1994). For a review of Bayesian approaches for model selection and prediction with the multivariate regression model see Brown et al. (2002) and references therein. A dimensionality reduction approach

called reduced-rank regression (Anderson, 1951; Izenman, 1975; Reinsel and Velu, 1998) minimizes (5.1) subject to $\text{rank}(B) \leq r$ for some $r \leq \min(p, q)$. The solution involves canonical correlation analysis, which amounts to finding r uncorrelated linear combinations of the predictors $X\boldsymbol{u}_1, \ldots, X\boldsymbol{u}_r$ to predict r uncorrelated linear combinations of the responses $Y\boldsymbol{v}_1, \ldots, Y\boldsymbol{v}_r$ so that the squared correlations between predictor canonical variates $X\boldsymbol{u}_k$ and response canonical variates $Y\boldsymbol{v}_k$ are maximized in order from $k = 1, \ldots, r$. Thus the method combines information from all of the q response variables into r canonical response variates that have the highest canonical correlation with the corresponding predictor canonical variates. As in the case of principal components regression, the interpretation of the reduced rank model is typically impossible in terms of the original predictors and responses.

Other approaches aimed at reducing the number of parameters in the coefficient matrix B involve solving,

$$\hat{B} = \underset{B}{\text{argmin}} \, \text{tr}\left[(Y - XB)^T(Y - XB)\right] \quad \text{subject to:} \quad C(B) \leq t, \tag{5.2}$$

where $C(B)$ is some constraint function. A method called factor estimation and selection (FES) was proposed in Yuan et al. (2007), who apply the constraint function $C(B) = \sum_{j=1}^{\min(p,q)} \sigma_j(B)$, where $\sigma_j(B)$ is the jth singular value of B. This constraint encourages sparsity in the singular values of $\hat{B}$, and hence reduces the rank of $\hat{B}$; however, unlike reduced rank regression, FES offers a continuous regularization path. A novel approach for imposing sparsity in the entries of $\hat{B}$ was taken by Turlach et al. (2005), who proposed the constraint function, $C(B) = \sum_{j=1}^{p} \max(|b_{j1}|, \ldots, |b_{jq}|)$. This method was recommended for model selection (sparsity identification), and not for prediction because of the bias of the L_∞-norm penalty. Imposing sparsity in $\hat{B}$ for the purposes of identifying "master predictors" was proposed by Peng et al. (2009), who applied a combined constraint function $C(B) = \lambda C_1(B) + (1 - \lambda)C_2(B)$ for

$\lambda \in [0, 1]$, where $C_1(B) = \sum_{j,k} |b_{jk}|$, the lasso constraint (Tibshirani, 1996) on the entries of B and $C_2(B) = \sum_{j=1}^{p} (b_{j1}^2 + \cdots + b_{jq}^2)^{0.5}$, the sum of the L_2-norms of the rows of B. The first constraint introduces sparsity in the entries of $\hat{B}$ and the second constraint introduces zeros for all entires in some rows of $\hat{B}$, meaning that some predictors are irrelevant for all q responses. Asymptotic properties for an estimator using this constraint with $\lambda = 0$ have also been established (Obozinski et al., 2008). This combined constraint approach provides highly interpretable models in terms of the prediction variables. However, all of the methods above that solve (5.2) do not account for correlated errors.

To directly exploit the correlation in the response variables to improve prediction performance, a method called Curds and Whey (C&W) was proposed by Breiman and Friedman (1997). C&W predicts the multivariate response with an optimal linear combination of the ordinary least squares predictors. The C&W linear predictor has the form $\tilde{Y} = \hat{Y}^{\mathrm{OLS}} M$, where M is a $q \times q$ shrinkage matrix estimated from the data. This method exploits correlation in the responses arising from shared random predictors as well as correlated errors.

In this chapter, we propose a method that combines some of the strengths of the estimators discussed above to improve prediction in the multivariate regression problem while allowing for interpretable models in terms of the predictors. We reduce the number of parameters using the lasso penalty on the entries of B while accounting for correlated errors. We accomplish this by simultaneously optimizing (5.1) with penalties on the entries of B and Ω. We call our new method multivariate regression with covariance estimation (MRCE). The method assumes predictors are not random; however, the resulting formulas for the estimates would be the same with random predictors. Our focus is on the conditional distribution of Y given X and thus, unlike in the Curds and Whey framework, the correlation of the response variables arises only from the correlation in the errors.

We also note that the use of lasso penalty on the entries of Ω has been considered by several authors in the context of covariance estimation (Yuan and Lin, 2007; d'Aspremont et al., 2008; Rothman et al., 2008; Friedman et al., 2008). However, here we use it in the context of a regression problem, thus making it an example of what one could call *supervised* covariance estimation: the covariance matrix here is estimated in order to improve prediction, rather than as a stand-alone parameter. This is a natural next step from the extensive covariance estimation literature, which has been given surprisingly little attention to date; one exception is the joint regression approach of Witten and Tibshirani (2009). Another less directly relevant example of such supervised estimation is the supervised principal components by Bair et al. (2006).

The remainder of the chapter is organized as follows: Section 5.2 describes the MRCE method and associated computational algorithms, Section 5.3 presents simulation studies comparing MRCE to competing methods, Section 5.4 presents an application of MRCE for predicting asset returns, and Section 5.5 concludes with a summary and discussion.

5.2 Joint estimation of B and Ω via penalized normal likelihood

5.2.1 The MRCE method

We propose a sparse estimator for B that accounts for correlated errors using penalized normal likelihood. We add two penalties to the negative log-likelihood function g to construct a sparse estimator of B depending on $\Omega = [\omega_{j'j}]$,

$$(\hat{B}, \hat{\Omega}) = \underset{B,\Omega}{\operatorname{argmin}} \left\{ g(B, \Omega) + \lambda_1 \sum_{j' \neq j} |\omega_{j'j}| + \lambda_2 \sum_{j=1}^{p} \sum_{k=1}^{q} |b_{jk}| \right\}, \tag{5.3}$$

where $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ are tuning parameters.

We selected the lasso penalty on the off-diagonal entries of the inverse error covariance Ω for two reasons. First, it ensures that an optimal solution for Ω has finite objective function value when there are more responses than samples ($q > n$); second, the penalty has the effect of reducing the number of parameters in the inverse error covariance, which is useful when q is large (Rothman et al., 2008). Other penalties such as the ridge penalty could be used when it is unreasonable to assume that the inverse error covariance matrix is sparse. If q is large, estimating a dense Ω means that the MRCE regression method has $O(q^2)$ additional parameters in Ω to estimate compared with doing separate lasso regressions for each response variable. Thus estimating a sparse Ω has considerably lower variability, and so we focus on the lasso penalty on Ω. We show in simulations that when the inverse error covariance matrix is not sparse, the lasso penalty on Ω still considerably outperforms ignoring covariance estimation altogether (i.e., doing a separate lasso regression for each response).

