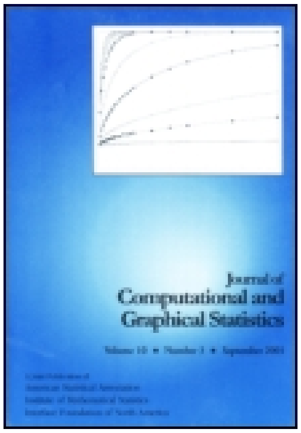

Journal of Computational and Graphical Statistics
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/ucgs20

Penalized Covariance Matrix Estimation using a Matrix-Logarithm Transformation

Xinwei Deng [a] & Kam-Wah Tsui [b]
[a] Department of Statistics, Virginia Tech
[b] Department of Statistics, University of Wisconsin-Madison

Accepted author version posted online: 03 Aug 2012

Disclaimer: This is a version of an unedited manuscript that has been accepted for publication. As a service to authors and researchers we are providing this version of the accepted manuscript (AM). Copyediting, typesetting, and review of the resulting proof will be undertaken on this manuscript before final publication of the Version of Record (VoR). During production and pre-press, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal relate to this version also.

Penalized Covariance Matrix Estimation using a Matrix-Logarithm Transformation

Xinwei Deng[1]* and Kam-Wah Tsui[2]

[1] *Department of Statistics, Virginia Tech*

[2] *Department of Statistics, University of Wisconsin-Madison*

(July 31, 2012)

Abstract

For statistical inferences that involve covariance matrices, it is desirable to obtain an accurate covariance matrix estimate with a well-structured eigen-system. We propose to estimate the covariance matrix through its matrix logarithm based on an approximate log-likelihood function. We develop a generalization of the Leonard and Hsu (1992) log-likelihood approximation that no longer requires a nonsingular sample covariance matrix. The matrix log-transformation provides the ability to impose a convex penalty on the transformed likelihood such that the largest and smallest eigenvalues of the covariance matrix estimate can be regularized *simultaneously*. The proposed method transforms the problem of estimating the covariance matrix into the problem of estimating a symmetric matrix, which can be solved efficiently by an iterative quadratic programming algorithm. The merits of the proposed method are illustrated by a simulation study and two real applications in classification and portfolio optimization.

1 Introduction

Covariance matrix estimation is of fundamental importance in multivariate analysis and many statistical applications. For example, principal component analysis (PCA) applies the eigen-decomposition

*Address for correspondence: Xinwei Deng, Department of Statistics, Virginia Tech, 211 Hutcheson Hall, Blacksburg, VA 24061 (E-mail: xdeng@vt.edu).

of the covariance matrix for dimension reduction. For the classification problem, linear discriminant analysis (LDA) and other procedures need the inverse of a covariance matrix to compute the classification rule. In finance, portfolio optimization often utilizes the inverse of the covariance matrix for minimizing the portfolio risk. Although one can estimate the covariance matrix and then obtain its inverse, the inversion can be computationally intensive and unstable as the dimension increases. It is desirable to obtain an accurate covariance matrix estimate with a well-structured eigen-system.

Suppose $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are independently and identically distributed p-dimensional random vectors which follow a multivariate normal distribution $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Without loss of generality, we assume that $\boldsymbol{\mu} = \boldsymbol{0}$. Let $\boldsymbol{S} = \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i'/n$ be the sample covariance matrix. The negative log-likelihood function of $\boldsymbol{\Sigma}$ given the sample, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, is proportional to

$$L_n(\boldsymbol{\Sigma}) = -\log|\boldsymbol{\Sigma}^{-1}| + \mathrm{tr}[\boldsymbol{\Sigma}^{-1}\boldsymbol{S}], \tag{1.1}$$

up to some constant. When $p < n$, $\boldsymbol{S}$ is the maximum likelihood estimate of $\boldsymbol{\Sigma}$. It is well known that $\boldsymbol{S}$ is not a stable estimate of $\boldsymbol{\Sigma}$ when p is large or p is close to the sample size n. As the dimension p increases, the largest eigenvalues of $\boldsymbol{S}$ tend to be systematically distorted, which can result in an ill-conditioned estimate of $\boldsymbol{\Sigma}$ (Stein, 1975; Johnstone, 2001). When $p > n$, $\boldsymbol{S}$ is singular and the smallest eigenvalue is zero. It is not appropriate to use $\boldsymbol{S}$ to obtain the estimate of $\boldsymbol{\Sigma}^{-1}$.

Many alternative estimates of $\boldsymbol{\Sigma}$ or $\boldsymbol{\Sigma}^{-1}$ have been proposed from different directions. One direction is to seek the sparsity in $\boldsymbol{\Sigma}^{-1}$ to improve the estimation accuracy and to explore the structure of the Gaussian graphical model. Research in this direction includes Meinshausen and Buhlmann (2006), Banerjee et al. (2006), Huang et al. (2006), Yuan and Lin (2007), Friedman et al. (2008), Rajaratnam et al. (2008), Levina et al. (2008), Peng et al. (2009), Cai et al. (2010) , Yuan (2010), and Zhou et al. (2010) among many others. Bickel and Levina (2008a) and Rothman et al. (2009) proposed thresholding methods to estimate $\boldsymbol{\Sigma}$ with attractive theoretical properties. Bickel and Levina (2008b) developed a banding estimate for $\boldsymbol{\Sigma}$, and Rothman et al. (2010) proposed to use Cholesky decomposition to construct a banding estimate of $\mathbf{\Sigma}$, which guarantees to be positive definite. Fan et al. (2008) developed a factor model method in estimating both $\mathbf{\Sigma}$ and its inverse.

Another direction is to construct shrinkage estimates of the covariance matrix. Early work was developed to shrink the eigenvalues of the sample covariance matrix (Dey and Srinivasan, 1985; Haff, 1991). Ledoit and Wolf (2006) considered an estimate of $\mathbf{\Sigma}$ to be a linear combination of the sample covariance matrix and a pre-chosen matrix. Recently, Won et al. (2009) proposed what they called a well-conditioned estimate of $\mathbf{\Sigma}$ by encouraging the condition number to be bounded. Recall that the condition number of a covariance matrix is its largest eigenvalue divided by its smallest eigenvalue. Note that these methods have retained the use of the eigenvectors of $\boldsymbol{S}$ in estimating $\mathbf{\Sigma}$ or $\mathbf{\Sigma}^{-1}$. It is known that the eigenvectors of $\boldsymbol{S}$ are not consistent as p increases (Johnstone and Lu, 2009). Thus, only shrinking the eigenvalues of $\boldsymbol{S}$ may not be sufficient to obtain a reasonable and accurate estimate of $\mathbf{\Sigma}$.

Since a covariance matrix is positive definite, it is natural to require a covariance matrix estimate to be positive definite. This mathematical restriction makes the covariance matrix estimation problem challenging. Note that the covariance matrix $\mathbf{\Sigma}$ can be expressed as a matrix exponential of a real symmetric matrix $\boldsymbol{A}$. That is, $\boldsymbol{A}$ is the matrix logarithm of the covariance matrix. More properties of the matrix logarithm can be found in Chiu et al. (1996). Expressing the likelihood function in terms of $\boldsymbol{A}$ releases the mathematical restriction. An estimate of the matrix logarithm thus automatically guarantees the resulting covariance matrix to be positive definite. Leonard and Hsu (1992) used this transformation method to provide a class of flexible covariance matrices in Bayesian inferences. Their method, however, relies on the availability of a nonsingular sample covariance matrix. Hence, their method is not applicable when the dimension of the sample vectors is larger than the sample size, where the sample covariance matrix becomes singular. We propose to significantly generalize the Leonard and Hsu result such that the sample covariance matrix no longer needs be non-singular. Using an appropriate penalty function together with the generalization of the Leonard and Hsu result, we show that one can *simultaneously* regularize the largest and smallest eigenvalues of the covariance matrix estimate, achieving a well-conditioned covariance matrix estimate. After transforming the problem of estimating a covariance matrix into the problem of estimating a real symmetric matrix $\boldsymbol{A}$, we develop an efficient iterative quadratic programming algorithm to solve the transformed problem. We call our proposed estimation method, Log-ME, to stand for “Logarithm-transformed Matrix Estimate”.

The remaining of this article is organized as follows. We describe how the matrix logarithm transformation is used for the log-likelihood function in Section 2. Section 3 details the development of our Log-ME method. We conduct a simulation study to bring out the merits of the Log-ME method in Section 4. In Section 5, we present two real applications of the Log-ME covariance matrix estimate: one in classification and another one in portfolio optimization. We also examine the performance of other covariance matrix estimates in Sections 4 and 5. We conclude this work with some discussion in Section 6.

2 Log-likelihood using Matrix Log-Transformation

Consider the spectral decomposition of the covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{TDT}'$, where $\boldsymbol{D} = diag(d_1, \ldots, d_p)$ is a diagonal matrix of the eigenvalues of $\boldsymbol{\Sigma}$, and $\boldsymbol{T}$ is an orthonormal matrix consisting of eigenvectors of $\boldsymbol{\Sigma}$. Assume that $d_1 \geq d_2 \geq \cdots \geq d_p > 0$. Let $\boldsymbol{A} = (a_{ij})_{p\times p} = \log(\boldsymbol{\Sigma})$ be the matrix logarithm of $\boldsymbol{\Sigma}$. That is, $\boldsymbol{\Sigma} = \sum_{k=0}^{\infty} \boldsymbol{A}^k/k! \equiv \exp(\boldsymbol{A})$, where $\exp(\boldsymbol{A})$ is called the matrix exponential of $\boldsymbol{A}$. Then

$$\boldsymbol{A} = \boldsymbol{TMT}', \tag{2.1}$$

where $\boldsymbol{M}$ is a diagonal matrix, $diag(\log(d_1), \ldots, \log(d_p))$. In terms of $\boldsymbol{A}$, the negative log-likelihood function in (1.1) becomes

$$L_n(\boldsymbol{A}) = \mathrm{tr}(\boldsymbol{A}) + \mathrm{tr}[\exp(-\boldsymbol{A})\boldsymbol{S}]. \tag{2.2}$$

