Computational Statistics and Data Analysis 188 (2023) 107823

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Ensemble LDA via the modified Cholesky decomposition

Zhenguo Gao [a], Xinye Wang [a], Xiaoning Kang [b,*]

[a] School of Mathematical Sciences, Shanghai Jiao Tong University, China
[b] Institute of Supply Chain Analytics and International Business College, Dongbei University of Finance and Economics, Dalia, China

ARTICLE INFO

Article history:
Received 23 October 2022
Received in revised form 12 July 2023
Accepted 13 July 2023
Available online 26 July 2023

Keywords:
Ensemble learning
High-dimensional
Precision matrix
Variable ordering

ABSTRACT

A binary classification problem in the high-dimensional settings is studied via the ensemble learning with each base classifier constructed from the linear discriminant analysis (LDA), and these base classifiers are integrated by the weighted voting. The precision matrix in the LDA rule is estimated by the modified Cholesky decomposition (MCD), which is able to provide us with a set of precision estimates by considering multiple variable orderings, and hence yield a group of different LDA classifiers. Such available LDA classifiers are then integrated to improve the classification performance. The simulation and the application studies are conducted to demonstrate the merits of the proposed method.

© 2023 Elsevier B.V. All rights reserved.

1. Introduction

We study the discriminant analysis with two classes in this work, which is a common problem in a wide range of real-world applications. There are many existing classification methods proposed in the literature, among which the linear discriminant analysis (LDA) is a classic and routinely used technique, as it is known that the LDA performs well in the low dimensions. However, traditional LDA is inapplicable for high-dimensional data since it is asymptotically as bad as random guessing (Shao et al., 2011), and even fails to work when the number of variables is much larger than the sample size. In order to enable LDA to accommodate the high-dimensional classification, many scholars have proposed novel methods, which can be roughly divided into two categories.

The approaches in the first category directly estimate the discriminant vector $\boldsymbol{\Omega}\ \boldsymbol{\delta}$ in the LDA decision rule, where $\boldsymbol{\Omega}$ is the precision matrix of variables, and $\boldsymbol{\delta}$ denotes the difference of the mean vectors between two classes. For example, Cai and Liu (2011) proposed an ℓ_1 minimization method via the infinity norm constraint by exploiting the approximate sparsity of $\boldsymbol{\Omega\delta}$, obtaining a classifier called the linear programming discriminant rule. Wu et al. (2009) investigated a penalized risk minimization method by regularizing the usual LDA loss function, achieving effective dimension reduction and feature selection. Clemmensen et al. (2011) suggested a sparse discriminant analysis based on the optimal scoring interpretation of LDA, where the classification and feature selection were performed simultaneously. Motivated by the least squares formulation of LDA, Mai et al. (2012) introduced to estimated the discriminant vector $\boldsymbol{\Omega\delta}$ for ultra-high-dimensional data by converting the LDA problem into a task of solving Lasso regression. Recently, Cai and Huang (2018) proposed several nonconvex penalty-based LDA algorithms, including the ℓ_0-based and sorted ℓ_1-based LDA methods. Liu et al. (2019) considered the structure information on features and proposed to impose a structure-based sparse penalty on the discriminant vector. The LDA were directly applied in these methods using different formulations with regularization on $\boldsymbol{\Omega\delta}$.

* Corresponding author at: 217 Jianshan Street, Dongbei University of Finance and Economics, Dalian, 116025, China.
E-mail address: xiaoningmike@126.com (X. Kang).

https://doi.org/10.1016/j.csda.2023.107823
0167-9473/© 2023 Elsevier B.V. All rights reserved.