The lasso penalty on B introduces sparsity in $\hat{B}$, which reduces the number of parameters in the model and provides interpretation. In classical regression ($q = 1$), the lasso penalty can offer major improvement in prediction performance when there is a relatively small number of relevant predictors. This penalty also ensures that an optimal solution for B is a function of Ω. Without a penalty on B (i.e., $\lambda_2 = 0$), the optimal solution for B is always $\hat{B}^{\text{OLS}}$.

To see the effect of including the error covariance when estimating an L_1-penalized B, assume that we know Ω and also assume $p < n$. Solving (5.3) for B with Ω fixed is a convex problem (see Section 5.2.2) and thus there exists a global minimizer B^{opt}. This implies that there exists a zero subgradient of the objective function at B^{opt} (see Theorem 3.4.3 page 127 in Bazaraa et al. (2006)). We express this in matrix notation as,

$$0 = 2n^{-1}X^TXB^{\text{opt}}\Omega - 2n^{-1}X^TY\Omega + \lambda_2\Gamma,$$

which gives,

$$B^{\text{opt}} = \hat{B}^{\text{OLS}} - \lambda_2(2n^{-1}X^TX)^{-1}\Gamma\Omega^{-1}, \tag{5.4}$$

where $\Gamma \equiv \Gamma(B^{\text{opt}})$ is a $p \times q$ matrix with entries $\gamma_{ij} = \text{sign}(b_{ij}^{\text{opt}})$ if $b_{ij}^{\text{opt}} \neq 0$ and otherwise $\gamma_{ij} \in [-1, 1]$ with specific values chosen to solve (5.4). Ignoring the correlation in the error is equivalent to assuming that $\Omega^{-1} = I$. Thus having highly correlated errors will have greater influence on the amount of shrinkage of each entry of B^{opt} than having mildly correlated errors.

5.2.2 Computational algorithms

The optimization problem in (5.3) is not convex; however, solving for either B or Ω with the other fixed is convex. We present an algorithm for solving (5.3) and a fast approximation to it.

Solving (5.3) for Ω with B fixed at a chosen point B_0 yields the optimization problem,

$$\hat{\Omega}(B_0) = \underset{\Omega}{\text{argmin}} \left\{ \text{tr}\left(\hat{\Sigma}_R\Omega\right) - \log|\Omega| + \lambda_1 \sum_{j' \neq j} |\omega_{j'j}| \right\}, \tag{5.5}$$

where $\hat{\Sigma}_R = \frac{1}{n}(Y - XB_0)^T(Y - XB_0)$. This is exactly the L_1-penalized covariance estimation problem considered by d'Aspremont et al. (2008), Yuan and Lin (2007), Rothman et al. (2008), and Friedman et al. (2008). The fastest available algorithm for solving the covariance optimization problem in (5.5) is called the graphical lasso (glasso), proposed by Friedman et al. (2008).

Solving (5.3) for B with Ω fixed at a chosen point Ω_0 yields the optimization problem,

$$\hat{B}(\Omega_0) = \underset{B}{\text{argmin}} \left\{ \text{tr}\left[\frac{1}{n}(Y - XB)^T(Y - XB)\Omega_0\right] + \lambda_2 \sum_{j=1}^{p}\sum_{k=1}^{q} |b_{jk}| \right\}, \tag{5.6}$$

which is convex if Ω_0 is non-negative definite. This follows because the trace term in

the objective function has the Hessian $2n^{-1}\Omega_0 \otimes X^T X$, which is non-negative definite because the Kronecker product of two symmetric non-negative definite matrices is also non-negative definite. A solution can be efficiently computed using cyclical-coordinate descent analogous to that used for solving the single output lasso problem (Friedman et al., 2007). We summarize the optimization procedure in Algorithm 1. We use the ridge penalized least-squares estimate $\hat{B}^{\text{RIDGE}} = (X^T X + \lambda_2 I)^{-1} X^T Y$ to scale our test of parameter convergence since it is always well defined (including when $p > n$).

Algorithm 1. *Given Ω and an initial value $\hat{B}^{(0)}$, let $S = X^T X$ and $H = X^T Y \Omega$.*

Step 1: *Set $\hat{B}^{(m)} \leftarrow \hat{B}^{(m-1)}$. Visit all entries of $\hat{B}^{(m)}$ in some sequence and for entry (r, c) update $\hat{b}_{rc}^{(m)}$ with the minimizer of the objective function along its coordinate direction given by,*

$$\hat{b}_{rc}^{(m)} \leftarrow \text{sign}\left(\hat{b}_{rc}^{(m)} + \frac{h_{rc} - u_{rc}}{s_{rr}\omega_{cc}}\right)\left(\left|\hat{b}_{rc}^{(m)} + \frac{h_{rc} - u_{rc}}{s_{rr}\omega_{cc}}\right| - \frac{n\lambda_2}{s_{rr}\omega_{cc}}\right)_+,$$

where $u_{rc} = \sum_{j=1}^{p}\sum_{k=1}^{q} \hat{b}_{jk}^{(m)} s_{rj}\omega_{kc}$.

Step 2: *If $\sum_{j,k} |\hat{b}_{jk}^{(m)} - \hat{b}_{jk}^{(m-1)}| < \epsilon \sum_{j,k} |\hat{b}_{jk}^{\text{RIDGE}}|$ then stop, otherwise goto Step 1.*

A full derivation of Algorithm 1 is found in the Section 5.6. Algorithm 1 is guaranteed to converge to the global minimizer if the given Ω is non-negative definite. This follows from the fact that the trace term in the objective function is convex and differentiable and the penalty term decomposes into a sum of convex functions of individual parameters (Tseng, 1988; Friedman et al., 2007). We set the convergence tolerance parameter $\epsilon = 10^{-4}$.

In terms of computational cost, we need to cycle through pq parameters, and for each compute u_{rc}, which costs at most $O(pq)$ flops, and if the least sparse iterate has v non-zeros, then computing u_{rc} costs $O(v)$. The worst case cost for the entire algorithm is $O(p^2 q^2)$.

Using (5.5) and (5.6) we can solve (5.3) using block-wise coordinate descent, that is, we iterate minimizing with respect to B and minimizing with respect to Ω.