A major advantage of using the matrix logarithm transformation is that it converts the problem of estimating a positive definite matrix $\boldsymbol{\Sigma}$ into a problem of estimating a real symmetric matrix $\boldsymbol{A}$. The estimates of $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}^{-1}$ can then be obtained through the matrix exponential of $\boldsymbol{A}$. Because of the matrix exponential term in (2.2), estimating $\boldsymbol{A}$ by directly minimizing $L_n(\boldsymbol{A})$ in (2.2) is nontrivial. To circumvent this challenge, we consider an approximation to the second term in (2.2) by using the Volterra integral equation (Bellman, 1970, page 175),

$$\exp(\boldsymbol{A}t) = \exp(\boldsymbol{A}_0 t) + \int_0^t \exp(\boldsymbol{A}_0(t-s))(\boldsymbol{A}-\boldsymbol{A}_0)\exp(\boldsymbol{A}s)ds, \quad 0 < t < \infty. \tag{2.3}$$

As in Leonard and Hsu (1992), we repeatedly apply (2.3) to obtain

$$\begin{aligned}\exp(\boldsymbol{A}t) = {} & \exp(\boldsymbol{A}_0 t) + \int_0^t \exp(\boldsymbol{A}_0(t-s))(\boldsymbol{A}-\boldsymbol{A}_0)\exp(\boldsymbol{A}_0 s)ds \\ & + \int_0^t \int_0^s \exp(\boldsymbol{A}_0(t-s))(\boldsymbol{A}-\boldsymbol{A}_0)\exp(\boldsymbol{A}_0(s-u))(\boldsymbol{A}-\boldsymbol{A}_0)\exp(\boldsymbol{A}_0 u)duds \\ & + \text{cubic and higher order terms,}\end{aligned} \tag{2.4}$$

where $\boldsymbol{A}_0 = \log(\boldsymbol{\Sigma}_0)$ and $\boldsymbol{\Sigma}_0$ is an initial estimate of $\boldsymbol{\Sigma}$. In the situation when p is close to or larger than n, the sample covariance matrix $\boldsymbol{S}$ is ill-conditioned and its matrix logarithm is not well-defined. It cannot be used as a proper initial estimate $\boldsymbol{\Sigma}_0$. Our equation (2.4) thus differs from that in Leonard and Hsu (1992) who used $\boldsymbol{\Sigma}_0 = \boldsymbol{S}$ as the initial estimate of $\boldsymbol{\Sigma}$. Taking $t = 1$ in (2.4) and substituting $\boldsymbol{A}, \boldsymbol{A}_0$ in (2.4) with $-\boldsymbol{A}, -\boldsymbol{A}_0$, we have

$$\begin{aligned}\exp(-\boldsymbol{A}) = {} & \boldsymbol{\Sigma}_0^{-1} - \int_0^1 \boldsymbol{\Sigma}_0^{s-1}(\boldsymbol{A}-\boldsymbol{A}_0)\boldsymbol{\Sigma}_0^{-s}ds \\ & + \int_0^1 \int_0^s \boldsymbol{\Sigma}_0^{s-1}(\boldsymbol{A}-\boldsymbol{A}_0)\boldsymbol{\Sigma}_0^{u-s}(\boldsymbol{A}-\boldsymbol{A}_0)\boldsymbol{\Sigma}_0^{-u}duds \\ & + \text{cubic and higher order terms.}\end{aligned} \tag{2.5}$$

Therefore,

$$\begin{aligned}\mathrm{tr}[\exp(-\boldsymbol{A})\boldsymbol{S}] =& \mathrm{tr}(\boldsymbol{S}\boldsymbol{\Sigma}_0^{-1}) - \int_0^1 \mathrm{tr}[(\boldsymbol{A}-\boldsymbol{A}_0)\boldsymbol{\Sigma}_0^{-s}\boldsymbol{S}\boldsymbol{\Sigma}_0^{s-1}]ds \\ &+ \int_0^1\int_0^s \mathrm{tr}[(\boldsymbol{A}-\boldsymbol{A}_0)\boldsymbol{\Sigma}_0^{u-s}(\boldsymbol{A}-\boldsymbol{A}_0)\boldsymbol{\Sigma}_0^{-u}\boldsymbol{S}\boldsymbol{\Sigma}_0^{s-1}]duds \\ &+ \text{cubic and higher order terms.} \qquad (2.6)\end{aligned}$$

By leaving out the cubic and higher order terms in (2.6), we approximate $L_n(\boldsymbol{A})$ in (2.2) by using $l_n(\boldsymbol{A})$ given below:

$$\begin{aligned}l_n(\boldsymbol{A}) =& \mathrm{tr}(\boldsymbol{S}\boldsymbol{\Sigma}_0^{-1}) - \left[\int_0^1 \mathrm{tr}[(\boldsymbol{A}-\boldsymbol{A}_0)\boldsymbol{\Sigma}_0^{-s}\boldsymbol{S}\boldsymbol{\Sigma}_0^{s-1}]ds - \mathrm{tr}(\boldsymbol{A})\right] \\ &+ \int_0^1\int_0^s \mathrm{tr}[(\boldsymbol{A}-\boldsymbol{A}_0)\boldsymbol{\Sigma}_0^{u-s}(\boldsymbol{A}-\boldsymbol{A}_0)\boldsymbol{\Sigma}_0^{-u}\boldsymbol{S}\boldsymbol{\Sigma}_0^{s-1}]duds. \qquad (2.7)\end{aligned}$$

We will show later in Section 3 that our Log-ME method using (2.7) provides an estimate of $\boldsymbol{A}$, that gives the same value for $l_n(\boldsymbol{A})$ in (2.7) and $L_n(\boldsymbol{A})$ in (2.2).

The integrations in (2.7) can be analytically solved through the spectral decomposition of $\boldsymbol{\Sigma}_0 = \boldsymbol{T}_0\boldsymbol{D}_0\boldsymbol{T}_0'$. Here $\boldsymbol{D}_0 = diag(d_1^{(0)}, \ldots, d_p^{(0)})$ with $d_i^{(0)}$'s as the eigenvalues of $\boldsymbol{\Sigma}_0$, and $\boldsymbol{T}_0 = (\boldsymbol{t}_1^{(0)}, \ldots, \boldsymbol{t}_p^{(0)})$ with $\boldsymbol{t}_i^{(0)}$ as the corresponding eigenvector for $d_i^{(0)}$. Define $\boldsymbol{B} = \boldsymbol{T}_0'(\boldsymbol{A}-\boldsymbol{A}_0)\boldsymbol{T}_0 = (b_{ij})_{p\times p}$, and $\tilde{\boldsymbol{S}} = \boldsymbol{T}_0'\boldsymbol{S}\boldsymbol{T}_0 = (\tilde{s}_{ij})_{p\times p}$. First, we can obtain,

$$\begin{aligned}\int_0^1 \mathrm{tr}[(\boldsymbol{A}-\boldsymbol{A}_0)\boldsymbol{\Sigma}_0^{-s}\boldsymbol{S}\boldsymbol{\Sigma}_0^{s-1}]ds - \mathrm{tr}(\boldsymbol{A}) &= \int_0^1 \mathrm{tr}[\boldsymbol{B}\boldsymbol{D}_0^{-s}\tilde{\boldsymbol{S}}\boldsymbol{D}_0^{s-1}]ds - \mathrm{tr}(\boldsymbol{A}) \\ &= Const + \sum_{i=1}^p \beta_{ii}b_{ii} + 2\sum_{i<j}\beta_{ij}b_{ij}, \qquad (2.8)\end{aligned}$$

where $Const$ denotes a constant and

$$\beta_{ii} = \frac{\tilde{s}_{ii}}{d_i^{(0)}} - 1,\ \beta_{ij} = \frac{\tilde{s}_{ij}(d_i^{(0)} - d_j^{(0)})/(d_i^{(0)}d_j^{(0)})}{(\log d_i^{(0)} - \log d_j^{(0)})}.$$

For the term that involves the double integral in (2.7), we have

$$\begin{aligned}\int_0^1\int_0^s \mathrm{tr}[(\boldsymbol{A}-\boldsymbol{A}_0)\boldsymbol{\Sigma}_0^{u-s}(\boldsymbol{A}-\boldsymbol{A}_0)\boldsymbol{\Sigma}_0^{-u}\boldsymbol{S}\boldsymbol{\Sigma}_0^{s-1}]duds &= \int_0^1\int_0^s \mathrm{tr}[\boldsymbol{B}\boldsymbol{D}_0^{u-s}\boldsymbol{B}\boldsymbol{D}_0^{-u}\tilde{\boldsymbol{S}}\boldsymbol{D}_0^{s-1}]duds \\ &= \sum_{i=1}^p \frac{1}{2}\xi_{ii}b_{ii}^2 + \sum_{i<j}\xi_{ij}b_{ij}^2 + 2\sum_{i=1}^p\sum_{j\neq i}\tau_{ij}b_{ii}b_{ij} + \sum_{k=1}^p\sum_{i<j,i\neq k,j\neq k}\eta_{kij}b_{ik}b_{kj}, \qquad (2.9)\end{aligned}$$

where,

$$\begin{aligned}\xi_{ii} &= \frac{\tilde{s}_{ii}}{d_i^{(0)}}, \\ \xi_{ij} &= \frac{\tilde{s}_{ii}/d_i^{(0)} - \tilde{s}_{jj}/d_j^{(0)}}{\log d_j^{(0)} - \log d_i^{(0)}} + \frac{(d_i^{(0)}/d_j^{(0)}-1)\tilde{s}_{ii}/d_i^{(0)} + (d_j^{(0)}/d_i^{(0)}-1)\tilde{s}_{jj}/d_j^{(0)}}{(\log d_j^{(0)} - \log d_i^{(0)})^2}, \\ \tau_{ij} &= \left[\frac{1/d_j^{(0)} - 1/d_i^{(0)}}{(\log d_j^{(0)} - \log d_i^{(0)})^2} + \frac{1/d_i^{(0)}}{\log d_j^{(0)} - \log d_i^{(0)}}\right]\tilde{s}_{ij}, \\ \eta_{kij} &= \left[\frac{1/d_i^{(0)} - 1/d_j^{(0)}}{\log(d_k^{(0)}/d_j^{(0)})\log(d_j^{(0)}/d_i^{(0)})} + \frac{1/d_j^{(0)} - 1/d_i^{(0)}}{\log(d_k^{(0)}/d_i^{(0)})\log(d_i^{(0)}/d_j^{(0)})} + \frac{2/d_k^{(0)} - 1/d_i^{(0)} - 1/d_j^{(0)}}{\log(d_k^{(0)}/d_i^{(0)})\log(d_k^{(0)}/d_j^{(0)})}\right]\tilde{s}_{ij}.\end{aligned}$$

Combing the equations in (2.7)-(2.9), we can rewrite $l_n(\boldsymbol{A})$ as a function of b_{ij}:

$$\begin{aligned}l_n(\boldsymbol{A}) = &\sum_{i=1}^p \frac{1}{2}\xi_{ii}b_{ii}^2 + \sum_{i<j}\xi_{ij}b_{ij}^2 + 2\sum_{i=1}^p\sum_{j\neq i}\tau_{ij}b_{ii}b_{ij} + \sum_{k=1}^p\sum_{i<j,i\neq k,j\neq k}\eta_{kij}b_{ik}b_{kj} \\ &-\left[\sum_{i=1}^p\beta_{ii}b_{ii} + 2\sum_{i<j}\beta_{ij}b_{ij}\right], \qquad (2.10)\end{aligned}$$

up to some constant. We see that $l_n(\boldsymbol{A})$ in (2.10) is a quadratic function of b_{ij}. Since the matrix $\boldsymbol{B}$ is a linear transformation of $\boldsymbol{A}$, $l_n(\boldsymbol{A})$ is also a quadratic function of $\boldsymbol{A}$.