The approaches in the second category focus on improving the precision matrix estimate by inducing its sparsity, which is subsequently used to replace the inverse sample covariance in the traditional LDA rule, hence achieving a better classification accuracy. A popular approach to obtain a sparse precision estimator is Glasso, which imposes an ℓ_1 penalty on the negative log-likelihood (Yuan and Lin, 2007). Later, some papers studied its theoretical property and contributed efficient algorithms for solving the Glasso objective (Friedman et al., 2007; Zhang et al., 2018). Another powerful tool for the sparse estimation of precision matrix is the modified Cholesky decomposition (MCD) proposed by Pourahmadi (1999). It can encourage the sparsity easily by imposing the penalty on the linear regressions and simultaneously guarantee the positive definiteness of the estimated matrix. However, there is an ordering issue when implementing the MCD in the sense that different variable orderings will lead to different estimates (Chang and Tsay, 2010; Kang et al., 2020; Wang et al., 2023). In practical applications, there is usually no pre-knowledge on the variable ordering. Hence some scholars suggested to determine an ordering via some criteria before applying the MCD (Dellaportas and Pourahmadi, 2012; Rajaratnam and Salzman, 2013). Nonetheless, these methods still rely on the assumption that variables have a potential ordering scheme. If there exists no intrinsic variable ordering in data, these methods may identify an incorrect ordering, which may in turn reduce the estimation accuracy and hence the classification performance. To overcome such ordering issue in the MCD, Zheng et al. (2017) developed a model averaging idea by considering multiple variable orderings to alleviate their effects and obtained accurate estimations.

In addition to the LDA for the discriminant analysis, the scholars have suggested the classification rules that allow the covariance matrices across classes to be different, such as the eigenvalue decomposition discriminant analysis (EDDA) and mixture discriminant analysis (MDA). The EDDA (Breiman, 1996) re-parameterized the covariance matrices for each class in terms of the eigenvalue decomposition. It provides a class of 14 possible models for the covariances, allowing the data to automatically be chosen among them via cross validation, which is more flexible than LDA. The main advantage of EDDA is the simple geometric interpretation by the concepts of shape, volume and orientation from the eigenvalue decomposition of covariances. However, it may fail when the number of variables is much larger than the sample size since its construction is based on the sample covariances for each class. Additionally, the MDA (Hastie and Tibshirani, 1996) generalized the LDA by assuming a mixture normal distribution rather than a single normal distribution for each class data. Although it contains the LDA as a special case, the MDA needs to determine the number of mixtures for each class.

In this paper, we study an ensemble classification rule for a two-class problem. The principle of the ensemble learning is to utilize certain randomization in the design of base classifiers, which is an effective modeling technique in machine learning, and many papers have contributed to investigating the ensemble learning. For example, Breiman (1996) proposed the famous Bagging algorithm, which integrates the classifiers based on training sets that are generated through the bootstrap. Random forests proposed by Breiman (2001) is an extension of the Bagging, where randomized feature selection is conducted on top of the Bagging. Adaboost introduced by Freund and Schapire (1997) is another classic ensemble learning algorithm, which improves the performance of the base classifier by adjusting the training sets. Ho (1998) studied a random subspaces method which randomly selects features for constructing base classifiers. In order to create training data for these base classifiers, Rodriguez et al. (2006) randomly split the feature set into multiple subsets and applied the principal component analysis to each of them. Durrant and Kabán (2015) employed random projections to create weak learners for classification purpose and made further theoretical analysis for the ensembles.

In this article, we adopt the model averaging idea to propose an ensemble Cholesky-based algorithm for high-dimensional binary classification problem based on the LDA. The MCD technique enables us to obtain a set of precision matrix estimates under different variable orderings. Accordingly, we can construct a set of LDA classifiers with each classifier corresponding to a precision estimate. These LDA classifiers are then integrated to produce an ensemble classifier based on the weighted voting technique.

The rest of the paper is organized as follows. Section 2 introduces the proposed algorithm as well as the MCD technique and voting in ensemble learning. The asymptotically theoretical results are established in Section 3. Section 4 reports the simulation results. Section 5 analyzes two real disease datasets. Section 6 concludes the paper with discussions.

2. Method

2.1. Model and algorithm

The proposed method is a two-step algorithm. In the first step, we construct LDA classifiers with different precision estimates with respect to multiple variable orderings in the MCD. The second step is to integrate such classifiers via the weighted voting of ensemble learning.

