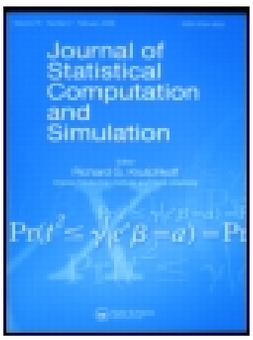

Journal of Statistical Computation and Simulation

ISSN: 0094-9655 (Print) 1563-5163 (Online) Journal homepage: https://www.tandfonline.com/loi/gscs20

An improved modified cholesky decomposition approach for precision matrix estimation

Xiaoning Kang & Xinwei Deng

To cite this article: Xiaoning Kang & Xinwei Deng (2019): An improved modified cholesky decomposition approach for precision matrix estimation, Journal of Statistical Computation and Simulation, DOI: 10.1080/00949655.2019.1687701

To link to this article: https://doi.org/10.1080/00949655.2019.1687701

JOURNAL OF STATISTICAL COMPUTATION AND SIMULATION
https://doi.org/10.1080/00949655.2019.1687701

An improved modified cholesky decomposition approach for precision matrix estimation

Xiaoning Kang [a] and Xinwei Deng [b]

[a]International Business College and Institute of Supply Chain Analytics, Dongbei University of Finance and Economics, Dalian, People's Republic of China; [b]Department of Statistics, Virginia Tech, Blacksburg, VA, USA

ABSTRACT

The modified Cholesky decomposition is commonly used for precision matrix estimation given a specified order of random variables. However, the order of variables is often not available or cannot be pre-determined. In this work, we propose a sparse precision matrix estimation by addressing the variable order issue in the modified Cholesky decomposition. The idea is to effectively combine a set of estimates obtained from multiple permutations of variable orders, and to efficiently encourage the sparse structure for the resultant estimate by the thresholding technique on the ensemble Cholesky factor matrix. The consistent property of the proposed estimate is established under some weak regularity conditions. Simulation studies are conducted to evaluate the performance of the proposed method in comparison with several existing approaches. The proposed method is also applied into linear discriminant analysis of real data for classification.

ARTICLE HISTORY
Received 18 January 2019
Accepted 29 October 2019

KEYWORDS
LDA; modified Cholesky decomposition; precision matrix; sparsity; order of variables

1. Introduction

The estimation of large sparse precision matrix is of fundamental importance in the multivariate analysis and various statistical applications. For example, in the classification problem, linear discriminant analysis (LDA) needs the precision matrix to compute the classification rule. In financial applications, portfolio optimization often involves the precision matrix in minimizing the portfolio risk. A sparse estimate of precision matrix not only provides a parsimonious model structure, but also gives meaningful interpretation on the conditional independence among the variables.

Suppose that $\boldsymbol{X} = (X_1, \ldots, X_p)'$ is a p-dimensional vector of random variables with an unknown covariance matrix $\boldsymbol{\Sigma}$. Without loss of generality, we assume that the expectation of $\boldsymbol{X}$ is zero. Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ be n independent and identically distributed observations following a multivariate normal distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ with mean equal to the zero vector and covariance matrix $\boldsymbol{\Sigma}$. The goal of this work is to estimate the precision matrix $\boldsymbol{\Omega} = (\omega_{ij})_{p\times p} = \boldsymbol{\Sigma}^{-1}$. Particular interest is to identify zero entries of ω_{ij}, since $\omega_{ij} = 0$ implies the conditional independence between X_i and X_j given all the other random variables. One way is that we obtain a sparse covariance matrix and then take its inverse. The

CONTACT Xinwei Deng xdeng@vt.edu

inverse, however, is often computationally intensive, especially in the high-dimensional cases. Moreover, the inverse of a sparse covariance matrix often would not result in sparse structure for the precision matrix. Therefore, it is desirable to obtain a sparse estimate directly to catch the underlying sparsity in the precision matrix.

The estimation of sparse precision matrix has attracted great attention in the literature. Meinshausen and Bühlmann [1] introduced a neighbourhood-based approach: it first estimates each column of the precision matrix by the scaled Lasso or Dantzig selector, and then adjusts the matrix estimator to be symmetric. Yuan and Lin [2] proposed the Graphical Lasso (Glasso) method, which gives a sparse and shrinkage estimator of $\boldsymbol{\Omega}$ by penalizing the negative log-likelihood as

$$\hat{\boldsymbol{\Omega}} = \arg\min_{\boldsymbol{\Omega}} \; -\log|\boldsymbol{\Omega}| + \mathrm{tr}[\boldsymbol{\Omega}\boldsymbol{S}] + \rho||\boldsymbol{\Omega}||_1,$$

where $\boldsymbol{S} = (1/n)\sum_{i=1}^{n} \boldsymbol{x}_i\boldsymbol{x}_i'$ is the sample covariance matrix, $\rho \geq 0$ is a tuning parameter, and $||\cdot||_1$ denotes the L_1 norm for the off-diagonal entries. Hence, the penalty term encourages some of the off-diagonal entries of the estimated $\boldsymbol{\Omega}$ to be exact zeroes. Different variations of the Glasso formulation have been later studied [3–8]. The corresponding theoretical properties of Glasso method are also developed [2,5,9,10]. In particular, the results from Raskutti et al. [5] and Rothman et al. [9] suggest that, although better than the sample covariance matrix, the Glasso estimate may not perform well when p is larger than the sample size n.

In addition, Fan, Fan and Lv [11] developed a factor model to estimate both covariance matrix and its inverse. They also studied the estimation in the asymptotic framework that both the dimension p and the sample size n go to infinity. Xue and Zou [12] introduced a rank-based approach for estimating high-dimensional nonparametric graphical models under a strong sparsity assumption that the true precision matrix has only a few nonzero entries. Wieringen and Peeters [13] studied the estimation of high-dimensional precision matrix based on the Ridge penalty. There are also a few work focusing on the inference for the precision matrix estimation. Drton and Perlman [14] proposed a new model selection strategy for Gaussian graphical models via hypotheses testings of the conditional independence between variables. Sun and Zhang [15] derived a residual-based estimator to construct confidence intervals for entries of the estimated precision matrix. Some recent Bayesian literature can also be found in the work of [16–20], among others.

Another set of commonly used methods is to consider the matrix decomposition for estimating sparse precision matrix. The modified Cholesky decomposition (MCD) approach was developed to estimate $\boldsymbol{\Omega}$ [21,22]. This method reduces the challenge of estimating a precision matrix into solving a sequence of regression problems, and provides an unconstrained and statistically interpretable parametrization of a precision matrix. Although the MCD approach is statistically meaningful, the resultant estimate depends on the order of the random variables $X_1, \ldots, X_p$. In many applications, the variables often do not have a natural order, that is, the variable order is not available or cannot be predetermined before the analysis. There are several Cholesky-based methods for estimating the precision matrix without specifying a natural order to the variables [23–25]. In this work, we propose an improved MCD approach to estimate the sparse precision matrix via addressing the variable order issue using the permutation idea in Zheng et al. [25]. By considering an ensemble estimate under multiple permutations of the variable orders, Zheng

et al. [25] introduced an order-averaged estimator for the large covariance matrix. However, they did not give the theoretical property of their estimator. Moreover, their estimator does not have sparsity, which is a very important and desired property for the matrix estimation in the high-dimensional cases. Hence, this paper improves the estimate of Zheng et al. [25] by encouraging the sparse structure in the precision matrix estimate. Specifically, we take average on the multiple estimates of the Cholesky factor matrix, and consequently construct the final estimate of the precision matrix. Since the averaged Cholesky factor matrix may not have sparse structure, we adopt the hard thresholding technique on the averaged Cholesky factor matrix to obtain the sparsity, thus leading to the sparse structure in the estimated precision matrix. The proposed method provides an accurate estimate and is able to capture the underlying sparse structure of the precision matrix. The sensitivity of the number of permutations of variable orders is studied in simulation. We also establish the consistency property of the proposed estimator regarding Frobenius norm under some appropriate conditions.

The remainder of the paper is divided into seven sections. In Section 2, we briefly review the MCD approach to estimate the precision matrix. In Section 3, we address the order issue of the MCD approach and propose an ensemble sparse estimate of $\boldsymbol{\Omega}$. The consistent property is established in Section 4. Simulation studies and illustrative examples of real data are presented in Sections 5 and 6, respectively. We conclude our work with some discussion in Section 7. The technical proof is given in Appendix.

2. Revisit of modified Cholesky decomposition

The key idea of the modified Cholesky decomposition approach is that the precision matrix $\boldsymbol{\Omega}$ can be decomposed using a unique lower triangular matrix $\boldsymbol{T}$ and a unique diagonal matrix $\boldsymbol{D}$ with positive diagonal entries [21] such that

$$\boldsymbol{\Omega} = \boldsymbol{T}'\boldsymbol{D}^{-1}\boldsymbol{T}.$$

The entries of $\boldsymbol{T}$ and the diagonal of $\boldsymbol{D}$ are unconstrained and interpretable as regression coefficients and corresponding variances when one variable X_j is regressed on its predecessors $X_1, \ldots, X_{j-1}$. Clearly, here an order for variables $X_1, \ldots, X_p$ is pre-specified. Specifically, consider $X_1 = \epsilon_1$, and for $j = 2, \ldots, p$, define

$$\begin{aligned} X_j &= \sum_{k=1}^{j-1} a_{jk} X_k + \epsilon_j \\ &= \boldsymbol{Z}_j' \boldsymbol{a}_j + \epsilon_j, \end{aligned} \tag{1}$$

where $\boldsymbol{Z}_j = (X_1, \ldots, X_{j-1})'$, and $\boldsymbol{a}_j = (a_{j1}, \ldots, a_{j,j-1})'$ is the corresponding vector of regression coefficients. The errors ϵ_j are assumed to be independent with zero mean and variance d_j^2. Denote $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_p)'$ and $\boldsymbol{D} = Cov(\boldsymbol{\epsilon}) = diag(d_1^2, \ldots, d_p^2)$. Then the p regression models in (1) can be expressed in the matrix form $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{\epsilon}$, where $\boldsymbol{A}$ is a lower triangular matrix with a_{jk} in the (j, k)th position, and 0 as its diagonal entries. Thus one can easily write $\boldsymbol{T}\boldsymbol{X} = \boldsymbol{\epsilon}$ with $\boldsymbol{T} = \boldsymbol{I} - \boldsymbol{A}$ to derive the expression of $\boldsymbol{\Omega} = \boldsymbol{T}'\boldsymbol{D}^{-1}\boldsymbol{T}$. The MCD approach therefore reduces the challenge of modelling a precision matrix into the task of modelling $(p - 1)$ regression problems.

Note that the MCD-based estimate is affected by the order of variables $X_1, \ldots, X_p$, since the regressions in (1) change when the order changes [23]. Hence, different orders of variables would result in different regressions, leading to different estimates of $\boldsymbol{T}$ and $\boldsymbol{D}$, and consequently different estimates of $\boldsymbol{\Omega}$. To demonstrate this clearly, we generate 20 observations from a 4-dimensional normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Omega}^{-1})$, where $\boldsymbol{\Omega}$ is a sparse matrix with 1 as its diagonal and $\boldsymbol{\Omega}_{13} = \boldsymbol{\Omega}_{31} = 0.5$. We consider two different variable orders $\pi_1 = (1, 2, 3, 4)$ and $\pi_2 = (1, 4, 3, 2)$, and obtain the corresponding estimates $\hat{\boldsymbol{\Omega}}_1$ and $\hat{\boldsymbol{\Omega}}_2$ based on the MCD (1) as follows, with the regression coefficients $\boldsymbol{a}_j$ estimated according to (3)

$$\hat{\boldsymbol{\Omega}}_1 = \begin{pmatrix} 1.80 & -0.13 & 0.75 & 0.06 \\ -0.13 & 1.94 & 0.24 & 0.07 \\ 0.75 & 0.24 & 0.83 & 0.08 \\ 0.06 & 0.07 & 0.08 & 1.41 \end{pmatrix} \quad \text{and}$$

$$\hat{\boldsymbol{\Omega}}_2 = \begin{pmatrix} 0.85 & 0.22 & 0.64 & 0.08 \\ 0.22 & 1.82 & -0.11 & 0.08 \\ 0.64 & -0.11 & 1.64 & 0.05 \\ 0.08 & 0.08 & 0.05 & 1.41 \end{pmatrix}.$$

Clearly, the estimates $\hat{\boldsymbol{\Omega}}_1$ and $\hat{\boldsymbol{\Omega}}_2$ are much different, due to the different variable orders used in the MCD. The variable orders significantly affect the Cholesky-based estimate. Hence, it is important to address this issue in the MCD-based approach. Wagaman and Levina [26] proposed an Isomap method to find the order of variables based on their correlations prior to applying banding techniques. Rajaratnam and Salzman [27] introduced a so-called 'best permutation algorithm' to recover the natural order of variables in autoregressive models for banded covariance matrix estimation, by minimizing the sum of the diagonals of $\boldsymbol{D}$ in the MCD approach. However, a natural variable order of $\boldsymbol{X}$ may not exist in practice, such as in the gene expression data or stock data. Moreover, the aforementioned methods for selecting a variable order are often designated for the banded matrix estimation. They are not suitable for the general matrix with no sparse structures. Therefore, in the next section, we propose a Cholesky-based ensemble sparse estimate of the precision matrix by addressing the order issue, with no assumption of sparse structure on the underlying matrix.