Denote $\boldsymbol{a} = \mathrm{vec}(\boldsymbol{A}) = (a_{11}, \ldots, a_{pp}, a_{12}, \ldots, \alpha_{p-1,p}, \ldots, a_{1,p})'$, which is a q-dimensional vector with $q = p(p-1)/2$ consisting of the lower triangular elements of $\boldsymbol{A}$. Similarly, define $\boldsymbol{a}^{(0)} = \mathrm{vec}(\boldsymbol{A}_0)$. Recall that $\boldsymbol{B} = \boldsymbol{T}_0'(\boldsymbol{A}-\boldsymbol{A}_0)\boldsymbol{T}_0 = (b_{ij})_{p\times p}$, where $b_{ij} = (\boldsymbol{t}_i^{(0)})^T(\boldsymbol{A}-\boldsymbol{A}_0)\boldsymbol{t}_j^{(0)}$. It means that b_{ij} is a linear function of $\boldsymbol{a}$. Denote by $\boldsymbol{f}_{ij} = \boldsymbol{t}_i^{(0)} * \boldsymbol{t}_j^{(0)}$ the product of the eigenvectors $\boldsymbol{t}_i^{(0)}$ and $\boldsymbol{t}_j^{(0)}$ as an $q \times 1$ vector such that

$$(\boldsymbol{a}-\boldsymbol{a}^{(0)})^T(\boldsymbol{t}_i^{(0)} * \boldsymbol{t}_j^{(0)}) = (\boldsymbol{t}_i^{(0)})^T(\boldsymbol{A}-\boldsymbol{A}_0)\boldsymbol{t}_j^{(0)} = b_{ij}.$$

for all possible realizations of $\boldsymbol{A} - \boldsymbol{A}^{(0)}$. Then $l_n(\boldsymbol{A})$ in (2.10) can be rewritten as $l_n(\boldsymbol{a})$:

$$\begin{aligned} l_n(\boldsymbol{a}) &= -2(\boldsymbol{a} - \boldsymbol{a}^{(0)})^T \boldsymbol{\beta} + (\boldsymbol{a} - \boldsymbol{a}^{(0)})^T \boldsymbol{Q}(\boldsymbol{a} - \boldsymbol{a}^{(0)}) \\ &= Const + (\boldsymbol{a} - \boldsymbol{a}^{(0)} - \boldsymbol{Q}^{-1}\boldsymbol{\beta})^T \boldsymbol{Q}(\boldsymbol{a} - \boldsymbol{a}^{(0)} - \boldsymbol{Q}^{-1}\boldsymbol{\beta}). \end{aligned} \tag{2.11}$$

where

$$\boldsymbol{Q} = \frac{1}{2}\sum_{i=1}^{p} \xi_{ii} \boldsymbol{f}_{ii} \boldsymbol{f}_{ii}^T + \sum_{i<j} \xi_{ij} \boldsymbol{f}_{ij} \boldsymbol{f}_{ij}^T + 2\sum_{i=1}^{p} \sum_{j \neq i} \tau_{ij} \boldsymbol{f}_{ii} \boldsymbol{f}_{ij}^T + \sum_{k=1}^{p} \sum_{i<j, i\neq k, j \neq k} \eta_{kij} \boldsymbol{f}_{ik} \boldsymbol{f}_{kj}^T,$$

and

$$\boldsymbol{\beta} = \sum_{i=1}^{p} \frac{1}{2}\beta_{ii} \boldsymbol{f}_{ii} + \sum_{i<j} \beta_{ij} \boldsymbol{f}_{ij}.$$

The expression in (2.11) shows that $l_n(\boldsymbol{a})$ can be viewed as the log-density function of a multivariate normal distribution with mean vector $\boldsymbol{\alpha}^{(0)} + \boldsymbol{Q}^{-1}\boldsymbol{\beta}$ and covariance matrix $\boldsymbol{Q}$.

3 Penalized Covariance Matrix Estimation

In this section, we describe the proposed Log-ME method in estimating a covariance matrix. We propose a regularized approach to estimating $\boldsymbol{\Sigma}$ by using the approximate log-likelihood function $l_n(\boldsymbol{A})$ in (2.10). Consider the penalty function $\|\boldsymbol{A}\|_F^2$, the Frobenius norm of $\boldsymbol{A}$, which is equivalent to $\mathrm{tr}(\boldsymbol{A}^2)$. From (2.1), $\mathrm{tr}(\boldsymbol{A}^2) = \sum_{i=1}^{p}(\log(d_i))^2$, where d_i is the ith eigenvalue of the covariance matrix $\boldsymbol{\Sigma}$. If d_i goes to zero or diverges to infinity, the value of $\log(d_i)$ goes to infinity in both cases. Therefore, such a penalty function can *simultaneously* regularize the largest and smallest eigenvalues of the covariance matrix estimate. We consider to estimate $\boldsymbol{\Sigma}$, or equivalently $\boldsymbol{A}$, by minimizing

$$l_{n,\lambda}(\boldsymbol{A}) = l_n(\boldsymbol{A}) + \lambda \mathrm{tr}(\boldsymbol{A}^2), \tag{3.1}$$

where λ is a tuning parameter. Note that $\mathrm{tr}(\boldsymbol{A}^2) = \mathrm{tr}((\boldsymbol{T}_0\boldsymbol{B}\boldsymbol{T}_0^{'}+\boldsymbol{A}_0)^2)$ is equivalent to $\mathrm{tr}(\boldsymbol{B}^2)+2\mathrm{tr}(\boldsymbol{B}\boldsymbol{\Gamma})$ up to some constant, where $\boldsymbol{\Gamma} = (\gamma_{ij})_{p\times p} = \boldsymbol{T}_0^{'}\boldsymbol{A}_0\boldsymbol{T}_0$. In terms of $\boldsymbol{B}$, the function in (3.1) becomes

$$\begin{aligned} l_{n,\lambda}(\boldsymbol{B}) = & \sum_{i=1}^{p}\frac{1}{2}\xi_{ii}b_{ii}^2 + \sum_{i<j}\xi_{ij}b_{ij}^2 + 2\sum_{i=1}^{p}\sum_{j\neq i}\tau_{ij}b_{ii}b_{ij} + \sum_{k=1}^{p}\sum_{i<j,i\neq k,j\neq k}\eta_{kij}b_{ik}b_{kj} \\ & - \left(\sum_{i=1}^{p}\beta_{ii}b_{ii} + 2\sum_{i<j}\beta_{ij}b_{ij}\right) \\ & + \lambda\left[\frac{1}{2}\sum_{i=1}^{p}b_{ii}^2 + \sum_{i<j}^{p}b_{ij}^2 + \sum_{i=1}^{p}\gamma_{ii}b_{ii} + 2\sum_{i<j}\gamma_{ij}b_{ij}\right]. \end{aligned} \tag{3.2}$$

Let $\hat{\boldsymbol{B}}$ be the minimizer of (3.2). Then $\hat{\boldsymbol{A}}$, an estimate of $\boldsymbol{A}$, can be obtained from $\hat{\boldsymbol{B}}$ through the relationship $\boldsymbol{A} = \boldsymbol{T}_0\boldsymbol{B}\boldsymbol{T}_0^{'} + \boldsymbol{A}_0$. We can then estimate $\boldsymbol{\Sigma}$ by using $\exp(\hat{\boldsymbol{A}})$. Note that the expression in (3.2) depends on an initial estimate $\boldsymbol{\Sigma}_0$, or equivalently, $\boldsymbol{A}_0$. We propose to iteratively use (3.2) to obtain $\hat{\boldsymbol{B}}$, the minimizer of (3.2). Specifically, the iterative algorithm is described as follows:

Algorithm 1.

Step 1: *Set an initial covariance matrix estimate* $\boldsymbol{\Sigma}_0$, *a positive definite matrix.*

Step 2: *Obtain the spectral decomposition* $\boldsymbol{\Sigma}_0 = \boldsymbol{T}_0\boldsymbol{D}_0\boldsymbol{T}_0'$, *and set* $\boldsymbol{A}_0 = \log(\boldsymbol{\Sigma}_0)$.

Step 3: *Compute* $\hat{\boldsymbol{B}}$ *by minimizing* $l_{n,\lambda}$ *in* (3.2). *Then obtain* $\hat{\boldsymbol{A}} = \boldsymbol{T}_0\hat{\boldsymbol{B}}\boldsymbol{T}_0^{'} + \boldsymbol{A}_0$, *and update the estimate of* $\boldsymbol{\Sigma}$ *by*

$$\hat{\boldsymbol{\Sigma}} = \exp(\hat{\boldsymbol{A}}) = \exp(\boldsymbol{T}_0\hat{\boldsymbol{B}}\boldsymbol{T}_0^{'} + \boldsymbol{A}_0).$$

Step 4: *Check if* $||\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}_0||_F^2$ *is less than a pre-specified positive tolerance value. Otherwise, set* $\boldsymbol{\Sigma}_0 = \hat{\boldsymbol{\Sigma}}$ *and go back to* **Step 2**.