Algorithm 2 (MRCE). *For fixed values of λ_1 and λ_2, initialize $\hat{B}^{(0)} = 0$ and $\hat{\Omega}^{(0)} = \hat{\Omega}(\hat{B}^{(0)})$.*

Step 1: *Compute $\hat{B}^{(m+1)} = \hat{B}(\hat{\Omega}^{(m)})$ by solving* (5.6) *using Algorithm 1.*

Step 2: *Compute $\hat{\Omega}^{(m+1)} = \hat{\Omega}(\hat{B}^{(m+1)})$ by solving* (5.5) *using the glasso algorithm.*

Step 3: *If $\sum_{j,k} |\hat{b}_{jk}^{(m+1)} - \hat{b}_{jk}^{(m)}| < \epsilon \sum_{j,k} |\hat{b}_{jk}^{\text{RIDGE}}|$ then stop, otherwise goto Step 1.*

Algorithm 2 uses block-wise coordinate descent to compute a local solution for (5.3). Steps 1 and 2 both ensure a decrease in the objective function value. In practice we found that for certain values of the penalty tuning parameters (λ_1, λ_2), the algorithm may take many iterations to converge for high-dimensional data. For such cases, we propose a faster approximate solution to (5.3).

Algorithm 3 (Approximate MRCE). *For fixed values of λ_1 and λ_2,*

Step 1: *Perform q separate lasso regressions each with the same optimal tuning parameter $\hat{\lambda}_0$ selected with a cross validation procedure. Let $\hat{B}^{\text{lasso}}_{\hat{\lambda}_0}$ denote the solution.*

Step 2: *Compute $\hat{\Omega} = \hat{\Omega}(\hat{B}^{\text{lasso}}_{\hat{\lambda}_0})$ by solving* (5.5) *using the glasso algorithm.*

Step 3: *Compute $\hat{B} = \hat{B}(\hat{\Omega})$ by solving* (5.6) *using Algorithm 1.*

The approximation summarized in Algorithm 3 is only iterative inside its steps. The algorithm begins by finding the optimally tuned lasso solution $\hat{B}^{\text{lasso}}_{\hat{\lambda}_0}$ (using cross validation to select the tuning parameter $\hat{\lambda}_0$), then computes an estimate for Ω using the glasso algorithm with $\hat{B}^{\text{lasso}}_{\hat{\lambda}_0}$ plugged in, and then solves (5.6) using this inverse covariance estimate. Note that one still must select two tuning parameters (λ_1, λ_2). The performance of the approximation is studied in Section 5.3.

5.3 Simulation study

5.3.1 Estimators

We compare the performance of the MRCE method, computed with the exact and the approximate algorithms, to other multivariate regression estimators that produce sparse estimates of B. We report results for the following methods:

- *Lasso*: Performing q separate lasso regressions, each with the same tuning parameter λ.
- *Separate lasso*: Perform q separate lasso regressions, each with its own tuning parameter.
- *MRCE*: The solution to (5.3) (Algorithm 2).
- *Approx. MRCE*: An approximate solution to (5.3) (Algorithm 3).

The ordinary least squares estimator $\hat{B}^{\text{OLS}} = (X^T X)^{-1} X^T Y$ and the Curds and Whey method of Breiman and Friedman (1997) are computed as a benchmark for low-dimensional models (they are not directly applicable when $p > n$).

We select tuning parameters minimizing the squared prediction error, accumulated over all q responses, of independently generated validation data of the same sample size ($n = 50$). This is similar to performing cross-validation and is used to save computing time for the simulations. For the MRCE methods, the two tuning parameters are selected simultaneously.

5.3.2 Models

In each replication for each model, we generate an $n \times p$ predictor matrix X with rows drawn independently from $N_p(0, \Sigma_X)$ where $\Sigma_X = [\sigma_{Xij}]$ is given by $\sigma_{Xij} = 0.7^{|i-j|}$. This model for the predictors was also used by Yuan et al. (2007) and Peng et al. (2009). Note that all of the predictors are generated with the same unit

marginal variance. The error matrix E is generated independently with rows drawn independently from $N_q(0, \Sigma_E)$. We consider two models for the error covariance,

- AR(1) error covariance: $\sigma_{Eij} = \rho_E^{|i-j|}$, with values of ρ_E ranging from 0 to 0.9.
- Fractional Gaussian Noise (FGN) error covariance:

$$\sigma_{Eij} = 0.5\left((|i-j|+1)^{2H} - 2|i-j|^{2H} + (|i-j|-1)^{2H}\right),$$

  with values of the Hurst parameter $H = 0.9, 0.95$.

The inverse error covariance for the AR(1) model is a tri-diagonal sparse matrix while its covariance matrix is dense, and thus this error covariance model completely satisfies the regularizing assumptions for the MRCE method, which exploits the correlated error and the sparse inverse error covariance. The FGN model is a standard example of long-range dependence and both the error covariance and its inverse are dense matrices. Varying H gives different degree of dependence, from $H = 0.5$ corresponding to an i.i.d. sequence to $H = 1$ corresponding to a perfectly correlated one. Thus the introduction of sparsity in the inverse error covariance by the MRCE method should not help; however, since the errors are highly correlated the MRCE method may still perform better than the lasso penalized regressions for each response, which ignore correlation among the errors. The sample size is fixed at $n = 50$ for all models.

We generate sparse coefficient matrices B in each replication using the matrix element-wise product,

$$B = W * K * Q,$$

where W is generated with independent draws for each entry from $N(0, 1)$, K has entries with independent Bernoulli draws with success probability s_1, and Q has rows that are either all one or all zero, where p independent Bernoulli draws with success probability s_2 are made to determine whether each row is the ones vector or the zeros

vector. Generating B in this manner, we expect $(1 - s_2)p$ predictors to be irrelevant for all q responses, and we expect each relevant predictor to be relevant for $s_1 q$ of the response variables.

5.3.3 Performance evaluation

We measure performance using model error, following the approach in Yuan et al. (2007), which is defined as,

$$\mathrm{ME}(\hat{B}, B) = \mathrm{tr}\left[(\hat{B} - B)^T \Sigma_X (\hat{B} - B)\right].$$

We also measure the sparsity recognition performance using true positive rate (TPR) and true negative rate (TNR),

$$\mathrm{TPR}(\hat{B}, B) = \frac{\#\{(i,j) : \hat{b}_{ij} \neq 0 \quad \text{and} \quad b_{ij} \neq 0\}}{\#\{(i,j) : b_{ij} \neq 0\}}, \tag{5.7}$$

$$\mathrm{TNR}(\hat{B}, B) = \frac{\#\{(i,j) : \hat{b}_{ij} = 0 \quad \text{and} \quad b_{ij} = 0\}}{\#\{(i,j) : b_{ij} = 0\}}. \tag{5.8}$$

Both the true positive rate and true negative rates must be considered simultaneously since OLS always has perfect TPR and $\hat{B} = 0$ always has perfect TNR.