3. The proposed sparse estimator

To address the order issue and obtain an accurate estimate $\hat{\boldsymbol{\Omega}} = (\hat{\omega}_{ij})_{p\times p}$, we take advantage of permutations to gain the flexibility such that we can ensemble the multiple estimates from different orders.

Define a permutation mapping $\pi : \{1, \ldots, p\} \to \{1, \ldots, p\}$, which represents a rearrangement of the order of the variables,

$$(1, \ldots, p) \to (\pi(1), \pi(2), \ldots, \pi(p)). \tag{2}$$

Define the corresponding permutation matrix $\boldsymbol{P}_\pi$ of which the entries in the jth column are all 0 except taking 1 at position $\pi(j)$. Denote the n by p data matrix by $\mathbb{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)'$.

Therefore, the transformed data matrix is

$$\mathbb{X}_\pi = \mathbb{X}\boldsymbol{P}_\pi = (\boldsymbol{x}_\pi^{(1)}, \ldots, \boldsymbol{x}_\pi^{(p)}),$$

where $\boldsymbol{x}_\pi^{(j)}$ is the jth column of $\mathbb{X}_\pi$, $j = 1, 2, \ldots, p$. The Lasso technique [28] is employed for the shrinkage purpose and for the situation where p is close to n or even larger than n. The idea of Lasso-type estimator for the Cholesky factor has been used in some literatures [5,23,29,30]. Under a given permutation π, obtain

$$\hat{\boldsymbol{a}}_{\pi(j)} = \arg\min_{\boldsymbol{a}_{\pi(j)}} \|\boldsymbol{x}_\pi^{(\pi(j))} - \mathbb{W}_\pi^{(\pi(j))}\boldsymbol{a}_{\pi(j)}\|_2^2 + \lambda_{\pi(j)}|\boldsymbol{a}_{\pi(j)}|_1, \text{ for } \pi(j) \neq 1, \tag{3}$$

and

$$\hat{d}^2_{\pi(j)} = \begin{cases} \widehat{Var}(\boldsymbol{x}_\pi^{(1)}), & \pi(j) = 1, \\ \widehat{Var}(\boldsymbol{x}_\pi^{(\pi(j))} - \mathbb{W}_\pi^{(\pi(j))}\hat{\boldsymbol{a}}_{\pi(j)}), & \text{otherwise}, \end{cases} \tag{4}$$

where $\mathbb{W}_\pi^{(j)}$ represents the first (j-1) columns of $\mathbb{X}_\pi$, $\lambda_{\pi(j)} \geq 0$ is a tuning parameter, and $|\cdot|$ stands for the vector L_1 norm. Here $\widehat{Var}(\boldsymbol{\alpha}) = (\sum_{i=1}^p (\alpha_i - \bar{\alpha})^2)/(n-1)$, where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_p)'$ is a p-dimensional vector, and $\bar{\alpha} = \frac{1}{n}\sum_{i=1}^p \alpha_i$. The tuning parameter is chosen by the cross validation. Then we can model the lower triangular matrix $\hat{\boldsymbol{T}}_\pi$ with ones on its diagonal and $\hat{\boldsymbol{a}}'_{\pi(j)}$ as its $\pi(j)$th row. Meanwhile, the diagonal matrix $\hat{\boldsymbol{D}}_\pi$ has its $\pi(j)$th diagonal element equal to $\hat{d}^2_{\pi(j)}$. Correspondingly, $\hat{\boldsymbol{\Omega}}_\pi = \hat{\boldsymbol{T}}'_\pi \hat{\boldsymbol{D}}_\pi^{-1}\hat{\boldsymbol{T}}_\pi$ will be a sparse precision matrix estimate under π. Transforming back to the original order, we can estimate $\boldsymbol{\Omega}$ as

$$\begin{aligned} \hat{\boldsymbol{\Omega}} &= \boldsymbol{P}_\pi \hat{\boldsymbol{\Omega}}_\pi \boldsymbol{P}'_\pi \\ &= \boldsymbol{P}_\pi \hat{\boldsymbol{T}}'_\pi \hat{\boldsymbol{D}}_\pi^{-1} \hat{\boldsymbol{T}}_\pi \boldsymbol{P}'_\pi \\ &= (\boldsymbol{P}_\pi \hat{\boldsymbol{T}}'_\pi \boldsymbol{P}'_\pi)(\boldsymbol{P}_\pi \hat{\boldsymbol{D}}_\pi^{-1} \boldsymbol{P}'_\pi)(\boldsymbol{P}_\pi \hat{\boldsymbol{T}}_\pi \boldsymbol{P}'_\pi) \\ &\triangleq \hat{\boldsymbol{T}}' \hat{\boldsymbol{D}}^{-1} \hat{\boldsymbol{T}}. \end{aligned} \tag{5}$$

Note that $\hat{\boldsymbol{T}} = \boldsymbol{P}_\pi \hat{\boldsymbol{T}}_\pi \boldsymbol{P}'_\pi$ may no longer be a lower triangular matrix, but it still contains the sparse structure. Suppose we randomly generate M permutation mappings π_k, $k = 1, \ldots, M$. The word 'randomly' here means generating a permutation with all $p!$ possible permutations being equally probable. Accordingly, we obtain the corresponding estimates $\hat{\boldsymbol{\Omega}}$, $\hat{\boldsymbol{T}}$, and $\hat{\boldsymbol{D}}$ in (5), denoted as $\hat{\boldsymbol{\Omega}}_k$, $\hat{\boldsymbol{T}}_k$, and $\hat{\boldsymbol{D}}_k$ for the permutation π_k.

Based on the multiple estimates $\hat{\boldsymbol{T}}_k$'s and $\hat{\boldsymbol{D}}_k$'s, we consider the ensemble estimate of $\boldsymbol{\Omega}$ as follows

$$\tilde{\boldsymbol{\Omega}} = \tilde{\boldsymbol{T}}' \tilde{\boldsymbol{D}}^{-1} \tilde{\boldsymbol{T}} \quad \text{with} \quad \tilde{\boldsymbol{T}} = \frac{1}{M}\sum_{k=1}^M \hat{\boldsymbol{T}}_k, \quad \tilde{\boldsymbol{D}} = \frac{1}{M}\sum_{k=1}^M \hat{\boldsymbol{D}}_k. \tag{6}$$

The estimate in (6) is able to achieve good estimation accuracy since it reduces the variability in the estimates of $\tilde{\boldsymbol{T}}$ and $\tilde{\boldsymbol{D}}$. It is worth pointing out that we do not consider the

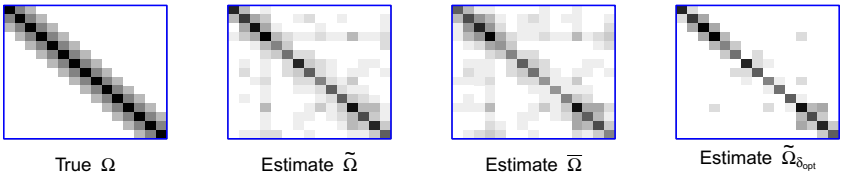

Figure 1. Heat maps for the true precision matrix $\boldsymbol{\Omega}$, the estimates $\tilde{\boldsymbol{\Omega}}$, $\bar{\boldsymbol{\Omega}}$ and the proposed estimate $\tilde{\boldsymbol{\Omega}}_{\delta_{opt}}$. Darker colour indicates higher density; lighter colour indicates lower density.

averaged estimate $\bar{\boldsymbol{\Omega}}$ based on the ensemble of $\hat{\boldsymbol{\Omega}}_k$ as in Zheng et al. [25]

$$\bar{\boldsymbol{\Omega}} = \frac{1}{M}\sum_{k=1}^{M}\hat{\boldsymbol{\Omega}}_k = \frac{1}{M}\sum_{k=1}^{M}\hat{\boldsymbol{T}}'_k\hat{\boldsymbol{D}}_k^{-1}\hat{\boldsymbol{T}}_k. \tag{7}$$

The reason is that the estimation error of $\hat{\boldsymbol{\Omega}}_k$ is already aggregated by the estimation error of $\hat{\boldsymbol{T}}_k$ and $\hat{\boldsymbol{D}}_k$. As shown in the simulations in Section 5, the estimate $\bar{\boldsymbol{\Omega}}$ does not give good performance on the estimation.

Although the method (6) is able to produce an accurate estimate $\tilde{\boldsymbol{\Omega}}$, it fails to capture any sparse structure of the true precision matrix, since $\tilde{\boldsymbol{T}}$ in (6) does not contain the sparsity. To illustrate this point, we generate 50 observations from normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Omega}^{-1})$, where $\boldsymbol{\Omega}$ is a 15×15 banded structure with main diagonal 1, the first sub-diagonal 0.5 and the second sub-diagonal 0.3. The first three panels of Figure 1 display the heat maps for the true precision matrix $\boldsymbol{\Omega}$, the estimates $\tilde{\boldsymbol{\Omega}}$ in (6) and $\bar{\boldsymbol{\Omega}}$ in (7). Clearly, there are many non-zeroes in the off-diagonal positions of estimates $\tilde{\boldsymbol{\Omega}}$ and $\bar{\boldsymbol{\Omega}}$.

Therefore, to encourage the sparse structure in the estimate of $\boldsymbol{T}$, we impose a hard thresholding on each entry of $\tilde{\boldsymbol{T}}$ in (6), which results in the sparsity of $\tilde{\boldsymbol{T}}$, hence leading to a sparse estimate of $\boldsymbol{\Omega}$. The resultant estimate of $\boldsymbol{\Omega}$ not only enjoys the sparse structure, but also requires no information on the variable order in the MCD before analysis.

The hard thresholding procedure is described as follows. let $\tilde{\boldsymbol{T}} = (\tilde{t}_{ij})_{p\times p}$ be the ensemble estimate obtained from method (6) and a hard thresholding is denoted by δ. Then $\tilde{\boldsymbol{T}}_\delta = (\tilde{t}_{ij}^{(\delta)})_{p\times p}$ is defined as

$$\tilde{t}_{ij}^{(\delta)} = \begin{cases} \tilde{t}_{ij}, & \text{if } |\tilde{t}_{ij}| > \delta, \\ 0, & \text{if } |\tilde{t}_{ij}| \le \delta, \end{cases} \tag{8}$$

then the sparse estimate $\tilde{\boldsymbol{\Omega}}_\delta = \tilde{\boldsymbol{T}}'_\delta\tilde{\boldsymbol{D}}^{-1}\tilde{\boldsymbol{T}}_\delta$.