We set the default initial $\boldsymbol{\Sigma}_0$ in **Step 1** to be $\boldsymbol{S}+\epsilon\boldsymbol{I}$, where $\boldsymbol{S}$ is the sample covariance matrix and ϵ is a pre-specified small positive value. Note that estimating $\boldsymbol{B}$ in **Step 3** may not be trivial as the dimension of $\boldsymbol{B}$ increases. Here we adopt the *shooting* algorithm (Fu, 1998) to estimate $\boldsymbol{B}$, which updates each entry of $\boldsymbol{B}$ iteratively until convergence. Since $l_{n,\lambda}(\boldsymbol{B})$ in (3.2) is a quadratic function of $\boldsymbol{B}$, there is a close-form solution to update one entry of $\boldsymbol{B}$ at each step of the *shooting*.

The advantage of our proposed iterative algorithm is that $\hat{\boldsymbol{A}}$ can be improved step-by-step as the initial estimate $\boldsymbol{\Sigma}_0$ is updated in each iteration. The proposition below states the fact that the solution $\hat{\boldsymbol{A}}$ of the proposed iterative algorithm provides the same quantity for $l_{n,\lambda}(\boldsymbol{A})$ in (3.1) and $L_{n,\lambda}(\boldsymbol{A}) \equiv L_n(\boldsymbol{A}) + \lambda \mathrm{tr}(\boldsymbol{A}^2)$.

Proposition 1. *Suppose $\hat{\boldsymbol{A}}$ is the solution of the proposed iterative algorithm when it converges. Then the value of $l_n(\hat{\boldsymbol{A}})$ in* (2.10) *is exactly the same as the value of $L_n(\hat{\boldsymbol{A}})$ in* (2.2)*, i.e., $l_n(\hat{\boldsymbol{A}}) = L_n(\hat{\boldsymbol{A}})$.*

Note that $L_{n,\lambda}(\boldsymbol{A})$ is the exact penalized log-likelihood function of $\boldsymbol{A}$. Because of the complicated matrix exponential term in its expression, it is very difficult to estimate $\boldsymbol{A}$ by directly minimizing $L_{n,\lambda}(\boldsymbol{A})$. We instead estimate $\boldsymbol{A}$ based on the approximate log-likelihood function $l_{n,\lambda}(\boldsymbol{A})$ in (3.1) by using the proposed iterative algorithm. The proposition indicates that when the proposed iterative algorithm converges at $\hat{\boldsymbol{A}}$, $l_n(\boldsymbol{A})$ can provide a good approximation to the likelihood function $L_n(\boldsymbol{A})$ around $\hat{\boldsymbol{A}}$. That is, $l_{n,\lambda}(\boldsymbol{A})$ can approximate $L_{n,\lambda}(\boldsymbol{A})$ well in a neighborhood region of $\hat{\boldsymbol{A}}$. Moreover, by iteratively updating the estimate of $\boldsymbol{A}$ in the proposed iterative algorithm, the solution $\hat{\boldsymbol{A}}$ can estimate the true $\boldsymbol{A}$ (denoted by $\tilde{\boldsymbol{A}}$) more accurately. The value of $L_n(\hat{\boldsymbol{A}})$ will be close to the optimal $L_n(\tilde{\boldsymbol{A}})$. To elaborate this point, simulate $\boldsymbol{x}_i$'s from a p-dimensional multivariate normal distribution $N(\boldsymbol{0}, \boldsymbol{I})$. That is, $\tilde{\boldsymbol{A}} = \log(\boldsymbol{I})$. The sample size is chosen to be $n = 20$, and the dimension $p = 30$. We obtain the estimate $\hat{\boldsymbol{A}}$ from the Log-ME method. To evaluate the estimation accuracy of $L_n(\hat{\boldsymbol{A}})$ in terms of $L_n(\tilde{\boldsymbol{A}})$, define a *Relative Loss* of $L_n(\hat{\boldsymbol{A}})$ to be

$$RL \equiv \frac{L_n(\tilde{\boldsymbol{A}}) - L_n(\hat{\boldsymbol{A}})}{L_n(\tilde{\boldsymbol{A}})} = \frac{L_n(\tilde{\boldsymbol{A}}) - l_n(\hat{\boldsymbol{A}})}{L_n(\tilde{\boldsymbol{A}})}. \tag{3.3}$$

This simulation is repeated 100 times for computing RL in (3.3). The values of RL's from the simulation indicate that the relative loss is around 6-9%, which shows that $L_n(\hat{\boldsymbol{A}})$ is close to the optimal likelihood $L_n(\tilde{\boldsymbol{A}})$.

4 Simulation Study

We conduct a simulation study to compare the Log-ME method with five other methods. We consider the data following a multivariate normal distribution $N(\mathbf{0}, \mathbf{\Sigma})$. Six different covariance models of $\mathbf{\Sigma} = (\sigma_{ij})_{p\times p}$ are used for comparison as follows.

- Model 1: $\mathbf{\Sigma} = \exp(\boldsymbol{A})$ where $\boldsymbol{A} = (a_{ij})_{p\times p}$ with $a_{ii} \sim N(0.25, 0.5^2)$ and $a_{ij} = a_{ji} \sim N(0, 0.5^2)$ for $i \neq j$. This covariance matrix is not sparse and is not banded.

- Model 2: $\mathbf{\Sigma}$ is constructed from an MA(2) model with $\sigma_{ii} = 1, \sigma_{i,i-1} = \sigma_{i-1,i} = 0.6, \sigma_{i,i-2} = \sigma_{i-2,i} = 0.3$, and all the other σ_{ij} are zeros. This covariance matrix is sparse and banded.

- Model 3: $\mathbf{\Sigma} = \boldsymbol{PMP}'$, where $\boldsymbol{M}$ is the covariance matrix defined in model 2, and $\boldsymbol{P}$ is a $p \times p$ matrix obtained by randomly permuting the rows of a $p \times p$ identity matrix. The covariance matrix in this model is sparse but not banded.

- Model 4: $\mathbf{\Sigma}^{-1} = (c_{ij})_{p\times p}$ with $c_{ii} = 1$ and $c_{ij} = 0.3$ if $i \neq j$. The inverse covariance matrix in this case is not sparse and is not banded.

- Model 5: $\mathbf{\Sigma} = \boldsymbol{PHP}'$, where $\boldsymbol{H}$ is constructed from an MA(1) model with $h_{ii} = 1, h_{i,i-1} = h_{i-1,i} = 0.4$, and all other h_{ij} are zeros, and $\boldsymbol{P}$ is a $p \times p$ matrix obtained by randomly permuting the rows of a $p \times p$ identity matrix. The covariance matrix in this model is very sparse but not banded.

- Model 6: $\mathbf{\Sigma} = \boldsymbol{GG}$ where $\boldsymbol{G}$ is the covariance matrix defined in model 3. The resulting covariance matrix is not sparse and is not banded.

Five covariance matrix estimation methods to be compared are (1) the LW estimate in Ledoit and Wolf (2006), (2) the banding estimate (denoted by Banding) in Rothman et al. (2010), (3) the graphical Lasso estimate (Glasso) in Yuan and Lin (2007), Friedman et al. (2008), (4) the estimate with a condition number constraint (denoted by CN) in Won et al. (2009), and (5) the

CLIME estimate in Cai et al. (2011). Ledoit and Wolf (2006) proposed to estimate $\boldsymbol{\Sigma}$ by a linear combination of the sample covariance matrix $\boldsymbol{S}$ and the identity matrix $\boldsymbol{I}$, i.e.,

$$\hat{\boldsymbol{\Sigma}}_{LW} = \tau\nu\boldsymbol{S} + (1-\nu)\boldsymbol{I},$$

where the optimal values of τ and ν are obtained by minimizing the Frobenius norm $E(\|\hat{\boldsymbol{\Sigma}}_{LW} - \boldsymbol{\Sigma}\|)$. Rothman et al. (2010) considers a banded estimate of $\boldsymbol{\Sigma}$ by using the Cholesky decomposition of $\boldsymbol{\Sigma} = \boldsymbol{LDL}'$, where $\boldsymbol{L}$ is the Cholesky factor and $\boldsymbol{D}$ is the diagonal matrix. The banding estimate $\hat{\boldsymbol{\Sigma}}$ is obtained by only estimating the first k subdiagonals of $\boldsymbol{L}$ and set the rest to zero, where k is a tuning parameter. The Glasso considers to estimate $\boldsymbol{\Sigma}^{-1} = (c_{ij})$ by minimizing

$$-\log|\boldsymbol{\Sigma}^{-1}| + \mathrm{tr}[\boldsymbol{\Sigma}^{-1}\boldsymbol{S}] + \lambda\sum_{i\neq j}|c_{ij}|,$$

where λ is a tuning parameter. Won et al. (2009) proposed to estimate $\boldsymbol{\Sigma}$ with a constraint on its condition number. Denote $u_1,\ldots,u_p$ to be the eigenvalues of $\boldsymbol{\Sigma}^{-1}$. They consider $\hat{\boldsymbol{\Sigma}} = \boldsymbol{T}\mathrm{diag}(\hat{u}_1^{-1},\ldots,\hat{u}_p^{-1})\boldsymbol{T}'$, where $\boldsymbol{T}$ is from the spectral decomposition of $\boldsymbol{S} = \boldsymbol{T}\mathrm{diag}(l_1,\ldots,l_p)\boldsymbol{T}'$. The $\hat{u}_1,\ldots,\hat{u}_p$ are obtained by solving the constraint optimization

$$\begin{aligned}\min_{u,u_1,\ldots,u_p} \quad & \sum_{i}^{p}(l_iu_i - \log u_i)\\ s.t. \quad & u \le u_i \le \kappa_{max}u,\ i = 1,\ldots,p,\end{aligned}$$

where κ_{max} is a tuning parameter. For the CLIME method, Cai et al. (2011) estimate the inverse covariance matrix $\boldsymbol{\Sigma}^{-1} = (c_{ij})$ by solving

$$\min_{\boldsymbol{\Sigma}^{-1}}\sum_{i,j}|c_{ij}| \text{ s.t. } |\boldsymbol{S}\boldsymbol{\Sigma}^{-1} - \boldsymbol{I}|_\infty \le \lambda,$$

where λ is a tuning parameter. Note that all six methods in comparison involve the tuning parameter. Common methods of choosing tuning parameters include cross-validation (Bickel and Levian, 2008b), information criteria such as the Bayesian information criterion (Yuan and Lin, 2007), and

Table 1: Simulation results for Models 1 and 2. Averages and standard errors are calculated from 100 runs.