5.3.4 Results

The model error performance for AR(1) error covariance model is reported in Table 5.1 for low-dimensional models, and Table 5.2 for high-dimensional models. We see that the margin by which MRCE and its approximation outperform the lasso and separate lasso in terms of model error increases as the error correlation ρ_E increases. This trend is consistent with the analysis of the subgradient equation given in (5.4), since the manner by which MRCE performs lasso shrinkage exploits highly correlated errors. Additionally, the MRCE method and its approximation outperform the

lasso and separate lasso more for sparser coefficient matrices. We omitted the exact MRCE method for $p = 60, q = 20$ and $p = q = 100$ because these cases were computationally intractable. All of the sparse estimators outperform the ordinary least squares method by a considerable margin. The Curds and Whey method, although designed to exploit correlation in the responses, is outperformed here because it does not introduce sparsity in B.

Table 5.1: Model error for the AR(1) error covariance models of low dimension. Averages and standard errors in parenthesis are based on 50 replications with $n = 50$. Tuning parameters were selected using a 10^x resolution.

p	q	ρ_E	s_1, s_2	OLS	lasso	sep.lasso	MRCE	ap.MRCE	C&W
20	20	0.9	0.1, 1	14.46	2.78	2.81	0.89	0.96	9.89
				(0.54)	(0.09)	(0.09)	(0.02)	(0.03)	(0.38)
20	20	0.7	0.1, 1	14.48	2.86	2.83	2.00	1.99	10.72
				(0.32)	(0.07)	(0.08)	(0.05)	(0.06)	(0.23)
20	20	0.5	0.1, 1	14.49	2.85	2.88	2.78	2.64	11.08
				(0.27)	(0.08)	(0.08)	(0.07)	(0.07)	(0.19)
20	20	0	0.1, 1	14.39	2.93	3.00	3.32	3.18	11.57
				(0.25)	(0.07)	(0.10)	(0.07)	(0.07)	(0.19)
20	20	0.9	0.5, 1	14.46	10.10	9.12	3.98	4.95	11.95
				(0.54)	(0.23)	(0.20)	(0.11)	(0.15)	(0.38)
20	20	0.7	0.5, 1	14.48	10.59	9.29	7.70	7.87	13.00
				(0.32)	(0.23)	(0.18)	(0.21)	(0.18)	(0.27)
20	20	0.5	0.5, 1	14.49	10.45	9.26	10.08	9.20	13.32
				(0.27)	(0.23)	(0.18)	(0.26)	(0.19)	(0.24)
20	20	0	0.5, 1	14.39	10.40	9.08	10.33	9.71	13.58
				(0.25)	(0.17)	(0.15)	(0.22)	(0.18)	(0.22)

The model error performance for FGN error covariance model is reported in Table 5.3 for low-dimensional models and in Table 5.4 for high-dimensional models. Although there is no sparsity in the inverse error covariance for the MRCE method and its approximation to exploit, we see that both methods are still able to provide considerable improvement over the lasso and separate lasso methods by exploiting the highly correlated error. As seen with the AR(1) error covariance model, as the amount of correlation increases (i.e., larger values of H), the margin by which the

Table 5.2: Model error for the AR(1) error covariance models of high dimension. Averages and standard errors in parenthesis are based on 50 replications with $n = 50$. Tuning parameters were selected using a 10^x resolution.

p	q	ρ_E	s_1, s_2	OLS	lasso	sep.lasso	MRCE	ap.MRCE
20	60	0.9	0.1, 1	45.16	8.53	8.63	2.49	2.71
				(1.21)	(0.20)	(0.20)	(0.04)	(0.05)
20	60	0.7	0.1, 1	44.34	8.52	8.65	5.92	5.90
				(0.75)	(0.15)	(0.16)	(0.09)	(0.09)
20	60	0.5	0.1, 1	43.92	8.54	8.63	8.57	7.99
				(0.61)	(0.15)	(0.14)	(0.14)	(0.13)
20	60	0	0.1, 1	43.53	8.55	8.60	9.93	9.37
				(0.50)	(0.12)	(0.13)	(0.14)	(0.12)
60	20	0.9	0.1, 1	NA	11.05	11.07	-	5.00
					(0.32)	(0.32)		(0.13)
60	20	0.7	0.1, 1	NA	10.91	11.03	-	8.84
					(0.26)	(0.26)		(0.19)
60	20	0.5	0.1, 1	NA	10.76	10.88	-	10.83
					(0.24)	(0.24)		(0.20)
60	20	0	0.1, 1	NA	10.79	10.88	-	12.63
					(0.20)	(0.19)		(0.20)
100	100	0.9	0.5, 0.1	NA	58.79	59.32	-	34.87
					(2.29)	(2.35)		(1.54)
100	100	0.7	0.5, 0.1	NA	59.09	59.60	-	60.12
					(2.22)	(2.30)		(2.02)

MRCE method and its approximation outperform competitors increases.

Table 5.3: Model error for the FGN error covariance models of low dimension. Averages and standard errors in parenthesis are based on 50 replications with $n = 50$. Tuning parameters were selected using a 10^x resolution.

p	q	H	s_1, s_2	OLS	lasso	sep.lasso	MRCE	ap.MRCE	C&W
20	20	0.95	0.1, 1	14.51	2.72	2.71	1.03	1.01	9.86
				(0.69)	(0.10)	(0.11)	(0.02)	(0.03)	(0.46)
20	20	0.90	0.1, 1	14.49	2.76	2.77	1.78	1.71	10.29
				(0.53)	(0.09)	(0.09)	(0.05)	(0.05)	(0.36)
20	20	0.95	0.5, 1	14.51	9.89	8.94	3.63	4.42	11.72
				(0.69)	(0.26)	(0.21)	(0.09)	(0.16)	(0.45)
20	20	0.90	0.5, 1	14.49	10.01	9.03	6.11	6.34	12.29
				(0.53)	(0.21)	(0.18)	(0.14)	(0.13)	(0.34)

Table 5.4: Model error for the FGN error covariance models of high dimension. Averages and standard errors in parenthesis are based on 50 replications with $n = 50$. Tuning parameters were selected using a 10^x resolution.