A large value of thresholding will improve the performance of capturing sparse structure, but reduce the estimation accuracy. To choose an appropriate value for hard thresholding, we suggest to use the Bayesian information criterion (BIC) which is to balance the tradeoff between the fitting of the likelihood function and the sparsity of the estimate. Specifically, for a given hard thresholding δ_l, $l = 1, \ldots, H$, the corresponding BIC(δ_l) [2] is computed by

$$\text{BIC}(\delta_l) = -\log|\tilde{\boldsymbol{\Omega}}_{\delta_l}| + \text{tr}[\tilde{\boldsymbol{\Omega}}_{\delta_l}\boldsymbol{S}] + \frac{\log n}{n}\sum_{i\le j}\tilde{e}_{ij}(l), \tag{9}$$

where $\tilde{\boldsymbol{\Omega}}_{\delta_l} = (\tilde{\omega}_{ij}^{(\delta_l)})_{p\times p} = \tilde{\boldsymbol{T}}_{\delta_l}'\tilde{\boldsymbol{D}}^{-1}\tilde{\boldsymbol{T}}_{\delta_l}$ using the hard thresholding δ_l. $\tilde{e}_{ij}(l) = 0$ if $\tilde{\omega}_{ij}^{(\delta_l)} = 0$, and $\tilde{e}_{ij}(l) = 1$ otherwise. The optimal hard thresholding δ_{opt} is chosen such that its corresponding BIC value is smallest. Then the proposed sparse precision estimate is

$$\tilde{\boldsymbol{\Omega}}_{\delta_{opt}} = \tilde{\boldsymbol{T}}_{\delta_{opt}}'\tilde{\boldsymbol{D}}^{-1}\tilde{\boldsymbol{T}}_{\delta_{opt}}. \tag{10}$$

Clearly, the method (6) can be viewed as a special case of the proposed estimate with hard thresholding $\delta = 0$. The fourth panel in Figure 1 shows the heat map of the proposed estimate $\tilde{\boldsymbol{\Omega}}_{\delta_{opt}}$ in (10). It has much less off-diagonal non-zeroes compared with $\tilde{\boldsymbol{\Omega}}$ and $\bar{\boldsymbol{\Omega}}$.

The algorithm of proposed estimate of sparse precision matrix $\boldsymbol{\Omega}$ based on the MCD is summarized as follows:

Algorithm 1: ***Step 1****: Input centred data.*

Step 2*: Randomly generate M permutation mappings* π_k *as in* (2), $k = 1, 2, \ldots, M$.

Step 3*: Under each permutation* π_k*, construct* $\hat{\boldsymbol{T}}_{\pi_k}$ *from the estimates of regression coefficients in* (3)*. Obtain* $\hat{\boldsymbol{D}}_{\pi_k}$ *from the corresponding residual variances in* (4).

Step 4*: Transform to the original order:* $\hat{\boldsymbol{T}}_k = \boldsymbol{P}_{\pi_k}\hat{\boldsymbol{T}}_{\pi_k}\boldsymbol{P}_{\pi_k}'$ *and* $\hat{\boldsymbol{D}}_k = \boldsymbol{P}_{\pi_k}\hat{\boldsymbol{D}}_{\pi_k}\boldsymbol{P}_{\pi_k}'$.

Step 5*:* $\tilde{\boldsymbol{T}} = \frac{1}{M}\sum_{k=1}^{M}\hat{\boldsymbol{T}}_k$, $\tilde{\boldsymbol{D}} = \frac{1}{M}\sum_{k=1}^{M}\hat{\boldsymbol{D}}_k$ *as in* (6).

Step 6*: Obtain* $\tilde{\boldsymbol{T}}_{\delta_{opt}}$ *from* (8) *by applying* δ_{opt} *to* $\tilde{\boldsymbol{T}}$*, where* δ_{opt} *is selected by* (9).

Step 7*:* $\tilde{\boldsymbol{\Omega}} = \tilde{\boldsymbol{T}}_{\delta_{opt}}'\tilde{\boldsymbol{D}}^{-1}\tilde{\boldsymbol{T}}_{\delta_{opt}}$ *as in* (10).

As seen in Algorithm 1, the proposed method attempts to balance between the accuracy and sparsity of the estimate for $\boldsymbol{\Omega}$. Meanwhile, we would like to point out that Algorithm 1 is also very flexible with respect to the objective in practice. If the practical objective does not focus on the sparse structure of the precision matrix, one can set the hard thresholding $\delta = 0$ for the estimation of $\boldsymbol{\Omega}$. As shown in Section 5, such an estimator has good performance in certain setting of covariance structure.

Note that the proposed method needs to choose the number of permutations M. To choose an appropriate number of permutations M for efficient computation, we have tried $M = 10$, 30, 50, 80, 100, 120 and 150 as the number of randomly selected permutations from all the possible permutations. The performance results are quite comparable when M is larger than 30 or 50. In this paper, we choose $M = 100$ for the proposed method. The sensitivity of the order of variables is also examined in the simulation.

4. Consistency property

In this section, the asymptotic property regarding the consistency of the proposed estimator is established. We start by introducing some notation. Let $\boldsymbol{\Omega}_0 = (\omega_{ij}^0)_{p\times p} = \boldsymbol{T}_0'\boldsymbol{D}_0^{-1}\boldsymbol{T}_0$ be the true precision matrix and its MCD. The singular values of matrix $\boldsymbol{A}$ are denoted by $sv_1(A) \geq sv_2(A) \geq \ldots \geq sv_p(A)$, which are the squared root of the eigenvalues of matrix $\boldsymbol{AA}'$. In order to theoretically construct the asymptotic property for the estimator $\tilde{\boldsymbol{\Omega}}_\delta = \tilde{\boldsymbol{T}}_\delta'\tilde{\boldsymbol{D}}^{-1}\tilde{\boldsymbol{T}}_\delta$, we assume that there exists a constant h such that

$$0 < 1/h < sv_p(\boldsymbol{\Omega}_0) \leq sv_1(\boldsymbol{\Omega}_0) < h < \infty. \tag{11}$$

The similar assumption is also made in [5,10,31]. It guarantees the positive definiteness property of $\boldsymbol{\Omega}_0$. Now we present the following theory. The proof is given in the Appendix.

Theorem 4.1: *Assume data are from Normal distribution* $N(\mathbf{0}, \mathbf{\Omega}_0^{-1})$. *Under (11), assume that* $p\log(p) = o(n)$, *and the tuning parameters* $\lambda_{\pi(j)}$ *in (3) satisfy* $\sum_{j=1}^{p} \lambda_{\pi(j)} = O(\log(p)/n)$. *The hard thresholding parameter* δ *satisfies* $\delta = O(\sqrt{((\log(p))/(nM))})$, *and* $M = O(p)$. *Then we have* $\|\tilde{\mathbf{\Omega}}_\delta - \mathbf{\Omega}_0\|_F \xrightarrow{P} 0$.

Theorem 4.1 establishes the consistency property of the estimator $\tilde{\mathbf{\Omega}}_\delta$ regarding Frobenius norm under some appropriate conditions. Here the assumption $M = O(p)$ is to allow the number of permutations of variable orders M to vary with the number of variables p. A larger value of M is recommended for the proposed method as the number of variables p increases. A numeric study to investigate the impact of the choice of M on the performance of the proposed method is conducted in the simulation section.

5. Simulation

In this section, we present a simulation study which evaluates the performance of the proposed method in comparison with several existing approaches. Two versions of the proposed method are considered, denoted by M1 and M2, respectively. *The proposed method M1* represents the estimate in (6) with hard thresholding $\delta = 0$. *The proposed method M2* stands for the estimate in (10) with hard thresholding chosen by the BIC criterion as in (9). Among the comparison methods, The first one is the MCD method for estimating $\mathbf{\Omega}$ with the order chosen by BIC criterion [32], denoted as BIC. The second compared approach is the Best Permutation Algorithm [27], denoted by BPA. It selects the order of variables such that $||\boldsymbol{D}||_F^2$ is minimized, where $||\cdot||_F$ denotes the Frobenius norm, and $\boldsymbol{D}$ is the diagonal matrix in the MCD approach. The third method is the estimate $\bar{\mathbf{\Omega}}$ in (7), denoted by AVE. The last method for comparison is the Graphical Lasso [1–3], denoted as Glasso. In all the Cholesky-based approaches, the Cholesky factor matrix $\boldsymbol{T}$ is constructed according to (3), with the tuning parameter chosen by the cross validation.

Denote by $\hat{\mathbf{\Omega}} = (\hat{\omega}_{ij})_{p\times p}$ an estimate for the covariance matrix $\mathbf{\Omega} = (\omega_{ij})_{p\times p}$. To measure the accuracy of a precision matrix estimate, we consider the Kullback-Leibler loss Δ_1, the entropy loss Δ_2 and the quadratic loss Δ_3 (up to some scale) as follows,

$$\Delta_1 = \frac{1}{p}\left(\mathrm{tr}[\mathbf{\Omega}^{-1}\hat{\mathbf{\Omega}}] - \log|\mathbf{\Omega}^{-1}\hat{\mathbf{\Omega}}| - p\right),$$

$$\Delta_2 = \frac{1}{p}\left(\mathrm{tr}[\hat{\mathbf{\Omega}}^{-1}\mathbf{\Omega}] - \log|\hat{\mathbf{\Omega}}^{-1}\mathbf{\Omega}| - p\right),$$

$$\Delta_3 = \frac{1}{p}\left[\mathrm{tr}(\mathbf{\Omega}^{-1}\hat{\mathbf{\Omega}} - \boldsymbol{I})\right]^2.$$

We also use the mean absolute error and mean squared error given by

$$\mathrm{MAE} = \frac{1}{p}\sum_{i=1}^{p}\sum_{j=1}^{p}|\hat{\omega}_{ij} - \omega_{ij}| \quad \text{and} \quad \mathrm{MSE} = \frac{1}{p}\sum_{i=1}^{p}\sum_{j=1}^{p}(\hat{\omega}_{ij} - \omega_{ij})^2.$$

In addition, to gauge the performance of the estimates in capturing the sparse structure, the false selection loss (FSL) are used, which is the summation of false positive (FP) and false

negative (FN). We say a FP occurs if a nonzero element in the true matrix is incorrectly estimated as a zero. Similarly, a FN occurs if a zero element in the true matrix is incorrectly identified as a nonzero. The FSL is computed in percentage as (FP + FN) / p^2. For each loss function above, we report the averages of the performance measures over 50 replicates.

We consider the following six precision matrix (i.e. precision matrix) structures.

Model 1. $\mathbf{\Omega}_1 = \text{MA}(0.5, 0.3)$. The main diagonal elements are 1 with first sub-diagonal elements 0.5 and second sub-diagonal elements 0.3.

Model 2. $\mathbf{\Omega}_2$ is generated by randomly permuting rows and corresponding columns of $\mathbf{\Omega}_1$.

Model 3. $\mathbf{\Omega}_3 = \begin{pmatrix} \text{CS}(0.5) & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{pmatrix}$, where CS(0.5) represents a 10×10 compound structure matrix with diagonal elements 1 and others 0.5. $\mathbf{0}$ indicates a matrix with all elements 0.

Model 4. $\mathbf{\Omega}_4 = \text{AR}(0.5)$. The conditional covariance between any two random variables X_i and X_j is fixed to be $0.5^{|i-j|}$, $1 \le i, j \le p$.

Model 5. $\mathbf{\Omega}_5^{-1} = \text{diag}(p, p-1, p-2, \ldots, 1)$.

Model 6. $\mathbf{\Omega}_6 = \boldsymbol{B}'\boldsymbol{H}^{-1}\boldsymbol{B}$, where $\boldsymbol{H} = 0.01 \times \boldsymbol{I}$, and $\boldsymbol{B} = (-\phi_{t,s})$ with $\phi_{t,t} = 1$, $\phi_{t+1,t} = 0.8$, and $\phi_{t,s} = 0$ otherwise.

Model 1 is a sparse banded structure. *Model* 2 permutates the rows and corresponding columns of *Model* 1 randomly. *Model* 3 is a block compound structure on the upper left corner. It is becoming more and more sparse as the dimension p increases. *Model* 4 is an autoregressive structure that has homogeneous variances and correlations declining with distance. This model is more dense than the other models. The structures of *Model* 5 and *Model* 6 are also used in Huang et al. (2006) [29]. For each model, we generate normally distributed data with three settings of sample sizes and variable sizes: (1) $n = 50$, $p = 30$; (2) $n = 50$, $p = 50$ and (3) $n = 50$, $p = 100$. Table 1 to Table 3 report the loss measures of the estimates averaged over 50 replicates and their corresponding standard errors (in parenthesis) for different approaches. For each model, the lowest averages regarding each measure are shown in bold.