		Model 1					Model 2				
p	Method	KL	EN	$Fnorm$	$\Delta_{1/p}$	Δ_1	KL	EN	$Fnorm$	$\Delta_{1/p}$	Δ_1
p=25	Log-ME	7.60	5.37	7.58	22.42	2.83	5.10	6.17	2.78	11.87	0.18
		(0.03)	(0.04)	(0.07)	(1.09)	(0.10)	(0.04)	(0.07)	(0.01)	(0.23)	(0.01)
	Glasso	6.71	9.58	8.28	60.66	2.01	4.58	6.69	3.26	9.31	1.41
		(0.06)	(0.14)	(0.09)	(0.67)	(0.16)	(0.04)	(0.09)	(0.04)	(0.25)	(0.05)
	LW	6.52	12.39	7.18	75.08	1.92	5.88	9.42	2.90	17.20	0.30
		(0.05)	(0.15)	(0.06)	(0.33)	(0.11)	(0.04)	(0.09)	(0.02)	(0.12)	(0.03)
	CN	8.25	22.04	8.90	77.15	2.49	6.67	12.53	3.79	16.25	1.63
		(0.03)	(0.17)	(0.09)	(0.14)	(0.19)	(0.04)	(0.12)	(0.04)	(0.07)	(0.05)
	Banding	13.99	25.86	10.73	45.31	3.41	2.36	2.00	1.63	17.01	0.32
		(0.12)	(0.93)	(0.09)	(2.42)	(0.16)	(0.05)	(0.04)	(0.02)	(1.00)	(0.02)
	CLIME	16.48	51.38	13.11	86.91	8.38	4.03	5.92	2.41	12.72	0.36
		(0.02)	(1.74)	(0.01)	(0.06)	(0.05)	(0.04)	(0.08)	(0.02)	(0.24)	(0.03)
p=50	Log-ME	39.27	22.50	32.48	700.81	17.00	11.01	12.92	4.78	19.34	0.15
		(0.35)	(0.07)	(0.15)	(1.57)	(0.21)	(0.06)	(0.15)	(0.01)	(0.09)	(0.01)
	Glasso	31.67	82.94	31.08	759.34	6.11	11.97	21.61	5.47	11.57	2.27
		(0.16)	(1.27)	(0.30)	(1.30)	(0.53)	(0.05)	(0.15)	(0.04)	(0.22)	(0.05)
	LW	30.43	129.24	26.37	803.22	5.80	17.04	30.49	4.94	20.91	0.41
		(0.09)	(1.01)	(0.18)	(0.56)	(0.33)	(0.05)	(0.19)	(0.01)	(0.10)	(0.03)
	CN	38.57	259.98	31.49	810.96	6.03	18.20	37.30	5.37	21.89	0.64
		(0.09)	(1.84)	(0.27)	(0.25)	(0.54)	(0.04)	(0.24)	(0.07)	(0.22)	(0.10)
	Banding	59.59	454.63	44.91	774.41	19.88	4.80	4.08	2.32	21.79	0.43
		(0.14)	(4.47)	(0.11)	(3.52)	(0.25)	(0.07)	(0.05)	(0.03)	(0.82)	(0.02)
	CLIME	65.37	706.83	49.32	827.68	28.85	10.59	17.30	3.89	18.18	0.37
		(0.07)	(11.48)	(0.01)	(0.04)	(0.08)	(0.08)	(0.25)	(0.02)	(0.12)	(0.02)
p=100	Log-ME	144.07	297.47	246.94	11739.00	119.92	42.59	46.29	7.80	13.09	0.11
		(0.90)	(0.89)	(0.36)	(0.79)	(0.39)	(0.05)	(0.30)	(0.01)	(0.03)	(0.01)
	Glasso	139.37	1225.96	188.32	11635.54	56.26	28.87	59.59	8.90	11.25	3.41
		(0.39)	(13.51)	(1.31)	(2.74)	(2.64)	(0.08)	(0.25)	(0.06)	(0.19)	(0.05)
	LW	139.68	2394.90	154.71	11761.00	18.77	43.13	86.18	7.92	22.91	0.39
		(0.21)	(16.65)	(0.68)	(0.74)	(1.32)	(0.04)	(0.32)	(0.01)	(0.05)	(0.02)
	CN	168.92	4993.90	190.06	11772.00	50.91	44.70	109.27	8.04	24.69	0.46
		(0.21)	(23.83)	(1.30)	(0.29)	(2.58)	(0.04)	(0.40)	(0.03)	(0.03)	(0.01)
	Banding	223.59	7564.70	265.54	11756.00	111.72	10.12	8.49	3.34	32.35	0.60
		(0.20)	(44.39)	(0.29)	(2.83)	(0.64)	(0.10)	(0.07)	(0.02)	(1.15)	(0.03)
	CLIME	272.93	3593.36	281.74	11797.90	146.34	26.13	49.65	6.29	20.72	0.83
		(0.17)	(5.00)	(0.02)	(0.44)	(0.01)	(0.12)	(0.46)	(0.02)	(0.08)	(0.01)

the independent validation set method (Levina et al, 2008). In this work, the tuning parameter in each method is selected through the independent validation set method under the likelihood loss, which uses the same sample size as the training set.

To evaluate the performance of each method, we consider five loss functions. The first three loss functions are the Kullback-Leibler (KL) loss, the entropy loss (EN), and the Frobenius norm (Fnorm), which are

$$KL = -\log|\hat{\mathbf{\Sigma}}^{-1}| + \mathrm{tr}(\hat{\mathbf{\Sigma}}^{-1}\mathbf{\Sigma}) - (-\log|\mathbf{\Sigma}^{-1}| + p),$$

Table 2: Simulation results for Models 3 and 4. Averages and standard errors are calculated from 100 runs.

		Model 3					Model 4				
p	Method	KL	EN	$Fnorm$	$\Delta_{1/p}$	Δ_1	KL	EN	$Fnorm$	$\Delta_{1/p}$	Δ_1
p=25	Log-ME	4.21	6.31	2.80	11.65	0.18	2.66	5.35	2.09	9.13	0.09
		(0.05)	(0.08)	(0.01)	(0.27)	(0.01)	(0.01)	(0.02)	(0.00)	(0.02)	(0.01)
	Glasso	4.52	6.64	3.25	9.39	1.37	2.34	9.55	2.73	9.27	1.26
		(0.04)	(0.09)	(0.03)	(0.23)	(0.05)	(0.02)	(0.11)	(0.03)	(0.04)	(0.03)
	LW	5.95	9.34	2.96	17.15	0.34	1.56	6.94	1.33	10.42	0.18
		(0.04)	(0.08)	(0.02)	(0.13)	(0.03)	(0.01)	(0.06)	(0.01)	(0.02)	(0.01)
	CN	6.67	12.44	3.77	16.26	1.61	1.62	7.15	1.41	10.53	0.15
		(0.03)	(0.10)	(0.03)	(0.08)	(0.05)	(0.01)	(0.04)	(0.01)	(0.01)	(0.01)
	Banding	12.47	21.44	4.12	18.79	0.59	2.12	8.18	1.88	9.42	0.51
		(0.10)	(0.76)	(0.05)	(1.30)	(0.04)	(0.02)	(0.04)	(0.01)	(0.04)	(0.02)
	CLIME	4.07	6.09	2.39	13.28	0.29	1.63	8.30	1.41	10.58	0.07
		(0.04)	(0.09)	(0.02)	(0.22)	(0.03)	(0.01)	(0.08)	(0.00)	(0.02)	(0.00)
p=50	Log-ME	15.96	17.95	4.77	17.30	0.15	5.09	13.66	3.02	20.09	0.18
		(0.07)	(0.18)	(0.01)	(0.10)	(0.01)	(0.00)	(0.02)	(0.00)	(0.01)	(0.01)
	Glasso	11.90	21.55	5.40	11.78	2.23	4.06	23.46	4.06	19.82	1.53
		(0.06)	(0.15)	(0.04)	(0.20)	(0.05)	(0.03)	(0.18)	(0.04)	(0.04)	(0.03)
	LW	17.01	30.81	4.92	21.00	0.39	2.22	16.65	1.45	21.20	0.18
		(0.05)	(0.17)	(0.01)	(0.08)	(0.03)	(0.01)	(0.06)	(0.01)	(0.01)	(0.01)
	CN	21.91	58.88	7.54	19.37	2.93	2.19	18.29	1.41	21.43	0.03
		(0.29)	(2.17)	(0.06)	(0.26)	(0.06)	(0.02)	(0.07)	(0.01)	(0.04)	(0.00)
	Banding	28.21	66.43	6.75	21.38	1.00	3.36	18.55	2.41	19.86	0.64
		(0.05)	(0.41)	(0.01)	(0.30)	(0.03)	(0.03)	(0.07)	(0.02)	(0.04)	(0.02)
	CLIME	10.58	17.65	3.89	18.23	0.37	2.29	19.40	1.55	21.34	0.11
		(0.06)	(0.22)	(0.02)	(0.13)	(0.02)	(0.01)	(0.07)	(0.01)	(0.01)	(0.00)
p=100	Log-ME	42.89	46.74	7.81	21.95	0.18	9.34	30.86	4.29	32.63	0.24
		(0.07)	(0.06)	(0.01)	(0.02)	(0.01)	(0.00)	(0.02)	0.01	0.04	0.04
	Glasso	28.83	59.99	8.89	11.19	3.51	7.05	54.22	6.00	41.12	1.83
		(0.08)	(0.29)	(0.05)	(0.15)	(0.05)	(0.04)	(0.21)	(0.04)	(0.04)	(0.03)
	LW	43.11	86.86	7.91	22.94	0.38	2.99	36.49	1.64	42.62	0.22
		(0.05)	(0.42)	(0.01)	(0.05)	(0.03)	(0.01)	(0.10)	(0.02)	(0.01)	(0.01)
	CN	45.33	102.76	9.45	22.75	0.73	3.72	41.54	2.43	42.85	0.20
		(0.05)	(0.39)	(0.02)	(0.01)	(0.01)	(0.03)	(0.57)	(0.04)	(0.01)	(0.01)
	Banding	57.40	140.69	9.64	24.30	1.19	5.14	39.84	3.15	41.07	0.76
		(0.05)	(0.50)	(0.00)	(0.16)	(0.02)	(0.04)	(0.09)	(0.02)	(0.03)	(0.02)
	CLIME	25.86	49.34	6.25	20.52	0.81	2.90	40.53	1.54	42.68	0.11
		(0.11)	(0.39)	(0.02)	(0.08)	(0.01)	(0.01)	(0.02)	(0.02)	(0.01)	(0.03)

Table 3: Simulation results for Models 5 and 6. Averages and standard errors are calculated from 100 runs.