p	q	H	s_1, s_2	OLS	lasso	sep.lasso	MRCE	ap.MRCE
20	60	0.95	0.1, 1	46.23	8.56	8.63	3.31	3.20
				(2.04)	(0.36)	(0.37)	(0.19)	(0.18)
20	60	0.90	0.1, 1	45.41	8.60	8.69	5.31	5.03
				(1.42)	(0.24)	(0.25)	(0.15)	(0.14)
60	20	0.95	0.1, 1	NA	11.15	11.23	-	4.84
					(0.35)	(0.36)		(0.12)
60	20	0.90	0.1, 1	NA	11.14	11.21	-	7.44
					(0.30)	(0.30)		(0.16)
100	100	0.95	0.5, 0.1	NA	58.28	58.86	-	31.85
					(2.36)	(2.44)		(1.26)
100	100	0.90	0.5, 0.1	NA	58.10	58.63	-	47.37
					(2.27)	(2.36)		(1.68)

We report the true positive rate and true negative rates in Table 5.5 for the AR(1) error covariance models and in Table 5.6 for the FGN error covariance models. We see that as the error correlation increases (larger values of ρ_E and H), the true positive rate for the MRCE method and its approximation increases, while the true negative rate tends to decrease. While all methods perform comparably on these sparsity measures, the substantially lower prediction errors obtained by the MRCE methods give them a clear advantage over other methods.

5.4 Example: Predicting Asset Returns

We consider a dataset of weekly log-returns of 9 stocks from 2004, analyzed in Yuan et al. (2007). We selected this dataset because it is the most recent dataset analyzed in the multivariate regression literature. The data are modeled with a first-order vector autoregressive model,

$$Y = \tilde{Y}B + E,$$

Table 5.5: True Positive Rate / True Negative Rate for the AR(1) error covariance models, averaged over 50 replications; $n = 50$. Standard errors are omitted, the largest standard error is 0.04 and most are less than 0.01. Tuning parameters were selected using a 10^x resolution.

p	q	ρ_E	s_1, s_2	lasso	sep.lasso	MRCE	ap.MRCE
20	20	0.9	0.1, 1	0.83/0.72	0.82/0.74	0.95/0.59	0.94/0.62
20	20	0.7	0.1, 1	0.83/0.71	0.82/0.73	0.89/0.60	0.89/0.63
20	20	0.5	0.1, 1	0.83/0.70	0.81/0.73	0.86/0.62	0.87/0.63
20	20	0	0.1, 1	0.84/0.70	0.82/0.72	0.85/0.63	0.85/0.64
20	20	0.9	0.5, 1	0.86/0.44	0.87/0.44	0.93/0.42	0.91/0.45
20	20	0.7	0.5, 1	0.85/0.47	0.87/0.42	0.86/0.51	0.86/0.52
20	20	0.5	0.5, 1	0.83/0.52	0.87/0.44	0.83/0.54	0.85/0.48
20	20	0	0.5, 1	0.84/0.50	0.87/0.43	0.84/0.51	0.82/0.56
20	60	0.9	0.1, 1	0.83/0.70	0.80/0.74	0.94/0.58	0.93/0.61
20	60	0.7	0.1, 1	0.84/0.71	0.81/0.73	0.89/0.61	0.89/0.62
20	60	0.5	0.1, 1	0.84/0.70	0.82/0.73	0.86/0.64	0.86/0.64
20	60	0	0.1, 1	0.83/0.71	0.81/0.74	0.85/0.63	0.85/0.65
60	20	0.9	0.1, 1	0.79/0.76	0.79/0.76	-	0.89/0.66
60	20	0.7	0.1, 1	0.79/0.76	0.78/0.76	-	0.85/0.65
60	20	0.5	0.1, 1	0.79/0.76	0.79/0.76	-	0.83/0.66
60	20	0	0.1, 1	0.79/0.76	0.79/0.76	-	0.81/0.66
100	100	0.9	0.5, 0.1	0.77/0.81	0.76/0.82	-	0.87/0.72
100	100	0.7	0.5, 0.1	0.78/0.81	0.76/0.82	-	0.82/0.72

Table 5.6: True Positive Rate / True Negative Rate for the FGN error covariance models averaged over 50 replications; $n = 50$. Standard errors are omitted, the largest standard error is 0.04 and most are less than 0.01. Tuning parameters were selected using a 10^x resolution.

p	q	H	s_1, s_2	lasso	sep.lasso	MRCE	ap.MRCE
20	20	0.95	0.1, 1	0.83/0.72	0.81/0.75	0.94/0.55	0.93/0.59
20	20	0.90	0.1, 1	0.84/0.71	0.83/0.73	0.90/0.59	0.89/0.61
20	20	0.95	0.5, 1	0.87/0.40	0.87/0.45	0.93/0.39	0.92/0.39
20	20	0.90	0.5, 1	0.86/0.43	0.87/0.45	0.88/0.51	0.90/0.43
20	60	0.95	0.1, 1	0.83/0.70	0.81/0.73	0.93/0.55	0.93/0.58
20	60	0.90	0.1, 1	0.83/0.70	0.81/0.73	0.90/0.58	0.90/0.60
60	20	0.95	0.1, 1	0.79/0.76	0.79/0.76	-	0.89/0.66
60	20	0.90	0.1, 1	0.79/0.76	0.78/0.76	-	0.87/0.65
100	100	0.95	0.5, 0.1	0.77/0.81	0.75/0.82	-	0.87/0.72
100	100	0.90	0.5, 0.1	0.77/0.81	0.75/0.82	-	0.83/0.71

where the response $Y \in \mathbb{R}^{T-1 \times q}$ has rows $\boldsymbol{y}_2, \ldots, \boldsymbol{y}_T$ and the predictor $\tilde{Y} \in \mathbb{R}^{T-1 \times q}$ has rows $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{T-1}$. Here $\boldsymbol{y}_t$ corresponds to the vector of log-returns for the 9 companies at week t. Let $B \in \mathbb{R}^{q \times q}$ denote the transition matrix. Following the approach of Yuan et al. (2007), we use log-returns from the first 26 weeks of the year ($T = 26$) as the training set, and the log-returns from the remaining 26 weeks of the year as the test set. Prediction performance is measured by the average mean-squared prediction error over the test set for each stock, with the model fitted using the training set. Tuning parameters were selected with 10-fold CV.

Average test squared error over the 26 test points is reported in Table 5.7, where we see that the MRCE method and its approximation have somewhat better performance than the lasso and separate lasso methods. The lasso estimate of the transition matrix B was all zeros, yielding the null model. Nonetheless, this results in prediction performance comparable, (i.e., within a standard error), to the FES method of Yuan et al. (2007) (copied directly from Table 3 on page 341), which was shown to be the best of several competitors for these data. This comparable performance of the null model suggests that the signal is very weak in this dataset. Separate lasso, MRCE, and its approximation estimated 3/81, 4/81, and 12/81 coefficients as non-zero, respectively.