Table 1 reports the averages and corresponding standard errors (in parenthesis) of different loss measures obtained from each method when $p = 30$. From the results it can be seen that, by addressing the order issue, the proposed methods M1 and M2 considerably outperform other approaches with respect to all the loss measures. Overall, the M1 performs the best under Δ_1, Δ_3 and MSE criteria, followed by M2. The M2 produces the minimum MAE in all the seven models. It also significantly dominates all the other approaches in terms of FSL except *Model* 3, where the M2 is the second best and inferior to the Glasso. Nevertheless, the M2 substantially outperforms the Glasso in *Model* 3 regarding all the other loss measures. Additionally, although the AVE gives the best performance for the loss function Δ_2, the M1 is much comparable. Particularly, from the perspective of models, the M2 generally gives the superior performance to the other methods in the sparse *Model* 5, and also shows advantage in *Model* 6. Moreover, from the perspective of variation, the proposed methods M1 and M2 result in a much smaller variability of the estimates for all the models in terms of Δ_1, Δ_3 and MSE. The AVE has comparable standard errors regarding Δ_2, and the Glasso gives the smallest standard errors under MAE.

Compared with the proposed methods, the MCD approach based on the BIC order selection (i.e. BIC) does not perform as well as M1 and M2, which implies that using a

Table 1. The averages and standard errors (in parenthesis) of estimates for $p = 30$.

		Δ_1	Δ_2	Δ_3	MAE	MSE	FSL (%)
Model 1	M1	**0.177 (0.004)**	0.168 (0.003)	**0.898 (0.099)**	1.667 (0.015)	**0.340 (0.005)**	83.293 (0.066)
	M2	0.278 (0.006)	0.246 (0.004)	2.932 (0.223)	**1.397 (0.013)**	0.460 (0.008)	**7.640 (0.189)**
	BIC	0.390 (0.016)	0.244 (0.005)	9.411 (0.705)	3.692 (0.293)	1.784 (0.324)	71.489 (0.814)
	BPA	0.274 (0.016)	0.198 (0.004)	5.201 (0.808)	2.347 (0.255)	0.880 (0.350)	54.151 (1.203)
	AVE	0.256 (0.009)	**0.158 (0.003)**	6.483 (0.475)	2.289 (0.103)	0.527 (0.053)	83.764 (0.031)
	Glasso	0.323 (0.008)	0.845 (0.035)	2.031 (0.132)	2.086 (0.011)	0.948 (0.017)	12.862 (0.563)
Model 2	M1	**0.175 (0.003)**	0.167 (0.002)	**0.859 (0.069)**	1.653 (0.010)	**0.335 (0.004)**	83.347 (0.044)
	M2	0.283 (0.005)	0.250 (0.004)	3.004 (0.171)	**1.391 (0.013)**	0.463 (0.008)	**7.502 (0.175)**
	BIC	0.371 (0.011)	0.239 (0.003)	8.443 (0.539)	3.539 (0.248)	1.654 (0.301)	71.320 (0.888)
	BPA	0.271 (0.007)	0.201 (0.004)	4.723 (0.272)	2.319 (0.097)	0.645 (0.056)	55.391 (1.225)
	AVE	0.248 (0.005)	**0.157 (0.002)**	6.072 (0.300)	2.235 (0.068)	0.490 (0.030)	83.804 (0.031)
	Glasso	0.329 (0.006)	0.862 (0.026)	2.039 (0.098)	2.097 (0.007)	0.967 (0.011)	12.071 (0.247)
Model 3	M1	0.086 (0.002)	0.161 (0.004)	**0.572 (0.056)**	2.024 (0.015)	0.794 (0.009)	84.920 (0.125)
	M2	**0.081 (0.001)**	0.215 (0.003)	0.604 (0.056)	**1.766 (0.007)**	0.848 (0.005)	10.209 (0.070)
	BIC	0.230 (0.019)	0.156 (0.005)	3.939 (0.653)	4.326 (0.533)	3.099 (0.933)	56.898 (2.788)
	BPA	0.144 (0.006)	0.235 (0.005)	1.388 (0.147)	2.492 (0.076)	1.074 (0.035)	35.267 (1.878)
	AVE	0.111 (0.005)	**0.116 (0.005)**	1.738 (0.164)	2.323 (0.097)	**0.752 (0.054)**	86.013 (0.072)
	Glasso	0.099 (0.002)	0.331 (0.005)	1.904 (0.076)	1.869 (0.004)	0.901 (0.003)	**9.667 (0.109)**
Model 4	M1	**0.136 (0.002)**	0.141 (0.002)	**0.665 (0.063)**	1.886 (0.011)	**0.380 (0.004)**	46.529 (0.068)
	M2	0.171 (0.003)	0.189 (0.004)	1.253 (0.094)	**1.764 (0.011)**	0.478 (0.007)	**44.382 (0.219)**
	BIC	0.301 (0.014)	0.202 (0.004)	5.925 (0.609)	3.341 (0.197)	1.409 (0.184)	45.649 (0.309)
	BPA	0.210 (0.006)	0.174 (0.003)	2.838 (0.235)	2.294 (0.056)	0.582 (0.029)	44.818 (0.479)
	AVE	0.184 (0.006)	**0.133 (0.002)**	3.683 (0.267)	2.149 (0.048)	0.429 (0.021)	46.671 (0.044)
	Glasso	0.203 (0.003)	0.467 (0.012)	1.707 (0.072)	2.279 (0.007)	0.801 (0.008)	45.173 (0.168)
Model 5	M1	0.047 (0.002)	0.038 (0.001)	0.546 (0.066)	0.070 (0.003)	0.007 (0.001)	70.636 (0.755)
	M2	**0.033 (0.001)**	**0.027 (0.001)**	**0.481 (0.057)**	**0.436 (0.072)**	**0.006 (0.001)**	**3.920 (0.651)**
	BIC	0.093 (0.009)	0.061 (0.003)	1.337 (0.227)	0.097 (0.010)	0.015 (0.003)	27.422 (2.092)
	BPA	0.082 (0.004)	0.057 (0.002)	0.992 (0.116)	0.129 (0.006)	0.014 (0.002)	21.680 (1.690)
	AVE	0.066 (0.005)	0.047 (0.002)	1.186 (0.164)	0.096 (0.005)	0.009 (0.001)	79.827 (0.838)
	Glasso	0.095 (0.002)	0.183 (0.005)	1.645 (0.079)	0.070 (0.000)	0.027 (0.000)	8.240 (0.344)
Model 6	M1	**0.129 (0.002)**	0.157 (0.003)	**0.357 (0.048)**	1.793 (0.012)	0.619 (0.011)	89.578 (0.046)
	M2	0.230 (0.004)	0.188 (0.003)	0.404 (0.037)	**1.366 (0.014)**	**0.579 (0.011)**	**10.071 (0.294)**
	BIC	0.261 (0.015)	0.163 (0.004)	5.583 (0.611)	3.572 (0.360)	2.798 (0.674)	58.284 (1.745)
	BPA	0.171 (0.009)	0.110 (0.004)	2.663 (0.264)	2.167 (0.171)	0.974 (0.162)	42.089 (1.527)
	AVE	0.162 (0.006)	**0.101 (0.002)**	3.744 (0.284)	2.203 (0.086)	0.698 (0.054)	89.920 (0.038)
	Glasso	0.138 (0.002)	0.189 (0.005)	0.837 (0.087)	1.738 (0.016)	0.708 (0.025)	29.289 (0.555)

single variable order in the MCD approach may be not helpful to improve the estimation accuracy, while the multiple orders would lead to a more accurate estimate. Also, the inferior performance of the AVE to the proposed methods implies that the way of assembling the available estimates obtained from multiple orders is important, i.e. the method (6) with the ensemble estimates $\tilde{\boldsymbol{T}}$ and $\tilde{\boldsymbol{D}}$ performs better than the method (7) of the ensemble estimate $\bar{\boldsymbol{\Omega}}$.

Table 2 and Table 3 present the comparison results regarding the loss measures Δ_1, Δ_2, Δ_3, MAE, MSE and FSL for $p = 50$ and $p = 100$, respectively. Tables show the similar conclusions as $p = 30$. The proposed methods generally give superior performances to the other approaches. As the number of variables p increases, the proposed methods work even more promising as expected. For example, the M2 performs better and better for *Model* 3 as the number of variables p increases, since this model is becoming more and more sparse. Compared with AVE, the proposed methods result in much smaller losses and standard errors in terms of Δ_2 when $p = 100$. In addition, the M1 performs the best in the dense *Model* 4 in all the settings of p, since the M1 is able to give an accurate estimate when the true model is not sparse.

Table 2. The averages and standard errors (in parenthesis) of estimates for $p = 50$.

		Δ_1	Δ_2	Δ_3	MAE	MSE	FSL (%)
Model 1	M1	**0.218 (0.003)**	**0.198 (0.002)**	**2.780 (0.148)**	1.894 (0.010)	**0.392 (0.004)**	89.043 (0.064)
	M2	0.316 (0.004)	0.274 (0.004)	6.779 (0.259)	**1.483 (0.010)**	0.509 (0.006)	**5.115 (0.097)**
	BIC	0.986 (0.101)	0.332 (0.006)	54.398 (6.024)	11.774 (1.775)	11.431 (2.665)	76.197 (0.827)
	BPA	0.383 (0.012)	0.249 (0.004)	15.434 (0.987)	3.172 (0.169)	1.155 (0.190)	53.430 (0.970)
	AVE	0.436 (0.017)	0.209 (0.003)	28.162 (1.925)	4.268 (0.272)	1.556 (0.246)	90.024 (0.024)
	Glasso	0.363 (0.003)	1.023 (0.016)	4.090 (0.126)	2.170 (0.004)	1.037 (0.005)	7.232 (0.055)
Model 2	M1	**0.217 (0.002)**	**0.202 (0.002)**	**2.502 (0.135)**	1.882 (0.009)	**0.404 (0.004)**	88.944 (0.052)
	M2	0.321 (0.003)	0.285 (0.003)	6.527 (0.261)	**1.516 (0.009)**	0.529 (0.005)	**5.160 (0.087)**
	BIC	0.814 (0.112)	0.321 (0.005)	32.928 (8.454)	9.104 (2.227)	6.361 (3.415)	74.122 (0.943)
	BPA	0.352 (0.012)	0.247 (0.004)	12.129 (0.903)	2.850 (0.173)	0.949 (0.165)	49.699 (1.079)
	AVE	0.407 (0.018)	0.203 (0.003)	24.151 (1.766)	4.203 (0.379)	1.685 (0.400)	89.984 (0.029)
	Glasso	0.360 (0.002)	0.998 (0.011)	3.896 (0.093)	2.164 (0.003)	1.031 (0.004)	7.181 (0.060)
Model 3	M1	0.075 (0.001)	0.134 (0.002)	1.146 (0.079)	1.538 (0.013)	0.563 (0.005)	88.010 (0.299)
	M2	**0.065 (0.001)**	0.142 (0.002)	**1.078 (0.075)**	**1.164 (0.005)**	**0.551 (0.004)**	**3.888 (0.032)**
	BIC	0.709 (0.284)	0.160 (0.005)	13.151 (3.098)	12.454 (5.710)	5.804 (1.822)	45.938 (2.656)
	BPA	0.141 (0.006)	0.174 (0.003)	2.889 (0.253)	2.049 (0.063)	0.856 (0.031)	24.037 (1.070)
	AVE	0.174 (0.024)	**0.133 (0.003)**	7.471 (1.964)	3.515 (0.563)	2.121 (0.808)	93.526 (0.102)
	Glasso	0.083 (0.001)	0.230 (0.003)	3.490 (0.090)	1.240 (0.003)	0.577 (0.002)	3.904 (0.053)
Model 4	M1	**0.161 (0.002)**	**0.170 (0.002)**	**1.698 (0.120)**	2.143 (0.009)	**0.448 (0.004)**	64.654 (0.075)
	M2	0.193 (0.002)	0.216 (0.003)	2.745 (0.157)	**1.888 (0.008)**	0.530 (0.005)	**29.626 (0.079)**
	BIC	0.651 (0.104)	0.256 (0.005)	29.444 (5.522)	8.095 (1.927)	5.766 (1.464)	55.331 (0.592)
	BPA	0.285 (0.017)	0.216 (0.003)	8.544 (1.462)	3.206 (0.387)	1.621 (0.849)	43.960 (0.553)
	AVE	0.283 (0.012)	0.173 (0.002)	13.04 (1.058)	3.468 (0.212)	0.994 (0.157)	65.379 (0.033)
	Glasso	0.233 (0.002)	0.581 (0.010)	3.520 (0.093)	2.400 (0.004)	0.882 (0.005)	30.557 (0.070)
Model 5	M1	0.047 (0.002)	0.038 (0.001)	1.035 (0.077)	0.049 (0.001)	**0.003 (0.000)**	46.182 (0.746)
	M2	**0.034 (0.001)**	**0.027 (0.001)**	**0.922 (0.071)**	**0.021 (0.001)**	**0.003 (0.000)**	**0.224 (0.038)**
	BIC	0.280 (0.076)	0.079 (0.004)	6.519 (1.542)	0.258 (0.094)	0.022 (0.010)	31.037 (2.411)
	BPA	0.107 (0.009)	0.064 (0.003)	2.622 (0.389)	0.126 (0.016)	0.016 (0.007)	19.256 (1.214)
	AVE	0.087 (0.006)	0.055 (0.002)	3.110 (0.255)	0.096 (0.008)	0.006 (0.001)	73.165 (1.283)
	Glasso	0.109 (0.001)	0.239 (0.005)	2.736 (0.119)	0.053 (0.000)	0.020 (0.000)	7.973 (0.338)
Model 6	M1	**0.148 (0.002)**	0.163 (0.002)	**0.253 (0.041)**	1.882 (0.008)	0.626 (0.007)	92.797 (0.044)
	M2	0.267 (0.004)	0.196 (0.002)	0.691 (0.081)	**1.374 (0.009)**	**0.587 (0.007)**	**6.059 (0.152)**
	BIC	0.552 (0.163)	0.202 (0.004)	17.680 (1.523)	9.558 (4.628)	5.275 (1.026)	57.403 (1.252)
	BPA	0.223 (0.008)	0.131 (0.003)	6.909 (0.478)	2.561 (0.116)	1.069 (0.084)	39.218 (1.120)
	AVE	0.243 (0.012)	**0.123 (0.002)**	12.312 (0.978)	3.611 (0.373)	1.964 (0.563)	93.768 (0.029)
	Glasso	0.196 (0.003)	0.306 (0.009)	2.832 (0.181)	2.085 (0.012)	1.094 (0.025)	21.902 (0.360)