		Model 5					Model 6				
p	Method	KL	EN	$Fnorm$	$\Delta_{1/p}$	Δ_1	KL	EN	$Fnorm$	$\Delta_{1/p}$	Δ_1
p=25	Log-ME	3.54	4.75	2.19	3.84	0.11	7.78	8.74	6.08	310.31	0.80
		(0.02)	(0.03)	(0.01)	(0.04)	(0.01)	(0.11)	(0.16)	(0.06)	(1.03)	(0.05)
	Glasso	3.34	5.11	2.64	2.88	1.11	8.33	11.55	7.15	390.85	3.21
		(0.03)	(0.07)	(0.03)	(0.10)	(0.03)	(0.10)	(0.31)	(0.10)	(6.36)	(0.13)
	LW	3.77	5.52	2.26	5.74	0.20	19.01	120.73	6.16	593.85	1.04
		(0.02)	(0.05)	(0.01)	(0.05)	(0.02)	0.64	(1.08)	(0.06)	(0.36)	(0.10)
	CN	4.62	6.45	3.37	3.69	1.26	27.32	298.57	7.69	602.81	2.93
		(0.03)	(0.04)	(0.02)	(0.02)	(0.01)	(0.07)	(2.34)	(0.09)	(0.11)	(0.13)
	Banding	5.96	10.29	2.95	5.92	0.37	17.52	21.28	6.72	1009.70	2.18
		(0.03)	(0.08)	(0.07)	(0.13)	(0.02)	(0.26)	(2.21)	(0.08)	(56.82)	(0.13)
	CLIME	2.70	3.87	1.83	3.55	0.22	15.71	57.57	7.86	591.05	2.62
		(0.03)	(0.08)	(0.02)	(0.11)	(0.02)	(0.08)	(0.63)	(0.03)	(0.16)	(0.05)
p=50	Log-ME	8.81	13.42	2.71	6.20	0.08	32.68	53.66	11.13	638.58	0.46
		(0.02)	(0.06)	(0.01)	(0.02)	(0.00)	(0.21)	(0.35)	(0.04)	(0.83)	(0.04)
	Glasso	7.94	13.76	4.13	2.88	1.52	27.69	76.68	13.55	612.60	5.82
		(0.04)	(0.11)	(0.03)	(0.08)	(0.03)	(0.13)	(0.89)	(0.10)	(1.77)	(0.13)
	LW	9.04	14.62	3.49	6.39	0.14	51.73	441.72	11.05	718.69	1.53
		(0.02)	(0.79)	(0.05)	(0.32)	(0.12)	(0.08)	(2.81)	(0.05)	(0.19)	(0.09)
	CN	12.81	23.52	6.83	2.68	2.78	66.63	975.69	15.46	721.74	5.94
		(0.04)	(0.58)	(0.03)	(0.19)	(0.03)	(0.77)	(37.18)	(0.15)	(0.36)	(0.13)
	Banding	12.08	21.56	4.21	6.28	0.31	80.44	1025.60	17.13	519.25	1.74
		(0.04)	(0.10)	(0.01)	(0.07)	(0.01)	(0.25)	(12.94)	(0.07)	(15.04)	(0.08)
	CLIME	6.45	10.48	2.78	4.58	0.10	36.59	98.77	13.42	700.81	2.66
		(0.04)	(0.13)	(0.01)	(0.05)	(0.01)	(0.13)	(0.89)	(0.04)	(0.62)	(0.07)
p=100	Log-ME	17.40	28.76	5.84	2.03	0.59	80.32	479.83	19.89	750.95	1.26
		(0.03)	(0.03)	(0.00)	(0.01)	(0.01)	(0.29)	(5.08)	(0.04)	(0.28)	(0.05)
	Glasso	18.05	34.31	6.38	2.51	2.05	77.17	344.15	24.84	679.05	9.97
		(0.05)	(0.15)	(0.04)	(0.07)	(0.03)	(0.18)	(2.31)	(0.14)	(0.87)	(0.14)
	LW	20.11	34.35	5.28	6.68	0.14	130.51	1377.90	18.98	759.12	2.08
		(0.02)	(0.13)	(0.00)	(0.02)	(0.01)	(0.10)	(6.87)	(0.03)	(0.13)	(0.09)
	CN	28.66	67.26	13.13	2.46	4.42	158.69	3005.30	29.57	760.32	10.51
		(0.04)	(0.76)	(0.06)	(0.04)	(0.07)	(0.17)	(12.68)	(0.15)	(0.07)	(0.15)
	Banding	24.38	44.11	5.97	6.14	0.27	175.20	2797.80	26.43	753.26	3.74
		(0.04)	(0.16)	(0.00)	(0.04)	(0.01)	(0.10)	(10.56)	(0.10)	(0.74)	(0.07)
	CLIME	15.85	28.79	4.47	5.24	0.11	85.23	282.74	21.42	747.04	4.16
		(0.06)	(0.28)	(0.01)	(0.05)	(0.00)	(0.17)	(2.51)	(0.03)	(0.29)	(0.03)

$$EN = \mathrm{tr}(\mathbf{\Sigma}^{-1}\hat{\mathbf{\Sigma}}) - \log|\mathbf{\Sigma}^{-1}\hat{\mathbf{\Sigma}}| - p,$$

and

$$Fnorm = ||\hat{\mathbf{\Sigma}} - \mathbf{\Sigma}||_F = \sqrt{\sum_{i,j}(\hat{\sigma}_{ij} - \sigma_{ij})^{ij}}.$$

Recall that d_1 is the largest eigenvalue and d_p is the smallest eigenvalue of the covariance matrix $\mathbf{\Sigma}$. Denote $\hat{d}_1$ and $\hat{d}_p$ to be their estimates. The other two loss functions are

$$\Delta_{1/p} = |\hat{d}_1/\hat{d}_p - d_1/d_p| \text{ and } \Delta_1 = |\hat{d}_1 - d_1|.$$

For each covariance model, we generated the training set in three scenarios: (1) $n = 50, p = 25$, (2) $n = 50, p = 50$, and (3) $n = 50, p = 100$. The simulation is repeated 100 times and the means and standard errors (in the parenthesis) of the losses are reported in Tables 1–3.

According to Tables 1–3, Log-ME generally outperforms the five other methods under the EN loss. Under the loss functions $\Delta_{1/p}$ and Δ_1 for measuring the eigen-structure of the estimates, the Log-ME overall gives more accurate estimation. When $\mathbf{\Sigma}$ has a banding structure as described in model 2, the Banding method perform well as expected, but Log-ME gives smaller values of $\Delta_{1/p}$ and Δ_1. When $\mathbf{\Sigma}$ is sparse but it does not have a banding structure as described in models 3 and 5, the Log-ME can perform better than the Banding method. In the situation of $\mathbf{\Sigma}^{-1}$ not having a sparse structure as described in Model 4, methods such as Glasso may not perform very well since it encourages the sparsity on Σ^{-1}. CLIME appears to perform well under the losses of KL, EN and $Fnorm$, while Log-ME gives more accurate measures for EN and Δ_1. When neither $\mathbf{\Sigma}$ nor $\mathbf{\Sigma}^{-1}$ has a sparse structure as in model 6, Log-ME generally shows better performance than the other five methods. It appears that Glasso and CLIME give comparable measures under the KL, EN, and $\Delta_{1/p}$ losses in model 6.

5 Real Data Examples

We illustrate the use of Log-ME in two real applications: classification of the Ionosphere data and portfolio optimization of the stock data. These applications require the estimate of the covariance matrix to be well-conditioned. We also examine the use of Log-ME covariance matrix estimate compared with other covariance matrix estimation methods including the CN estimate, the LW estimate, the Glasso method, the Banding method, and the CLIME method.

5.1 Classification of Ionosphere Data

The first example deals with the Ionosphere data from UCI repository (Blake et al., 1998). This radar data was collected by a system consisting of a phased array of high-frequency antennas. The targets were free electrons in the ionosphere. “Good” radar returns are those showing evidence of some type of structure in the ionosphere. “Bad” returns are those do not have this type of structure. The data contains 351 observations with 34 variables. The response is labelled as 1 for “Good” and -1 for ”Bad” of radar returns from the ionosphere using neural networks (Sigillito et al., 1989). As a classification problem, we apply linear discriminant analysis (LDA) with an appropriate covariance matrix estimate. The purpose here is to illustrate how different covariance matrix estimates can affect the classification performance of LDA. We randomly divide the data into a training set, a validation set and a test set. The training set is used to estimate the covariance matrix. The validation set is used to choose the tuning parameter for the methods under comparison. The misclassification error is computed based on the test set. The sizes of the training set and the validation set are chosen to be $n = 40$. We compute the misclassification error of LDA using the covariance matrix estimates obtained by the Log-ME method, the CN method, the LW method, the Glasso method, the Banding method, the CLIME method, and the sample covariance matrix $\boldsymbol{S}$, respectively. Note that in this example, $p = 34$ is very close to $n = 40$. The comparison procedure is repeated 100 times and the boxplots of the misclassification errors are displayed in

Figure 1. We see that LDA with Log-ME and LDA with LW have comparable performance, and both have a lower misclassification error than the other five methods.