We report the estimate of the unit lag coefficient matrix B for the approximate MRCE method in Table 5.8, which is the least sparse estimate, identifying 12 non-zero entries. The estimated unit lag coefficient matrix for separate lasso, MRCE, and approximate MRCE all identified the log-return for Walmart at week $t-1$ as a relevant predictor for the log-return of GE at week t, and the log-return for Ford at week $t-1$ as a relevant predictor for the log return of Walmart at week t. The FES does not provide any interpretation.

We also report the signs of the estimate for the inverse error covariance matrix for the MRCE method in Table 5.9. A non-zero entry (i, j) means that we estimate

Table 5.7: Average testing squared error for each output (company) $\times 1000$, based on 26 testing points. Standard errors are reported in parenthesis. The results for the FES method where copied from Table 3 in Yuan et al. (2007).

	OLS	sep.lasso	lasso	MRCE	ap.MRCE	FES
Walmart	0.98	0.44	0.42	0.41	0.41	0.40
	(0.27)	(0.10)	(0.12)	(0.11)	(0.11)	
Exxon	0.39	0.31	0.31	0.31	0.31	0.29
	(0.08)	(0.07)	(0.07)	(0.07)	(0.07)	
GM	1.68	0.71	0.71	0.71	0.69	0.62
	(0.42)	(0.17)	(0.17)	(0.17)	(0.17)	
Ford	2.15	0.77	0.77	0.77	0.77	0.69
	(0.61)	(0.25)	(0.25)	(0.25)	(0.25)	
GE	0.58	0.45	0.45	0.45	0.45	0.41
	(0.15)	(0.09)	(0.09)	(0.09)	(0.09)	
ConocoPhillips	0.98	0.79	0.79	0.79	0.78	0.79
	(0.24)	(0.22)	(0.22)	(0.22)	(0.22)	
Citigroup	0.65	0.61	0.66	0.62	0.62	0.59
	(0.17)	(0.13)	(0.14)	(0.13)	(0.13)	
IBM	0.62	0.49	0.49	0.49	0.47	0.51
	(0.14)	(0.10)	(0.10)	(0.10)	(0.09)	
AIG	1.93	1.88	1.88	1.88	1.88	1.74
	(0.93)	(1.02)	(1.02)	(1.02)	(1.02)	
AVE	1.11	0.72	0.72	0.71	0.71	0.67
	(0.14)	(0.12)	(0.12)	(0.12)	(0.12)	

that ϵ_i is correlated with ϵ_j given the other errors (or ϵ_i is partially correlated with ϵ_j). We see that AIG (an insurance company) is estimated to be partially correlated with most of the other companies, and companies with similar products are partially correlated, such as Ford and GM (automotive), GE and IBM (technology), as well as Conoco Phillips and Exxon (oil). These results make sense in the context of financial data.

5.5 Summary and discussion

We proposed the MRCE method to produce a sparse estimate of the multivariate regression coefficient matrix B. Our method explicitly accounts for the correlation of

Table 5.8: Estimated coefficient matrix B for approximate MRCE. Results are rounded to the nearest tenth, and coefficients that are exactly zero are denoted by "0".

	Wal	Exx	GM	Ford	GE	CPhil	Citi	IBM	AIG
Walmart	0	0	0	0	0	0	0.1	0.1	0
Exxon	0	0	0	0	0	0	0	0	0
GM	0	0	0	0	0	0	0	0	0
Ford	-0.1	0.0	0.0	0	0	0	0	-0.0	-0.0
GE	0	0	0	0	0	0.0	0	0	0
ConocoPhillips	0	0.0	0	0	0	0	0	-0.0	0
Citigroup	0	0	0.0	0	0	0	0	0	0
IBM	0	0	0	0	0	0	0	0	0
AIG	0	0	0.0	0	0	0	0	0	0

Table 5.9: Signs of the inverse error covariance estimate for MRCE

	Wal	Exx	GM	Ford	GE	CPhil	Citi	IBM	AIG
Walmart	+	0	-	0	0	0	0	0	-
Exxon	0	+	0	0	0	-	0	0	-
GM	-	0	+	-	-	0	-	-	-
Ford	0	0	-	+	0	0	0	0	0
GE	0	0	-	0	+	0	-	-	-
CPhillips	0	-	0	0	0	+	0	0	-
Citigroup	0	0	-	0	-	0	+	0	-
IBM	0	0	-	0	-	0	0	+	-
AIG	-	-	-	0	-	-	-	-	+

the response variables. We also developed a fast approximate algorithm for computing MRCE which has roughly the same performance in terms of model error. These methods were shown to outperform q separate lasso penalized regressions (which ignore the correlation in the responses) in simulations when the responses are highly correlated, even when the inverse error covariance is dense.

Although we considered simultaneous L_1-penalization of B and Ω, one could use other penalties that introduce less bias instead, such as SCAD (Fan and Li, 2001; Lam and Fan, 2009). In addition, this work could be extended to the situation when the response vector samples have serial correlation, in which case the model would

involve both the error covariance and the correlation among the samples.

5.6 Derivation of Algorithm 1

The objective function for Ω fixed at Ω_0 is now,

$$f(B) = g(B, \Omega_0) + \lambda_2 \sum_{j=1}^{p} \sum_{k=1}^{q} |b_{jk}|.$$

We can solve for B with cyclical coordinate descent. Express the directional derivatives as,

$$\begin{aligned}
\frac{\partial f^+}{\partial B} &= \frac{2}{n} X^T X B \Omega - \frac{2}{n} X^T Y \Omega + \lambda_2 1(b_{ij} \geq 0) - \lambda_2 1(b_{ij} < 0) \\
\frac{\partial f^-}{\partial B} &= -\frac{2}{n} X^T X B \Omega + \frac{2}{n} X^T Y \Omega - \lambda_2 1(b_{ij} > 0) + \lambda_2 1(b_{ij} \leq 0),
\end{aligned}$$

where the indicator $1()$ is understood to be a matrix. Let $S = X^T X$ and $H = X^T Y \Omega$ and $u_{rc} = \sum_{j=1}^{p} \sum_{k=1}^{q} b_{jk} s_{rj} \omega_{kc}$. To update a single parameter b_{rc} we have the directional derivatives,

$$\begin{aligned}
\frac{\partial f^+}{\partial b_{rc}} &= u_{rc} - h_{rc} + n\lambda_2 1(b_{ij} \geq 0) - n\lambda_2 1(b_{ij} < 0) \\
\frac{\partial f^-}{\partial b_{rc}} &= -u_{rc} + h_{rc} - n\lambda_2 1(b_{ij} > 0) + n\lambda_2 1(b_{ij} \leq 0).
\end{aligned}$$