Moreover, to investigate the impact of the choice of M on the performance of the proposed methods, we compute six loss measures for different values of $M = 10$, 30, 50, 80, 100, 120 and 150. Figure 2 and 3 display the corresponding results obtained from the proposed methods M1 and M2 for *Model* 2. The solid line, dashed line and dotted line represent three situations where $p = 30$, 50 and 100, respectively. To clearly distinguish three different lines for the loss measure FSL, here we present NFSL $=$ FP $+$ FN rather than its percentage form. Overall, it is clear to see that almost all of the lines are significantly decreasing in the range of $M = (10, 30)$, and are stable over other values of M. The dotted lines ($p = 100$) are slightly decreasing in the range of $M = (30, 50)$ for some criteria since a relative large value of M is needed for large p.

In a brief summary, the numerical results show that the proposed methods give a superior performance over some other conventional approaches. The M2 performs well when the underlying precision matrix is sparse. It is able to catch the sparse structure of $\mathbf{\Omega}$. In comparison, although the M1 does not provide a sparse estimate, it gives an accurate estimate with respect to Δ_1 – Δ_3. Hence, the M1 method is suitable when the true $\mathbf{\Omega}$ is not sparse.

Table 3. The averages and standard errors (in parenthesis) of estimates for $p = 100$.

		Δ_1	Δ_2	Δ_3	MAE	MSE	FSL (%)
Model 1	M1	**0.275 (0.002)**	**0.248 (0.002)**	**9.296 (0.354)**	2.180 (0.009)	**0.484 (0.003)**	92.118 (0.077)
	M2	0.360 (0.003)	0.319 (0.002)	17.52 (0.474)	**1.628 (0.005)**	0.588 (0.003)	**2.991 (0.034)**
	BIC	5.549 (0.618)	0.473 (0.007)	901.919 (148.717)	67.248 (7.801)	201.993 (59.170)	73.237 (0.668)
	BPA	1.043 (0.200)	0.343 (0.004)	349.150 (162.716)	10.927 (2.649)	100.857 (57.985)	47.570 (0.994)
	AVE	1.587 (0.070)	0.372 (0.006)	499.315 (37.701)	15.332 (0.699)	18.726 (2.345)	94.940 (0.009)
	Glasso	0.392 (0.002)	1.171 (0.012)	9.690 (0.205)	2.222 (0.002)	1.086 (0.003)	3.750 (0.016)
Model 2	M1	**0.271 (0.002)**	**0.250 (0.002)**	**8.470 (0.273)**	2.180 (0.009)	**0.487 (0.003)**	92.128 (0.075)
	M2	0.354 (0.002)	0.320 (0.002)	16.103 (0.372)	**1.630 (0.005)**	0.588 (0.004)	**2.998 (0.029)**
	BIC	5.238 (0.610)	0.482 (0.007)	730.631 (82.704)	58.675 (6.936)	136.441 (35.581)	74.513 (0.642)
	BPA	0.795 (0.107)	0.340 (0.005)	153.276 (51.777)	7.778 (1.569)	31.015 (16.681)	46.890 (1.020)
	AVE	1.668 (0.073)	0.374 (0.007)	536.125 (39.379)	16.182 (0.702)	22.304 (2.673)	94.924 (0.010)
	Glasso	0.389 (0.002)	1.157 (0.012)	9.672 (0.200)	2.219 (0.002)	1.083 (0.003)	3.732 (0.016)
Model 3	M1	0.071 (0.001)	0.093 (0.001)	3.194 (0.130)	1.274 (0.012)	0.3778 (0.004)	85.552 (0.443)
	M2	**0.058 (0.001)**	**0.089 (0.001)**	**2.938 (0.122)**	**0.739 (0.005)**	**0.344 (0.003)**	**1.162 (0.019)**
	BIC	1.559 (0.414)	0.162 (0.004)	158.478 (36.040)	24.432 (7.157)	83.644 (26.354)	40.689 (1.361)
	BPA	0.266 (0.027)	0.150 (0.003)	21.652 (4.107)	3.545 (0.429)	4.181 (1.567)	22.407 (1.081)
	AVE	0.976 (0.087)	0.201 (0.006)	223.731 (33.502)	16.309 (1.326)	35.368 (6.228)	97.729 (0.036)
	Glasso	0.078 (0.001)	0.166 (0.002)	8.309 (0.151)	0.781 (0.002)	0.352 (0.002)	1.212 (0.021)
Model 4	M1	**0.195 (0.002)**	**0.205 (0.001)**	**5.732 (0.270)**	2.415 (0.009)	**0.517 (0.003)**	77.789 (0.155)
	M2	0.224 (0.002)	0.247 (0.002)	8.074 (0.330)	**2.006 (0.005)**	0.584 (0.003)	**16.170 (0.027)**
	BIC	2.949 (0.465)	0.352 (0.007)	303.160 (61.744)	37.093 (6.194)	67.431 (20.123)	57.449 (0.865)
	BPA	0.449 (0.029)	0.276 (0.004)	38.051 (5.745)	4.646 (0.380)	3.652 (1.510)	38.676 (0.590)
	AVE	1.105 (0.058)	0.301 (0.005)	261.304 (23.585)	12.519 (0.617)	12.556 (1.557)	81.760 (0.013)
	Glasso	0.259 (0.002)	0.690 (0.007)	8.239 (0.148)	2.489 (0.003)	0.943 (0.003)	16.559 (0.018)
Model 5	M1	0.054 (0.002)	0.041 (0.001)	2.952 (0.183)	0.032 (0.001)	**0.002 (0.000)**	22.498 (0.454)
	M2	**0.040 (0.001)**	**0.031 (0.001)**	**2.655 (0.163)**	**0.014 (0.001)**	**0.002 (0.000)**	**1.216 (0.481)**
	BIC	2.065 (0.567)	0.121 (0.006)	211.015 (55.567)	0.983 (0.298)	1.077 (0.099)	31.034 (1.683)
	BPA	0.230 (0.020)	0.088 (0.003)	16.510 (2.034)	0.147 (0.013)	0.011 (0.002)	18.975 (0.780)
	AVE	0.779 (0.077)	0.147 (0.006)	159.517 (28.408)	0.504 (0.049)	0.182 (0.051)	82.719 (1.102)
	Glasso	0.119 (0.001)	0.274 (0.002)	2.798 (0.116)	0.037 (0.000)	0.012 (0.000)	9.786 (0.172)
Model 6	M1	**0.177 (0.002)**	0.171 (0.001)	**0.128 (0.020)**	1.982 (0.007)	0.629 (0.006)	93.839 (0.073)
	M2	0.301 (0.003)	0.203 (0.002)	2.879 (0.199)	**1.386 (0.007)**	**0.593 (0.006)**	**3.242 (0.062)**
	BIC	0.847 (0.068)	0.261 (0.005)	90.849 (6.230)	10.432 (1.044)	19.070 (2.642)	52.967 (0.897)
	BPA	0.397 (0.035)	0.162 (0.004)	40.191 (7.931)	4.760 (0.619)	6.980 (2.771)	34.222 (0.806)
	AVE	0.378 (0.011)	**0.161 (0.003)**	52.509 (2.354)	4.454 (0.138)	1.921 (0.129)	96.417 (0.025)
	Glasso	0.313 (0.004)	0.617 (0.012)	12.965 (0.368)	2.475 (0.007)	1.743 (0.019)	12.325 (0.172)

6. Application

In this section, we apply the proposed method of estimating $\boldsymbol{\Omega}$ for the linear discriminant analysis (LDA). To overcome the drawback of the classic LDA in high-dimensional data, we consider a new classification rule by using the proposed sparse precision estimate. A gene expression data set and hand movement data are used to evaluate the performance of the proposed classification rule.

6.1. LDA via the proposed estimate of Ω

In the classification problem, LDA is one commonly used technique. Consider a classification problem with K classes. Each observation belongs to some class $k \in 1, 2, \ldots, K$. Denote by C_k the class of training set observation $\boldsymbol{x}_i$. Let $\hat{\boldsymbol{\mu}}_k$ be the $p \times 1$ vector of the sample mean of the training data in class k, and $\hat{\boldsymbol{\Sigma}}_{LDA} = (1/(n-K)) \sum_{k=1}^{K} \sum_{i \in C_k} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k)'$ be the estimated within-class covariance matrix based on the training data. Then LDA classification rule is: classify a test observation $\boldsymbol{x}$ to class k^* if $k^* =$

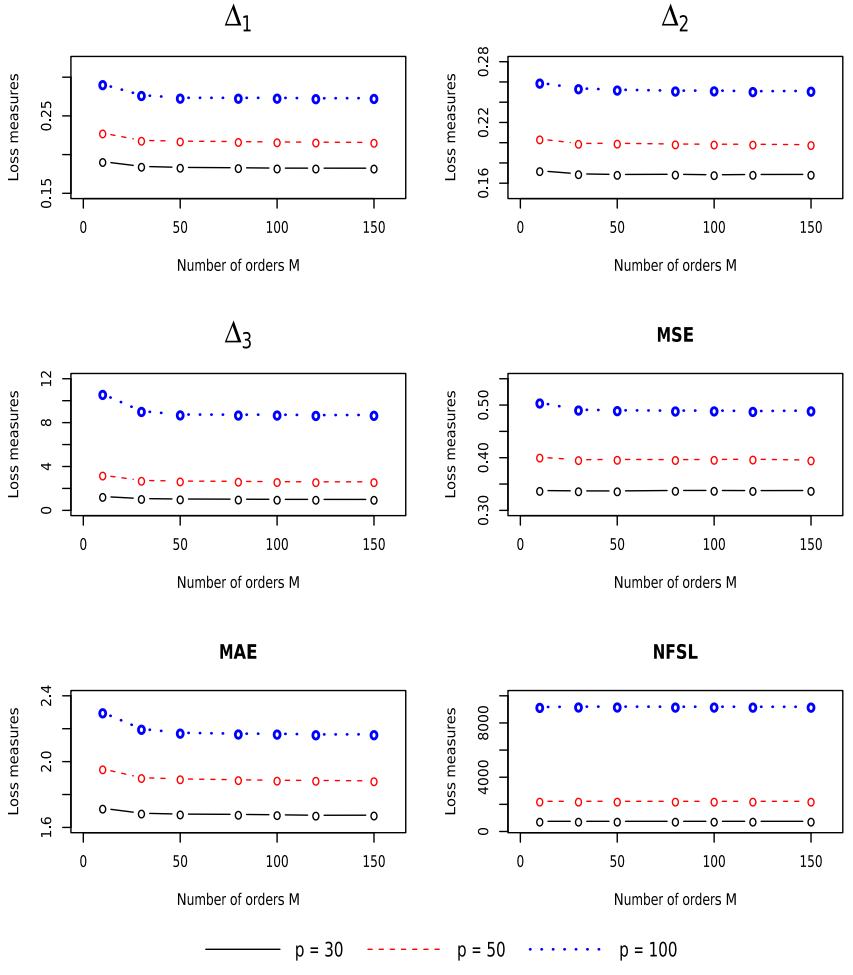

Figure 2. Plot of six loss measures of the proposed M1 against the number of orders *M* for *Model* 2