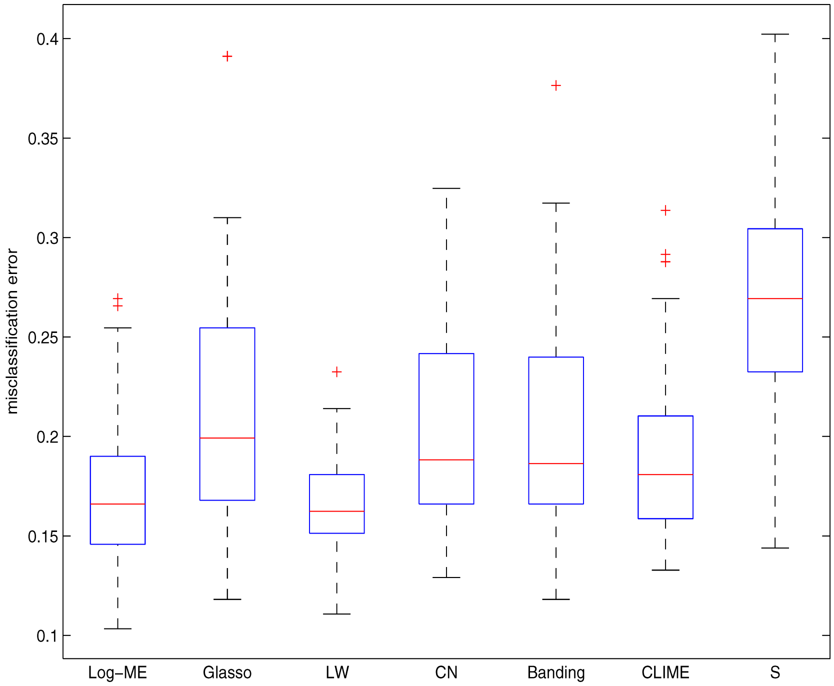

Figure 1: The boxplots of test misclassification errors from 100 replications.

To further demonstrate the merits of the Log-ME method to improve LDA for classification, we conduct a different simulation study for the unconditional misclassification error of LDA, where the class populations are constructed based on the ionosphere data. Let $\bar{\boldsymbol{x}}_1$, $\bar{\boldsymbol{x}}_2$, and $\boldsymbol{S}$ be the sample means and the sample covariance matrix from the ionosphere data. Define $\boldsymbol{\delta} = \bar{\boldsymbol{x}}_2 - \bar{\boldsymbol{x}}_1$. We construct two normal populations $N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$ and $N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$ by setting the population means to be $\boldsymbol{\mu}_1 = \bar{\boldsymbol{x}}_1$, $\boldsymbol{\mu}_2 = \boldsymbol{\mu}_1 + \boldsymbol{\delta} d$, and the population covariance matrix to be $\boldsymbol{\Sigma} = \boldsymbol{S}$. Here d is a scalar to control the distance between $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, and its value is varied as $d = 0.1, 0.3, \ldots, 1.5$.

For each value of d, we simulate a training set consisting of 40 data points for class 1 from $N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$ and 40 data points for class 2 from $N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$. The covariance matrix estimate is obtained from the training set. Since in simulation, we know the true population means and the true covariance matrix, the conditional misclassification error of LDA can be calculated explicitly as $\gamma(\hat{\boldsymbol{\Sigma}}, \hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\mu}}_2)$. In order to remove the effect of fluctuating values of $\hat{\boldsymbol{\mu}}_1$ and $\hat{\boldsymbol{\mu}}_2$, we use instead $\gamma(\hat{\boldsymbol{\Sigma}}) \equiv \gamma(\hat{\boldsymbol{\Sigma}}, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2)$ with the true $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ to examine the effect of different covariance matrix estimation methods. The same seven covariance matrix estimation methods were considered. This simulation is repeated 100 times and the unconditional misclassification error is approximated by averaging the 100 values of $\gamma(\hat{\boldsymbol{\Sigma}})$.

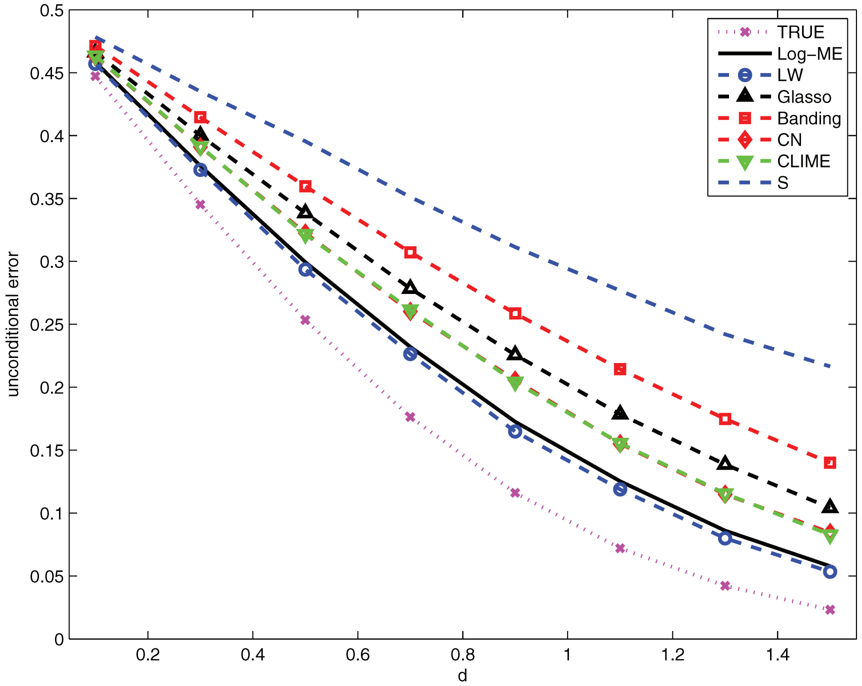

Figure 2: The unconditional misclassification errors of LDA over d, where the population mean of class 1 is $\boldsymbol{\mu}_1 = \bar{\boldsymbol{x}}_1$ and the population mean of class 2 is $\boldsymbol{\mu}_2 = \boldsymbol{\mu}_1 + \boldsymbol{\delta} d$.

Figure 2 shows the unconditional misclassification errors of LDA for various values of d. Note that the optimal unconditional misclassification error of LDA becomes smaller as the value of $\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\| = \|\boldsymbol{\delta}\| d$ increases. The results in Figure 2 show that LDA with Log-ME method performs much better than LDA with the CN method, the Glasso method, the Banding method, and the CLIME method. We observe that the unconditional errors of the CLIME and CN methods are very close in Figure 2. The Log-ME and LW methods perform comparably, and their unconditional misclassification errors of LDA are close to the optimal misclassification error.

5.2 Portfolio Optimization of Stock Data

In this section, we apply the Log-ME method in an application of portfolio optimization. In mean-variance optimization, the risk of a portfolio $\boldsymbol{w} = (w_1, \ldots, w_p)$ is measured by the standard deviation $\sqrt{\boldsymbol{w}^T \boldsymbol{\Sigma} \boldsymbol{w}}$ of its return (Markowitz, 1952), where $w_i \geq 0$ and $\sum_i^p w_i = 1$. The estimated minimum variance portfolio optimization problem is

$$\min_{\boldsymbol{w}} \boldsymbol{w}^T \hat{\boldsymbol{\Sigma}} \boldsymbol{w} \qquad (5.1)$$
$$\text{s.t.} \sum_i^p w_i = 1,$$

where $\hat{\boldsymbol{\Sigma}}$ is an estimate of the true covariance matrix $\boldsymbol{\Sigma}$. We expect that an accurate covariance matrix estimate $\hat{\boldsymbol{\Sigma}}$ will lead to a better portfolio strategy.

We consider the weekly returns of 30 components of the Dow Jones Industrial Index. The data of adjusted close prices of the weekly returns are extracted in the past three and a half years from January 8th, 2007 to June 28th, 2010, which are available from Yahoo! Finance (http://finance.yahoo.com). We used the first 50 observations of the weekly return data as the training set, the next 50 observations as the validation set, and *the remaining data* for the test set. Denote $\boldsymbol{X}_{ts}$ to be the test set and $\boldsymbol{S}_{ts}$ to be the sample covariance matrix of $\boldsymbol{X}_{ts}$. The performance of a portfolio $\boldsymbol{w}$ is measured by the *realized return*

$$R(\boldsymbol{w}) = \sum_{\boldsymbol{x} \in \boldsymbol{X}_{ts}} \boldsymbol{w}^T \boldsymbol{x}, \qquad (5.2)$$

and the *realized risk*

$$\sigma(\boldsymbol{w}) = \sqrt{\boldsymbol{w}^T \boldsymbol{S}_{ts} \boldsymbol{w}}. \qquad (5.3)$$

The optimal portfolio $\tilde{\boldsymbol{w}}$ in (5.1) is computed with $\hat{\boldsymbol{\Sigma}}$ estimated by the Log-ME method, the CN method, the LW method, the Glasso method, the Banding method, the CLIME method and the sample covariance matrix $\boldsymbol{S}$, separately. The realized returns and the realized risks for these seven

Table 4: The comparison of the realized return and the realized risk.

	Log-ME	CN	***S***	LW	Glasso	Banding	CLIME
Realized return $R(\tilde{\boldsymbol{w}})$	0.218	0.128	0.059	0.062	0.193	0.192	0.211
Realized risk $\sigma(\tilde{\boldsymbol{w}})$	0.029	0.024	0.035	0.025	0.028	0.029	0.029

methods are reported in Table 4. The Log-ME and CLIME methods gave a comparable performance in terms of the realized return and the realized risk. The Log-ME method produced a portfolio with a larger realized return $R(\tilde{\boldsymbol{w}})$ than the portfolio using the covariance matrix estimate from the remaining five methods. While the realized risk using the Log-ME method is comparable to that using the CN method, the LW method, the Glasso method and the Banding method. It appears that the Log-ME method improves the portfolio strategy resulting in more realized returns.

To provide further insight of the improvements of the portfolio strategy using the Log-ME method for covariance matrix estimation, we evaluate the performance of the portfolio $\tilde{\boldsymbol{w}}$ in different periods. Given a stating week, we use the first 50 observations of the weekly returns as the training set, the next 50 observations of the weekly returns as the validation set, and *the third* 50 *observations* of the weekly returns as the test set. By shifting the starting week one ahead every time from the week of January 8th, 2007 to the week of August 20th, 2007, we can evaluate the portfolio strategy of 33 different consecutive test periods, i.e., the first test period is from December 8th, 2008 to November 16th, 2009, and the last test period is from July 20th, 2009 to June 28th, 2010. We calculate the realized return in (5.2) and the realized risk in (5.3) for each test period using the optimal portfolio $\tilde{\boldsymbol{w}}$ based on the corresponding training set. The optimal portfolio $\tilde{\boldsymbol{w}}$ is computed with $\hat{\boldsymbol{\Sigma}}$ estimated by the same seven methods mentioned above. The realized returns and the realized risks for different test periods are shown in Figure 3 and Figure 4.