Let b_{rc}^0 be our current iterate. The unpenalized univariate minimizer $\hat{b}_{rc}^*$ solves,

$$\hat{b}_{rc}^* s_{rr} \omega_{cc} - b_{rc}^0 s_{rr} \omega_{cc} + u_{rc} - h_{rc} = 0,$$

implying $\hat{b}_{rc}^* = b_{rc}^0 + \frac{h_{rc} - u_{rc}}{s_{rr} \omega_{cc}}$. If $\hat{b}_{rc}^* > 0$, then we look leftward and by convexity the penalized minimizer is $\hat{b}_{rc} = \max(0, \hat{b}_{rc}^* - \frac{n\lambda_2}{s_{rr} \omega_{cc}})$. Similarly if $\hat{b}_{rc}^* < 0$ then we look to the right and by convexity the penalized univariate minimizer is $\hat{b}_{rc} = \min(0, \hat{b}_{rc}^* +$

$\frac{n\lambda_2}{s_{rr}\omega_{cc}}$), thus $\hat{b}_{rc} = \text{sign}(\hat{b}^*_{rc})(|\hat{b}^*_{rc}| - \frac{n\lambda_2}{s_{rr}\omega_{cc}})_+$. Also if $\hat{b}^*_{rc} = 0$, which has probability zero, then both the loss and penalty part of the objective function are minimized and the parameter stays at 0. We can write this solution as,

$$\hat{b}_{rc} = \text{sign}\left(b^0_{rc} + \frac{h_{rc} - u_{rc}}{s_{rr}\omega_{cc}}\right)\left(\left|b^0_{rc} + \frac{h_{rc} - u_{rc}}{s_{rr}\omega_{cc}}\right| - \frac{n\lambda_2}{s_{rr}\omega_{cc}}\right)_+ .$$

BIBLIOGRAPHY

BIBLIOGRAPHY

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., and Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc Natl Acad Sci USA*, 96(12):6745–6750.

Anderson, T. (1951). Estimating linear restrictions on regression coefficients for multivariate normal distributions. *Ann. Math. Statist.*, 22:327–351.

Antoniadis, A. and Fan, J. (2001). Regularization of wavelet approximations. *J. Amer. Statist. Assoc.*, 96(455):939–955.

Bair, E., Hastie, T., Paul, D., and Tibshirani, R. (2006). Prediction by supervised principal components. *J. Amer. Statist. Assoc.*, 101(473):119–137.

Bazaraa, M. S., Sherali, H. D., and Shetty, C. M. (2006). *Nonlinear Programming: Theory and Algorithms*. Wiley, New Jersey, 3rd edition.

Bedrick, E. and Tsai, C. (1994). Model selection for multivariate regression in small samples. *Biometrics*, 50:226–231.

Bickel, P. J. and Levina, E. (2004). Some theory for Fisher's linear discriminant function, "naive Bayes", and some alternatives when there are many more variables than observations. *Bernoulli*, 10(6):989–1010.

Bickel, P. J. and Levina, E. (2008a). Covariance regularization by thresholding. *Ann. Statist.*, 36(6):2577–2604.

Bickel, P. J. and Levina, E. (2008b). Regularized estimation of large covariance matrices. *Ann. Statist.*, 36(1):199–227.

Breiman, L. (1995). Better subset regression using the nonnegative garrote. *Technometrics*, 37(4):373–384.

Breiman, L. and Friedman, J. H. (1997). Predicting multivariate responses in multiple linear regression (Disc: p37-54). *J. Roy. Statist. Soc., Ser. B*, 59:3–37.

Brown, P., Vannucci, M., and Fearn, T. (2002). Bayes model averaging with selection of regressors. *J. R. Statist. Soc. B*, 64:519–536.

Chaudhuri, S., Drton, M., and Richardson, T. S. (2007). Estimation of a covariance matrix with zeros. *Biometrika*, 94(1):199–216.

d'Aspremont, A., Banerjee, O., and El Ghaoui, L. (2008). First-order methods for sparse covariance selection. *SIAM Journal on Matrix Analysis and its Applications*, 30(1):56–66.

Dempster, A. (1972). Covariance selection. *Biometrics*, 28:157–175.

Dey, D. K. and Srinivasan, C. (1985). Estimation of a covariance matrix under Stein's loss. *Ann. Statist.*, 13(4):1581–1591.

Donoho, D. L. and Johnstone, I. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81:425–455.

Donoho, D. L., Johnstone, I. M., Kerkyacharian, G., and Pickard, D. (1995). Wavelet shrinkage: asymptopia? (with discussion). *J. Roy. Statist. Soc., Ser. B*, 57:301–369.

Drton, M. and Perlman, M. D. (2008). A SINful approach to Gaussian graphical model selection. *J. Statist. Plann. Inference*, 138(4):1179–1200.

Eisen, M., Spellman, P., Brown, P., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci. USA*, 95(25):14863–14868.

El Karoui, N. (2008). Operator norm consistent estimation of large dimensional sparse covariance matrices. *Ann. Statist.*, 36(6):2717–2756.

Fan, J. (1997). Comments on "Wavelets in statistics: A review" by A. Antoniadis. *J. Italian Statist. Assoc.*, 6:131–139.

Fan, J., Fan, Y., and Lv, J. (2008a). High dimensional covariance matrix estimation using a factor model. *Journal of Econometrics*, 147:186–197.

Fan, J., Feng, Y., and Wu, Y. (2009). Network exploration via the adaptive LASSO and SCAD penalties. *Annals of Applied Statistics*, 3:521–541.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Statist. Assoc.*, 96(456):1348–1360.

Fan, J. and Peng, H. (2004). Nonconcave penalized likelihood with a diverging number of parameters. *Ann. Statist.*, 32(3):928–961.

Fan, J., Wang, M., and Yao, Q. (2008b). Modelling multivariate volatilities via conditionally uncorrelated components. *J. Royal Statist. Soc. Ser. B.* To appear.

Friedman, J., Hastie, T., and Tibshirani, R. (2007). Pathwise coordinate optimization. *Annals of Applied Statistics*, 1(2):302–332.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

Fu, W. (1998). Penalized regressions: the bridge versus the lasso. *Journal of Computational and Graphical Statistics*, 7(3):397–416.

Fujikoshi, Y. and Satoh, K. (1997). Modified AIC and C_p in multivariate linear regression. *Biometrika*, 84:707–716.