$\arg\max_k \eta_k(\boldsymbol{x})$, where

$$\eta_k(\boldsymbol{x}) = \boldsymbol{x}'\hat{\boldsymbol{\Sigma}}_{LDA}^{-1}\hat{\boldsymbol{\mu}}_k - \tfrac{1}{2}\hat{\boldsymbol{\mu}}_k'\hat{\boldsymbol{\Sigma}}_{LDA}^{-1}\hat{\boldsymbol{\mu}}_k + \log\pi_k \tag{12}$$

and π_k is the frequency of class k in the training data set. This method works well if the training sample size n is larger than the number of random variables p. However, when p is close to n, Bickel and Levina (33) [33] showed that LDA is asymptotically as bad as random guessing. Even worse, when $n < p$, the within-class covariance matrix $\hat{\boldsymbol{\Sigma}}_{LDA}$ is singular and the classical LDA breaks down. There are different approaches developed to address these problems in literature [34–38].

To overcome the singular issue, we suggest a classification rule using the proposed sparse estimate instead of $\hat{\boldsymbol{\Sigma}}_{LDA}^{-1}$ in (12). An accurate estimation of inverse within-class covariance matrix is expected to lead to accurate classification performance. In the following subsections, two real classification data sets are used to evaluate the performance of the proposed estimate, obtained respectively from M1 and M2, in comparison with other approaches,

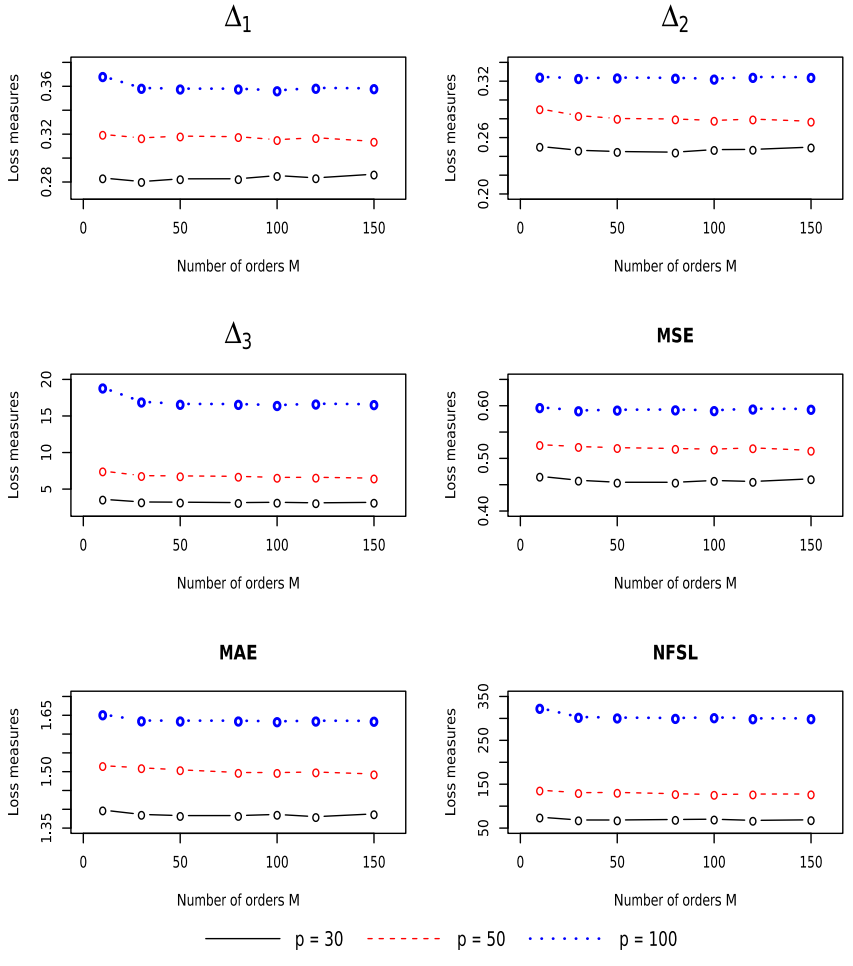

Figure 3. Plot of six loss measures of the proposed M2 against the number of orders *M* for *Model* 2

including BIC, BPA, AVE and Glasso. Apart from these, the generalized LDA [35], C5.0 [39] and diagonal linear discriminant analysis [40] are also considered, denoted by GLDA, C5 and DLDA. The GLDA replaces $\mathbf{\Sigma}_{LDA}^{-1}$ in (12) with the generalized precision matrix. C5 builds decision trees from a set of training data, using the concept of entropy. On each iteration of the algorithm, it iterates through every unused variable and calculates the entropy. It then selects the variable which has the smallest entropy value. The DLDA is a modification to LDA, where the off-diagonal elements of the pooled sample covariance matrix are set to be zeroes.

6.2. Lymphoma data

The data set includes two classes. It contains expression values for 2647 genetic probes and 77 samples, 58 of which are obtained from patients suffering from diffuse large B-cell lymphoma, while the remaining 19 samples are derived from follicular lymphoma type. Data are available online at http://ico2s.org/datasets/microarray.html. We randomly split

Table 4. Misclassification error of the proposed methods compared with other approaches for Lymphoma data.

Method	M1	M2	BIC	AVE	BPA	Glasso	GLDA	DLDA	C5
Error	7	6	16	11	9	8	13	7	9

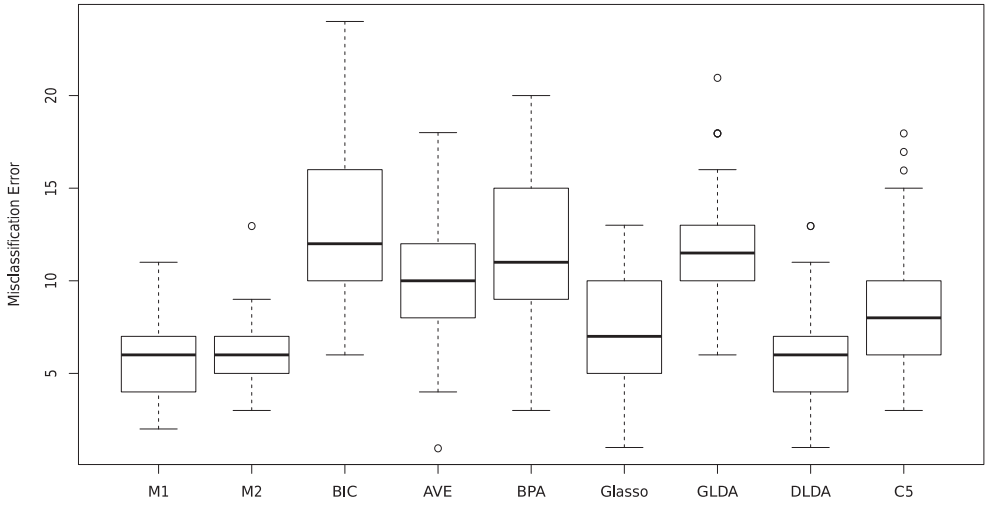

Figure 4. Boxplot of misclassification error comparison for proposed methods with other approaches under the randomly splitting training and testing data from Lymphoma data.

the samples into two groups: training set of 35 samples and testing set of 42 samples. Then the variable screening procedure is performed through two sample t-test. Specifically, for each variable, t-test is conducted against the two classes of the training data such that variables with large values of test statistics are ranked as significant variables, and the top 50 significant variables are selected for data classification. The results of misclassification error for each approach are summarized in Table 4. Overall, the proposed methods are better than other approaches. The M2 is the best with the minimum misclassification error. The M1 and M2 perform better than the BIC and AVE. Additionally, the AVE gives superior performance to the BIC in terms of smaller misclassification error as expected. The BIC and GLDA do not give the accurate classification.

Furthermore, we randomly partition 35 observations of the samples as a new training data set and the remaining 42 observations as a new testing data set. Figure 4 shows the boxplot of the misclassification errors for each method by repeating the above procedure over 50 times based on the top 50 significant gene expressions. It is clear that the M1, M2 and DLDA are the best, followed by C5, Glasso and AVE, which further confirms that an efficient way of organizing the available estimates will lead to a small misclassification error. In this example, both the M2 and DLDA perform quite well due to the underlying conditional independence between the gene variables. M2 appears to be better because of its smaller misclassification error and narrower width. In addition, the AVE performs better than BIC due to the superiority of the multiple orders over one order. The BIC, BPA and GLDA are not as good as other approaches.

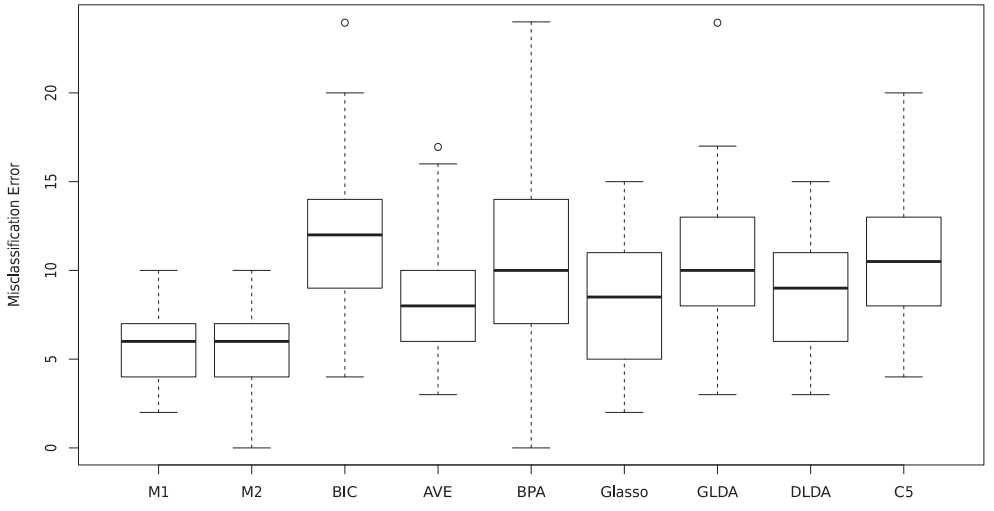

Figure 5. Boxplot of misclassification error comparison for proposed methods with other approaches under randomly selected 50 gene expressions from Lymphoma data.

Although DLDA shows comparable performance as M2 in the lymphoma example, we will demonstrate its drawbacks in the following. The reason DLDA works well, we believe, is that the contribution of every single variable of top 50 is relatively much more significant than the contribution of their interactions. As a result, the underlying precision matrix might be a diagonal matrix. To confirm our opinion, we assess the performance of each method using lymphoma data set based on a new group of 50 variables, which are randomly selected from all the 2647 gene expressions. Obviously, a single variable from these randomly selected 50 variables would not play a role as great as the top 50 significant variables. Therefore, their interactions are supposed to make some contributions, hence resulting in a sparse but not a diagonal precision matrix. In practice, we use the same partitioned training and testing data sets used for Figure 4, and randomly select 50 gene expressions as variables. Figure 5 displays the misclassification errors of each method from the above procedure. It is clear that the M2 performs much better than DLDA. The M1 still gives a good performance due to its accurate estimate.