The results in Figures 3 and 4 show that Log-ME can lead to a better portfolio strategy with higher returns and lower risks than the sample covariance matrix. Note that the CLIME method has a slightly higher realized return than Log-ME, but it also gives higher realized risk. The log-ME method has relatively higher risks than the CN method, the LW method, the Glasso method and the

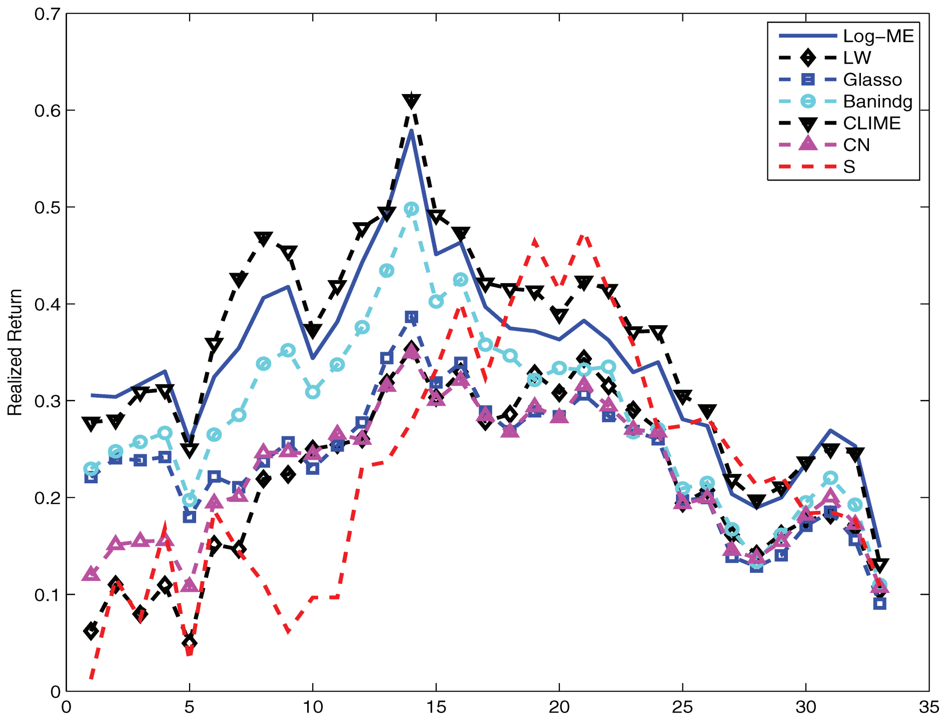

Figure 3: The comparison of realized returns in different test periods.

Banding method. But it provides much larger realized returns than these four methods. It indicates that the Log-ME method can give an overall better portfolio strategy.

6 Discussion

We have proposed a new covariance matrix estimation method, Log-ME, through the matrix logarithm based on a penalized likelihood function. Log-ME regularizes the largest and smallest eigenvalues simultaneously by imposing a convex penalty on the transformed likelihood function. It leads to an accurate estimate of the covariance matrix with a well-structured eigen-system. Moreover, the iterative nature of the Log-ME method can always improve the covariance matrix estimates proposed in the literature, as long as their proposed estimate is positive definite, by using their estimate as the initial estimate in our iterative algorithm. The Log-ME covariance matrix estimate results from the iterative quadratic programming algorithm in Section 3, which converges

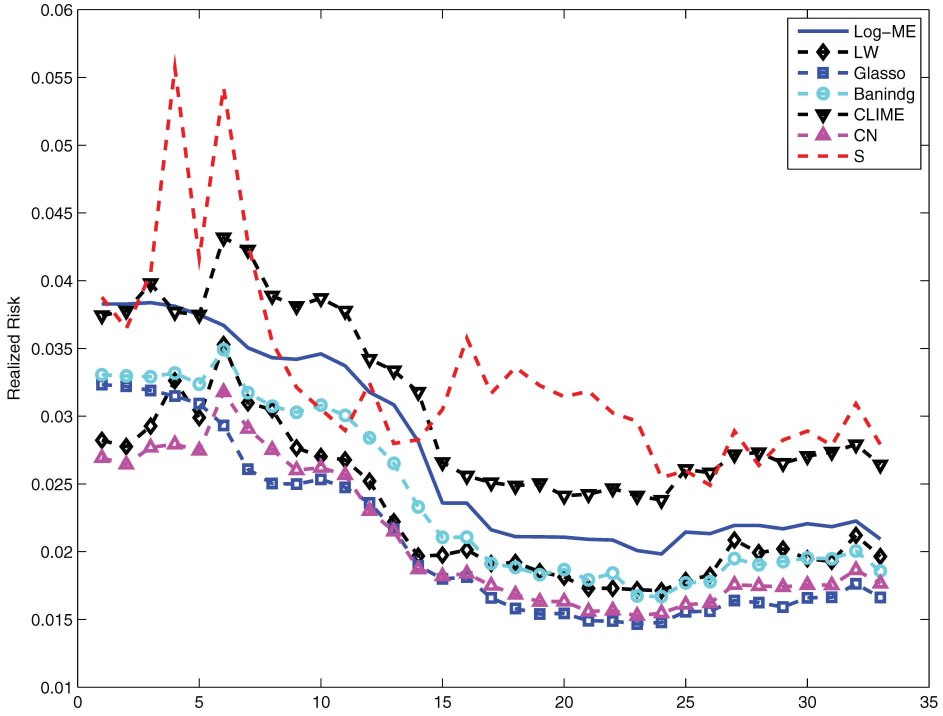

Figure 4: The comparison of realized risks in different test periods.

rather quickly in our study. Further rigorous investigation of the algorithm's convergence is of interest. Another line of research is to study asymptotic properties of the Log-ME covariance matrix estimate as both the sample size and the dimension of covariance matrix increase. Such a theoretical study is not trivial because the estimate is obtained through its matrix logarithm based on an approximate likelihood function. These studies will be reported elsewhere.

Leonard and Hsu's (1992) Bayesian covariance matrix estimation approach motivated our current study. One can use the new approximation method described in Section 2 to extend their Bayesian method. Our log-likelihood approximation in Section 3 is iterative while Leonard and Hsu's log-likelihood approximation is not. When the negative log-prior distribution is used as the penalty term, the iterative nature of our procedure allows us to actually obtain the optimal log covariance matrix, $\boldsymbol{A}$, that maximizes the posterior. For a given prior distribution of $\boldsymbol{a} = \text{vec}(\boldsymbol{A}) = (a_{11}, \ldots, a_{pp}, a_{12}, \ldots, \alpha_{p-1,p}, \ldots, a_{1,p})'$, one can derive an approximate posterior distribution of $\boldsymbol{a}$. As observed in Section 3, if the negative log prior density is proportional to $||\boldsymbol{A}||_F^2$, then the resulting posterior of $\boldsymbol{a}$ is approximately a multivariate normal distribution. Other prior distributions can be considered. One example is a class of priors with log-concave densities. Another example is a negative log prior density proportional to $\|\boldsymbol{A}\|_1 = \sum_{i,j} |a_{ij}|$, resulting in a L_1 penalty function for $\boldsymbol{A}$.

Supplemental Materials

Title: Matlab code for the proposed Log-ME method

Matlab_Log-ME.rar This zip file contains the matlab codes to implement the proposed Log-ME methods (Matlab codes) and the data set used in Section 5.2.

References

Banerjee, O., d'Aspremont, A., and Natsoulis, G. (2006). Convex optimization techniques for fitting sparse Gaussian graphical models. *Proceedings of the 23rd International Conference on Machine Learning*, 89–96.

Bellman, R. (1970). *Introduction to matrix analysis*, New York: McGraw-Hill.

Bickel, P. J. and Levina E. (2008a). Covariance regularization by thresholding. *Annals of Statistics*, **36**, 2577–2604.

Bickel, P. J. and Levina., E. (2008b). Regularized estimation of large covariance matrices. *Annals of Statistics*, **36**, 199–227.

Blake, C. L., Newman, D. J., Hettich, S., and Merz, C. J. (1998). UCI respository of machine learning databases.

Cai, T., Liu, W., and Luo, X. (2011). A constrained l_1 minimization approach to sparse precision matrix estimation. *Journal of American Statistical Association*, **106**, 594–607.

Chiu, T. Y. M., Leonard, T., and Tsui, K. W. (1996). The matrix-logarithmic covariance model. *Journal of the American Statistical Association*, **91**, 198–210.

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

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

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

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

Haff, L. R. (1991). The variational form of certain Bayes estimators. *Annals of Statistics*, **19**, 1163–1190.

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

Johnstone, I. M. (2001). On the distribution of the largest eigenvalue in principal components analysis. *Annals of Statistics*, **29**, 295–327.

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

Leonard, T. and Hsu, J. S. J. (1992). Bayesian inference for a covariance matrix. *Annals of Statistics*, **20**, 1669–1696.

Ledoit, O. and Wolf, M. (2004). 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**, 245–263.

Markowitz, H. (1952). Portfolio selection. *Journal of Finance*, **7**, 77–91.

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

Peng, J., Wang, P., Zhou, N., and Zhu, J. (2009). Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, **104**, 735–746.

Rajaratnam, B., Massam, H., and Carvalho, C. (2008). Flexible covariance estimation in graphical Gaussian models. *Annals of Statistics*, **36**, 2818–2849.

Rothman, A. J., Levina, E., and Zhu, J. (2009). Generalized thresholding of large covariance matrices. *Journal of the American Statistical Association*, **104**, 177–186.

Rothman, A. J., Levina, E., and Zhu, J. (2010). A new approach to Cholesky-based estimation of high-dimensional covariance matrices. *Biometrika*, **97(3)**, 539–550.

Sigillito, V. G., Wing, S. P., Hutton, L. V., and Baker, K. B. (1989). Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest*, 262–266.

Stein, C. (1975). Estimation of a covariance matrix. *Reitz Lecture*, IMS-ASA Annual Meeting.

Won, J. H., Lim, J., Kim, S. J., and Rajaratnam, B. (2009). Maximum likelihood covariance estimation with a condition number constraint. Technical Report, Department of Statistics, Stanford University.

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

Yuan, M. (2010). Sparse inverse covariance matrix estimation via linear programming. *Journal of Machine Learning Research*, **11**, 2261–2286.

Zhou, S., Rutimann, P., Xu, M., and Buhlmann, P. (2010). High-dimensional covariance estimation based on Gaussian graphical models. *arXiv:1009.0530*