Furrer, R. and Bengtsson, T. (2007). Estimation of high-dimensional prior and posterior covariance matrices in Kalman filter variants. *Journal of Multivariate Analysis*, 98(2):227–255.

Golub, G. H. and Van Loan, C. F. (1989). *Matrix Computations*. The John Hopkins University Press, Baltimore, Maryland, 2nd edition.

Haff, L. R. (1980). Empirical Bayes estimation of the multivariate normal covariance matrix. *Ann. Statist.*, 8(3):586–597.

Hastie, T., Tibshirani, R., Eisen, M., Alizadeh, A., Levy, R., Staudt, L., Botstein, D., and Brown, P. (2000). Identifying distinct sets of genes with similar expression patterns via gene shaving. *Genome Biology*, 1(2):1–21.

Huang, J., Liu, N., Pourahmadi, M., and Liu, L. (2006). Covariance matrix selection and estimation via penalised normal likelihood. *Biometrika*, 93(1):85–98.

Hunter, D. R. and Li, R. (2005). Variable selection using mm algorithms. *Ann. Statist.*, 33(4):1617–1642.

Izenman, A. J. (1975). Reduced-rank regression for the multivariate linear model. *Journal of Multivariate Analysis*, 5(2):248–264.

Johnstone, I. M. (2001). On the distribution of the largest eigenvalue in principal components analysis. *Ann. Statist.*, 29(2):295–327.

Johnstone, I. M. and Lu, A. Y. (2004). Sparse principal components analysis. Unpublished manuscript.

Kalisch, M. and Bühlmann, P. (2007). Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *J. Mach. Learn. Res.*, 8:613–636.

Khan, J., Wei, J., Ringner, M., Saal, L., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C. R., Peterson, C., and Meltzer, P. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, 7(6):673–679.

Krzanowski, W. (1979). Between-groups comparison of principal components. *J. Amer. Statist. Assoc.*, 74(367):703–707.

Lam, C. and Fan, J. (2009). Sparsistency and rates of convergence in large covariance matrices estimation. *Annals of Statistics*. To appear.

Ledoit, O. and Wolf, M. (2003). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88:365–411.

Levina, E., Rothman, A. J., and Zhu, J. (2008). Sparse estimation of large covariance matrices via a nested Lasso penalty. *Annals of Applied Statistics*, 2(1):245–263.

Lin, S. P. and Perlman, M. D. (1985). A Monte Carlo comparison of four estimators for a covariance matrix. In Krishnaiah, P. R., editor, *Multivariate Analysis*, volume 6, pages 411–429. Elsevier Science Publishers.

Mardia, K. V., Kent, J. T., and Bibby, J. M. (1979). *Multivariate Analysis*. Academic Press, New York.

Meinshausen, N. and Bühlmann, P. (2006). High dimensional graphs and variable selection with the Lasso. *Ann. Statist.*, 34(3):1436–1462.

Obozinski, G., Wainwright, M. J., and Jordan, M. I. (2008). Union support recovery in high-dimensional multivariate regression. Technical Report 761, UC Berkeley, Department of Statistics.

Paul, D. (2007). Asymptotics of sample eigenstructure for a large dimensional spiked covariance model. *Stat. Sinica*, 17(4):1617–1642.

Peng, J., Zhu, J., Bergamaschi, A., Han, W., Noh, D.-Y., Pollack, J. R., and Wang, P. (2009). Regularized multivariate regression for identifying master predictors with application to integrative genomics study of breast cancer. *Annals of Applied Statistics*. To appear.

Pourahmadi, M. (1999). Joint mean-covariance models with applications to longitudinal data: unconstrained parameterisation. *Biometrika*, 86:677–690.

Pourahmadi, M. (2007). Cholesky decompositions and estimation of a covariance matrix: orthogonality of variance-correlation parameters. *Biometrika*, 94(4):1006–1013.

Reinsel, G. (1997). *Elements of Multivariate Time Series Analysis*. Springer, New York, 2nd edition.

Reinsel, G. and Velu, R. (1998). *Multivariate Reduced-rank Regression: Theory and Applications*. Springer, New York.

Rothman, A. J., Bickel, P. J., Levina, E., and Zhu, J. (2008). Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515.

Rothman, A. J., Levina, E., and Zhu, J. (2009). Generalized thresholding of large covariance matrices. *J. Amer. Statist. Assoc. (Theory and Methods)*, 104(485):177–186.

Rothman, A. J., Levina, E., and Zhu, J. (2010). A new approach to Cholesky-based covariance regularization in high dimensions. *Biometrika*. To appear.

Saulis, L. and Statulevičius, V. A. (1991). *Limit Theorems for Large Deviations*. Kluwer Academic Publishers, Dordrecht.

Smith, M. and Kohn, R. (2002). Parsimonious covariance matrix estimation for longitudinal data. *J. Amer. Statist. Assoc.*, 97(460):1141–1153.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc., Ser. B*, 58:267–288.

Tseng, P. (1988). Coordinate ascent for maximizing nondifferentiable concave functions. Technical Report LIDS-P, 1840, Massachusetts Institute of Technology, Laboratory for Information and Decision Systems.

Turlach, B. A., Venables, W. N., and Wright, S. J. (2005). Simultaneous variable selection. *Technometrics*, 47(3):349–363.

Wagaman, A. S. and Levina, E. (2009). Discovering sparse covariance structures with the Isomap. *J. Comp. Graph. Statist.*, 18. To appear.

Wang, L., Zhu, J., and Zou, H. (2007). Hybrid huberized support vector machines for microarray classification. In *ICML '07: Proceedings of the 24th International Conference on Machine Learning*, pages 983–990, New York, NY, USA. ACM Press.

Watkins, D. S. (1991). *Fundamentals of matrix computations*. John Wiley & Sons, Inc., New York, NY, USA.

Witten, D. M. and Tibshirani, R. (2009). Covariance-regularized regression and classification for high-dimensional problems. *Journal of the Royal Statistical Society, Series B*, 71(3):615–636.

Wong, F., Carter, C., and Kohn, R. (2003). Efficient estimation of covariance selection models. *Biometrika*, 90:809–830.

Wu, W. B. and Pourahmadi, M. (2003). Nonparametric estimation of large covariance matrices of longitudinal data. *Biometrika*, 90:831–844.

Yuan, M., Ekici, A., Lu, Z., and Monteiro, R. (2007). Dimension reduction and coefficient estimation in multivariate linear regression. *Journal of The Royal Statistical Society Series B*, 69(3):329–346.

Yuan, M. and Lin, Y. (2007). Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35.

Zou, H. (2006). The adaptive lasso and its oracle properties. *J. Amer. Statist. Assoc.*, 101(476):1418–1429.