6.3. Hand movement data

To evaluate the performance of the proposed methods in multiple classification problems, the second data set contains 15 classes of 24 observations each with each class referring to a hand movement type. The hand movement is represented as a two dimensional curve performed by the hand in a period of time, where each curve is characterized by 90 variables. The data are available online at https://archive.ics.uci.edu/ml/datasets/Libra+Movement. The data set is randomly split into the training set of 160 observations and testing set of 200 observations. Table 5 reports the misclassification errors for each approach. The proposed M1 dominates all the other methods attributed to the accurate precision matrix estimate. M2, especially DLDA, performs not well possibly due to the non-sparse structure of the underlying precision matrix. BPA and Glasso are comparable with BIC and AVE. GLDA does not provide a accurate classification.

Table 5. Misclassification error of the proposed methods compared with other approaches for Hand Movement data.

Method	M1	M2	BIC	AVE	BPA	Glasso	GLDA	DLDA	C5
Error	55	75	74	74	77	73	97	84	85

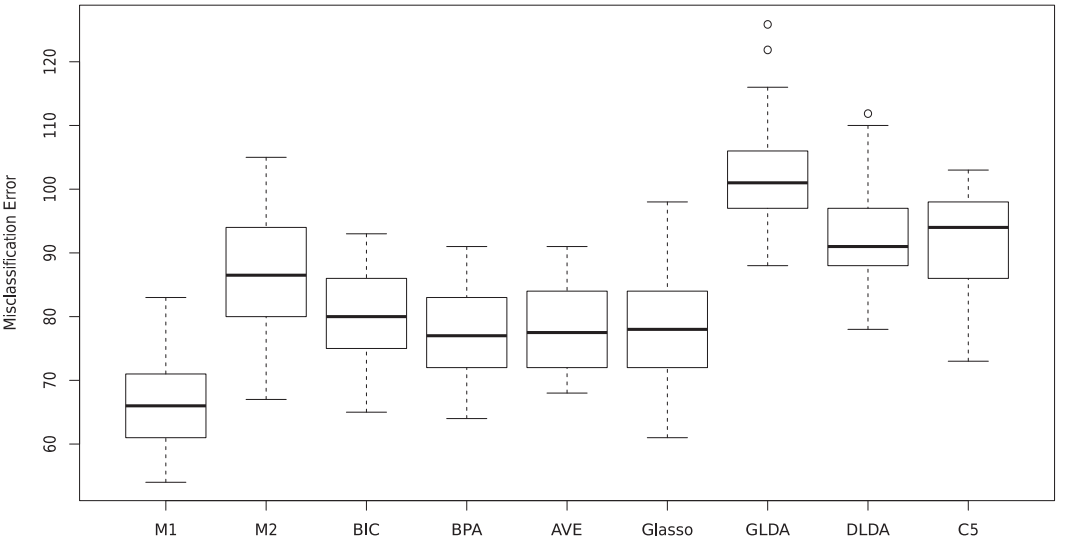

Figure 6. Boxplot of misclassification error comparison for proposed methods with other approaches under the randomly splitting training and testing data from Hand Movement data.

Moreover, we randomly partition 160 observations as a new training data set and the remaining 200 observations as a new testing data set. Figure 6 presents the boxplot of the misclassification errors by repeating the above procedure over 50 times. The proposed M1 outperforms the other approaches as it gives an accurate estimate. BIC, BPA, AVE and Glasso are comparable with each other. The performances of M2, C5 and DLDA are not as well as others. GLDA gives the highest misclassification error. This 15 classes data example demonstrates that the proposed method works consistently well in the multiple classification settings.

7. Discussion

In this paper, we have improved the Cholesky-based approach for sparse precision matrix estimation by introducing an ensemble method. Based on the modified Cholesky decomposition of a precision matrix, the proposed estimator is properly assembled from a set of multiple estimates of $\boldsymbol{T}$ and $\boldsymbol{D}$ under different orders of random variables. Hard thresholding technique is applied to the ensemble estimate of Cholesky factor matrix $\boldsymbol{T}$ to encourage the sparse structure. The resulting estimator does not require the prior knowledge of the order of variables. Although we employ the Lasso penalty in the regression (3) to solve the coefficients, other regularization methods can be considered, such as Ridge penalty or adaptive Lasso. The simulation studies show the superior performance of our proposed

method in terms of loss measures and ability of capturing sparsity. The advantage of considering multiple orders over one single order is illustrated by comparison of the proposed method with the BIC and BPA approaches.

Finally, we would like to remark that sometimes real data may include abnormal observations. Hence, robustness is a very important property we need to consider when proposing an estimator. Compared to the method (6), alternatively, we consider the ensemble estimate by the element-wise median of $\hat{\boldsymbol{T}}$ and $\hat{\boldsymbol{D}}$ instead of taking average

$$\hat{\boldsymbol{\Omega}} = \hat{\boldsymbol{T}}'\hat{\boldsymbol{D}}^{-1}\hat{\boldsymbol{T}} \text{ with } \hat{\boldsymbol{T}} = \text{med}(\hat{\boldsymbol{T}}_k),\ \hat{\boldsymbol{D}} = \text{med}(\hat{\boldsymbol{D}}_k).$$

This estimator is supposed to be more robust than the proposed method. It is also able to provide a sparse estimate for the precision matrix without a natural variable order, and is applicable in high dimensions. We will further investigate the robustness of this estimator in the future work.

Acknowledgement

The authors thank the Editor and referees for their insightful and helpful comments that have improved the original manuscript. Xiaoning Kang's research was supported by the Science Education Foundation of Liaoning Province (LN2019Q21), and the National Natural Science Foundation of China (71871047). Xinwei Deng's research was supported by the National Natural Science Foundation of China (71531004).

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Xiaoning Kang http://orcid.org/0000-0003-0394-6240
Xinwei Deng http://orcid.org/0000-0002-1560-2405

References

[1] Meinshausen N, Bhlmann P. High-dimensional graphs and variable selection with the lasso. Ann Statist. 2006;34(3):1436–1462.
[2] Yuan M, Lin Y. Model selection and estimation in the Gaussian graphical model. Biometrika. 2007;94(1):19–35.
[3] Friedman J, Hastie T, Tibshirani R. Sparse inverse covariance estimation with the graphical lasso. Biostatistics. 2008;9(3):432–441.
[4] Rocha GV, Zhao P, Yu B. A path following algorithm for sparse pseudo-likelihood inverse covariance estimation (splice). Preprint, 2008. arXiv:0807.3734.
[5] Rothman AJ, Bickel PJ, Levina E, et al. Sparse permutation invariant covariance estimation. Electron J Stat. 2008;2:494–515.
[6] Yuan M. Efficient computation of ℓ_1 regularized estimates in Gaussian graphical models. J Comput Graph Stat. 2008;17(4):809–826.
[7] Deng X, Yuan M. Large Gaussian covariance matrix estimation with Markov structures. J Comput Graph Stat. 2009;18(3):640–657.
[8] Yuan M. High dimensional inverse covariance matrix estimation via linear programming. J Mach Learn Res. 2010;11:2261–2286.

[9] Raskutti G, Yu B, Wainwright MJ, et al. Model Selection in Gaussian Graphical Models: High-Dimensional Consistency of ℓ_1-regularized MLE. In Advances in Neural Information Processing Systems. 2009; p. 1329–1336.
[10] Lam C, Fan J. Sparsistency and rates of convergence in large covariance matrix estimation. Ann Stat. 2009;37(6):4254–4278.
[11] Fan J, Fan Y, Lv J. High dimensional covariance matrix estimation using a factor model. J Econom. 2008;147(1):186–197.
[12] Xue L, Zou H. Regularized rank-based estimation of high-dimensional nonparanormal graphical models. Ann Statist. 2012;40(5):2541–2571.
[13] Wieringen WN, Peeters CF. Ridge estimation of inverse covariance matrices from high-dimensional data. Comput Stat Data Anal. 2016;103:284–303.
[14] Drton M, Perlman MD. A SINful approach to Gaussian graphical model selection. J Stat Plan Inference. 2008;138(4):1179–1200.
[15] Sun T, Zhang C. Comment: minimax estimation of large covariance matrices under ℓ-1 norm. Statist Sinica. 2012;22:1354–1358.
[16] Cheng Y, Lenkoski A. Hierarchical Gaussian graphical models: beyond reversible jump. Electron J Stat. 2012;6:2309–2331.
[17] Wang H. Bayesian graphical lasso models and efficient posterior computation. Bayesian Anal. 2012;7(4):867–886.
[18] Bhadra A, Mallick BK. Joint high-dimensional Bayesian variable and covariance selection with an application to eQTL analysis. Biometrics. 2013;69(2):447–457.
[19] Scutari M. On the prior and posterior distributions used in graphical modelling. Bayesian Anal. 2013;8(3):505–532.
[20] Mohammadi A, Wit EC. Bayesian structure learning in sparse Gaussian graphical models. Bayesian Anal. 2015;10(1):109–138.
[21] Pourahmadi M. Joint mean-covariance models with applications to longitudinal data: unconstrained parameterisation. Biometrika. 1999;86(3):677–690.
[22] Pourahmadi M. Foundations of time series analysis and prediction theory. New York: John Wiley & Sons; 2001.
[23] Chang C, Tsay RS. Estimation of covariance matrix via the sparse Cholesky factor with lasso. J Stat Plan Inference. 2010;140(12):3858–3873.
[24] Chen Z, Leng C. Local linear estimation of covariance matrices via Cholesky decomposition. Stat Sin. 2015;25(3):1249–1263.
[25] Zheng H, Tsui KW, Kang X, et al. Cholesky-based model averaging for covariance matrix estimation. Statist Theory Related Fields. 2017;1(1):48–58.
[26] Wagaman AS, Levina E. Discovering sparse covariance structures with the isomap. J Comput Graph Stat. 2009;18(3):551–572.
[27] Rajaratnam B, Salzman J. Best permutation analysis. J Multivar Anal. 2013;121:193–223.
[28] Tibshirani R. Regression shrinkage and selection via the lasso. J R Statist Soc Ser B (Methodological). 1996;58:267–288.
[29] Huang JZ, Liu N, Pourahmadi M, et al. Covariance matrix selection and estimation via penalised normal likelihood. Biometrika. 2006;93(1):85–98.
[30] Kang X, Xie C, Wang M. A Cholesky-based estimation for large-dimensional covariance matrices. J Appl Stat. 2019. in press. doi:10.1080/02664763.2019.1664424.
[31] Guo J, Levina E, Michailidis G, et al. Joint estimation of multiple graphical models. Biometrika. 2011;98(1):1–15.
[32] Dellaportas P, Pourahmadi M. Cholesky-GARCH models with applications to finance. Stat Comput. 2012;22(4):849–855.
[33] Bickel PJ, Levina E. Some theory for fisher's linear discriminant function,naive bayes', and some alternatives when there are many more variables than observations. Bernoulli. 2004;10(6):989–1010.
[34] Friedman JH. Regularized discriminant analysis. J Am Stat Assoc. 1989;84(405):165–175.

[35] Howland P, Park H. Generalizing discriminant analysis using the generalized singular value decomposition. IEEE Trans Pattern Anal Mach Intell. 2004;26(8):995–1006.
[36] Guo Y, Hastie T, Tibshirani R. Regularized linear discriminant analysis and its application in microarrays. Biostatistics. 2006;8(1):86–100.
[37] Fan J, Fan Y. High dimensional classification using features annealed independence rules. Ann Stat. 2008;36(6):2605–2637.
[38] Shao J, Wang Y, Deng X, et al. Sparse linear discriminant analysis by thresholding for high dimensional data. Ann Statist. 2011;39(2):1241–1265.
[39] Quinlan RC. 4.5: Programs for machine learning morgan. San Francisco, USA: Kaufmann Publishers Inc; 1993.
[40] Dudoit S, Fridlyand J, Speed TP. Comparison of discrimination methods for the classification of tumors using gene expression data. J Am Stat Assoc. 2002;97(457):77–87.
[41] Jiang X. Joint estimation of covariance matrix via Cholesky Decomposition [Doctoral dissertation]. 2012.

Appendix

Lemma A.1: *Assume a positive definite matrix* $\boldsymbol{\Omega}$ *has corresponding modified Cholesky decomposition*

$$\boldsymbol{\Omega} = \boldsymbol{T}'\boldsymbol{D}^{-1}\boldsymbol{T},$$

and there exist c_1 *and* c_2 *such that* $0 < c_1 < sv_p(\boldsymbol{\Omega}) \leq sv_1(\boldsymbol{\Omega}) < c_2 < \infty$, *then there exist constants* g_1 *and* g_2 *such that*

$$g_1 < sv_p(\boldsymbol{T}) \leq sv_1(\boldsymbol{T}) < g_2$$
$$g_1 < sv_p(\boldsymbol{D}) \leq sv_1(\boldsymbol{D}) < g_2.$$

Also we have

$$\|\boldsymbol{T}\|_F^2 = O(1) \quad \textit{and} \quad \|\boldsymbol{D}\|_F^2 = O(1).$$

The proof of Lemma A.1 can be found in Jiang [41], thus is omitted here.

Let $\boldsymbol{\Omega}_{0\pi} = \boldsymbol{T}'_{0\pi}\boldsymbol{D}_{0\pi}^{-1}\boldsymbol{T}_{0\pi}$ be the MCD for the true precision matrix under a variable order π. Let $Z_{\pi_k} = \{(j,k) : k < j, t_{0jk}^{(\pi_k)} \neq 0\}$ be the collection of nonzero elements in the lower triangular part of matrix $\boldsymbol{T}_{0\pi_k}$. Denote by s the maximum of the cardinality of Z_{π_k} for $k = 1, 2, \ldots, M$. Then we have the following Lemma.

Lemma A.2: *Assume data are from Normal distribution* $N(\mathbf{0}, \boldsymbol{\Omega}_0^{-1})$. *Under (11), assume that the tuning parameters* $\lambda_{\pi(j)}$ *in (3) satisfy* $\sum_{j=1}^{p} \lambda_{\pi(j)} = O(\log(p)/n)$. *Then* $\hat{\boldsymbol{T}}_\pi$ *and* $\hat{\boldsymbol{D}}_\pi$ *have the following consistent properties*

$$\|\hat{\boldsymbol{T}}_\pi - \boldsymbol{T}_{0\pi}\|_F = O_p(\sqrt{s\log(p)/n}),$$
$$\|\hat{\boldsymbol{D}}_\pi - \boldsymbol{D}_{0\pi}\|_F = O_p(\sqrt{p\log(p)/n}).$$

Proof of Lemma A.2: The proof is the same as that of Theorem 3.1 in Jiang [41] except two key differences. The first one is Jiang [41] considered data that are from different groups but share the similar structure. $\boldsymbol{\Sigma}^{(j)}$ was used to indicate the covariance matrix for the jth group, $j = 1, 2, \ldots, J$. Hence, the proof of Lemma A.2 is a special case with the number of data group $J = 1$. The second difference between Theorem 3.1 in [41] and our Lemma A.2 is the penalty term. Jiang (2012) [41] had two penalties λ and β satisfying $\lambda + \beta = O(log(p)/n)$. While in our proposed method, such assumption on the penalty terms is replaced with $\sum_{j=1}^{p} \lambda_{\pi(j)} = O(\log(p)/n)$. Hence, we omit the detailed proof here. ■

Proof of Theorem 4.1: Let $\boldsymbol{\Delta}_T = \tilde{\boldsymbol{T}}_\delta - \boldsymbol{T}_0$ and $\boldsymbol{\Delta}_D = \tilde{\boldsymbol{D}} - \boldsymbol{D}_0$, then $\|\tilde{\boldsymbol{\Omega}}_\delta - \boldsymbol{\Omega}_0\|_F^2$ can be decomposed as follows,

$$\begin{aligned}\|\tilde{\boldsymbol{\Omega}}_\delta - \boldsymbol{\Omega}_0\|_F^2 &= \|\tilde{\boldsymbol{T}}_\delta' \tilde{\boldsymbol{D}}^{-1} \tilde{\boldsymbol{T}}_\delta - \boldsymbol{T}_0' \boldsymbol{D}_0^{-1} \boldsymbol{T}_0\|_F^2 \\ &= \|(\boldsymbol{\Delta}_T' + \boldsymbol{T}_0')\tilde{\boldsymbol{D}}^{-1}(\boldsymbol{\Delta}_T + \boldsymbol{T}_0) - \boldsymbol{T}_0' \boldsymbol{D}_0^{-1} \boldsymbol{T}_0\|_F^2 \\ &\le \|\boldsymbol{\Delta}_T' \tilde{\boldsymbol{D}}^{-1} \boldsymbol{T}_0\|_F^2 + \|\boldsymbol{T}_0' \tilde{\boldsymbol{D}}^{-1} \boldsymbol{\Delta}_T\|_F^2 + \|\boldsymbol{\Delta}_T' \tilde{\boldsymbol{D}}^{-1} \boldsymbol{\Delta}_T\|_F^2 + \|\boldsymbol{T}_0'(\tilde{\boldsymbol{D}}^{-1} - \boldsymbol{D}_0^{-1})\boldsymbol{T}_0\|_F^2.\end{aligned}$$

Next, we bound these four terms separately. From (11) and Lemma A.1, we have $\|\boldsymbol{T}_0\|_F^2 = O(1)$ and $\|\boldsymbol{D}_0\|_F^2 = O(1)$. Then $\|\tilde{\boldsymbol{D}}\|_F^2 = \|\tilde{\boldsymbol{D}} - \boldsymbol{D}_0 + \boldsymbol{D}_0\|_F^2 \le \|\boldsymbol{\Delta}_D\|_F^2 + \|\boldsymbol{D}_0\|_F^2 = O_p(1)$. Similarly we have $\|\tilde{\boldsymbol{D}}^{-1}\|_F^2 = O_p(1)$. It is because that the single values of $\boldsymbol{\Omega}$ are bounded, then the single values of $\boldsymbol{\Omega}^{-1}$ are bounded. This together with Lemma A.1 results in $\|\boldsymbol{D}_0^{-1}\|_F^2 = O_p(1)$, and hence $\|\tilde{\boldsymbol{D}}^{-1}\|_F^2 = O_p(1)$. Hence, it is easy to obtain

$$\|\boldsymbol{\Delta}_T' \tilde{\boldsymbol{D}}^{-1} \boldsymbol{T}_0\|_F^2 \le \|\boldsymbol{\Delta}_T'\|_F^2 \|\tilde{\boldsymbol{D}}^{-1}\|_F^2 \|\boldsymbol{T}_0\|_F^2 = O_p(\|\boldsymbol{\Delta}_T\|_F^2).$$

Apply the same principle, we have $\|\boldsymbol{T}_0' \tilde{\boldsymbol{D}}^{-1} \boldsymbol{\Delta}_T\|_F^2 = O_p(\|\boldsymbol{\Delta}_T\|_F^2)$. For the third term,

$$\|\boldsymbol{\Delta}_T' \tilde{\boldsymbol{D}}^{-1} \boldsymbol{\Delta}_T\|_F^2 \le \|\boldsymbol{\Delta}_T'\|_F^2 \|\tilde{\boldsymbol{D}}^{-1}\|_F^2 \|\boldsymbol{\Delta}_T\|_F^2 = o_p(\|\boldsymbol{\Delta}_T\|_F^2).$$

For the fourth term,

$$\|\boldsymbol{T}_0'(\tilde{\boldsymbol{D}}^{-1} - \boldsymbol{D}_0^{-1})\boldsymbol{T}_0\|_F^2 \le \|\boldsymbol{T}_0'\|_F^2 \|\tilde{\boldsymbol{D}}^{-1} - \boldsymbol{D}_0^{-1}\|_F^2 \|\boldsymbol{T}_0\|_F^2 = O_p(\|\tilde{\boldsymbol{D}} - \boldsymbol{D}_0\|_F^2).$$

Therefore, we have

$$\|\tilde{\boldsymbol{\Omega}}_\delta - \boldsymbol{\Omega}_0\|_F^2 = O_p(\|\tilde{\boldsymbol{T}}_\delta - \boldsymbol{T}_0\|_F^2) + O_p(\|\tilde{\boldsymbol{D}} - \boldsymbol{D}_0\|_F^2). \tag{A1}$$

Next, we derive $O_p(\|\tilde{\boldsymbol{T}}_\delta - \boldsymbol{T}_0\|_F^2)$ and $O_p(\|\tilde{\boldsymbol{D}} - \boldsymbol{D}_0\|_F^2)$. For any variable order $\pi_k, k = 1, 2, \ldots, M$, we have

$$\begin{aligned}\|\hat{\boldsymbol{T}}_k - \boldsymbol{T}_0\|_F^2 &= \|\boldsymbol{P}_{\pi_k} \hat{\boldsymbol{T}}_{\pi_k} \boldsymbol{P}_{\pi_k}' - \boldsymbol{P}_{\pi_k} \boldsymbol{T}_{0\pi_k} \boldsymbol{P}_{\pi_k}'\|_F^2 \\ &= \|\boldsymbol{P}_{\pi_k}(\hat{\boldsymbol{T}}_{\pi_k} - \boldsymbol{T}_{0\pi_k})\boldsymbol{P}_{\pi_k}'\|_F^2 \\ &= \|\hat{\boldsymbol{T}}_{\pi_k} - \boldsymbol{T}_{0\pi_k}\|_F^2 \\ &= O_p(s\log(p)/n),\end{aligned}$$

where the third equality results from the fact that the Frobenius norm of a matrix is invariant on the permutation matrix, and the fourth equality is provided by Lemma A.2. This leads to the consistent property of $\tilde{\boldsymbol{T}} = (1/M)\sum_{k=1}^M \hat{\boldsymbol{T}}_k$ in (6) as follows,

$$\begin{aligned}\|\tilde{\boldsymbol{T}} - \boldsymbol{T}_0\|_F^2 &= \left\| \frac{1}{M} \sum_{k=1}^M \hat{\boldsymbol{T}}_k - \boldsymbol{T}_0 \right\|_F^2 \\ &= \frac{1}{M^2} \left\| \sum_{k=1}^M \hat{\boldsymbol{T}}_k - M\boldsymbol{T}_0 \right\|_F^2 \\ &\le \frac{1}{M^2} \sum_{k=1}^M \left\| \hat{\boldsymbol{T}}_k - \boldsymbol{T}_0 \right\|_F^2 \\ &= O_p\left(\frac{s\log(p)}{nM} \right).\end{aligned}$$

Similarly, we can establish

$$\|\tilde{\boldsymbol{D}} - \boldsymbol{D}_0\|_F^2 = O_p\left(\frac{p\log(p)}{nM}\right) = O_p\left(\frac{\log(p)}{n}\right). \tag{A2}$$

From the property of Frobenius norm and consistency of $\tilde{\boldsymbol{T}}$, we have

$$\begin{aligned}\|\tilde{\boldsymbol{T}}_\delta - \boldsymbol{T}_0\|_F^2 &\le \|\tilde{\boldsymbol{T}}_\delta - \tilde{\boldsymbol{T}}\|_F^2 + \|\tilde{\boldsymbol{T}} - \boldsymbol{T}_0\|_F^2 \\ &\le \delta^2 p^2 + O_p\left(\frac{s\log(p)}{nM}\right) \\ &= O_p\left(\frac{(s+p^2)\log(p)}{nM}\right) \\ &= O_p\left(\frac{p\log(p)}{n}\right),\end{aligned} \tag{A3}$$

where the fourth equality is given by $s \le p^2$. Therefore, from (A1), together with the consistent properties of $\tilde{\boldsymbol{D}}$ and $\tilde{\boldsymbol{T}}_\delta$ in (A2) and (A3), it is easy to obtain

$$\begin{aligned}\|\tilde{\boldsymbol{\Omega}}_\delta - \boldsymbol{\Omega}_0\|_F^2 &= O_p(\|\tilde{\boldsymbol{T}}_\delta - \boldsymbol{T}_0\|_F^2) + O_p(\|\tilde{\boldsymbol{D}} - \boldsymbol{D}_0\|_F^2) \\ &= O_p\left(\frac{p\log(p)}{n}\right) + O_p\left(\frac{\log(p)}{n}\right) \\ &= O_p\left(\frac{p\log(p)}{n}\right) \\ &\xrightarrow{P} 0.\end{aligned}$$

■