Suppose $\boldsymbol{X} = (X_1, \ldots, X_p)^{\mathrm{T}}$ is a p-dimensional vector of random variables, which follows multivariate normal distributions with different means for two classes, but sharing the same covariance matrix as follows

$$G_1 : \boldsymbol{X}|y = 0 \sim N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}) \quad \text{and} \quad G_2 : \boldsymbol{X}|y = 1 \sim N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}), \tag{1}$$

where $y \in \{0, 1\}$ is the class label. Denote the precision matrix by $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$. Accordingly, the class label predicted from LDA for a new observation $\boldsymbol{x}$ is

$$h(\boldsymbol{x}) = \mathbb{I}\left\{\ln\frac{\pi_0}{\pi_1} + \left(\boldsymbol{x} - \frac{\boldsymbol{\mu}_0 + \boldsymbol{\mu}_1}{2}\right)^{\mathrm{T}} \boldsymbol{\Omega}\left(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1\right) < 0\right\}, \tag{2}$$

where $\mathbb{I}(\cdot)$ is the indicator function. Here π_0 and π_1 are the prior probabilities for the two classes. Denote the observed data by $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ with n_0 samples from class 0 and n_1 samples belonging to class 1. In practice, the population parameters $\pi_0, \pi_1, \boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$ in Equation (2) are estimated from samples as

$$\hat{\pi}_0 = \frac{n_0}{n}, \quad \hat{\pi}_1 = \frac{n_1}{n}, \quad \hat{\boldsymbol{\mu}}_0 = \frac{1}{n_0}\sum_{y_i=0}\boldsymbol{x}_i \quad \text{and} \quad \hat{\boldsymbol{\mu}}_1 = \frac{1}{n_1}\sum_{y_i=1}\boldsymbol{x}_i. \tag{3}$$

In the classification rule (2), an appropriate and sparse estimate of precision matrix is desired to accommodate high-dimensional data and hence improve the classification performance. We thus propose to obtain a set of sparse precision estimates from MCD under multiple variable orderings, leading to different base classifiers as in (2). Then the weighted voting is subsequently employed to predict the class label for a new observation by assigning reasonable weights to each of the constructed base classifiers. The weights are determined based on the misclassification errors from the base classifiers in the way that a larger weight value would be assigned to the classifier that produces a relatively smaller misclassification error, such that the classifier with a good performance is allowed to contribute more in predicting the class label. The proposed classification procedure is presented in Algorithm 1. Sections 2.2 and 2.3 introduce the MCD and weighted voting techniques respectively that are involved in Algorithm 1.

Algorithm 1: Ensemble LDA algorithm via the MCD.

Input: Training set $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$;
Variable orderings group $\sigma = \{\sigma_1, \ldots, \sigma_M\}$;
LDA classifier $\mathfrak{L}$;
MCD method $\mathfrak{M}$.

Step 1. Construct the LDA classifier with the sparse precision estimate from the MCD under each variable ordering, and compute the misclassification errors E_t of each base classifier on training set:

for $t = 1, 2, \ldots, M$ **do**
$\hat{\boldsymbol{\Omega}}_t = \mathfrak{M}(S, \sigma_t)$;
$\hat{h}_t = \mathfrak{L}(S, \hat{\boldsymbol{\Omega}}_t)$;
$E_t = \left(\sum_{i=1}^n \mathbb{I}(\hat{h}_t(\boldsymbol{x}_i) \neq y_i)\right)/n$;
end

Step 2. Determine the weights according to E_t and ensemble base classifiers via weighted voting:

for $t = 1, 2, \ldots, M$ **do**
$w_t = e^{-E_t}/\left(\sum_{i=1}^M e^{-E_i}\right)$;
end

Output: The proposed classifier $H(\boldsymbol{x}) = \underset{j\in\{0,1\}}{\arg\max}\sum_{t=1}^M w_t \mathbb{I}(h_t(\boldsymbol{x}) = j)$.

2.2. Modified Cholesky decomposition

This section introduces the MCD approach for the sparse estimation of the precision matrix. Without losing of generality, we assume the expectation of random vector $E\boldsymbol{X} = \boldsymbol{0}$. The precision matrix $\boldsymbol{\Omega}$ can be decomposed by a lower triangular matrix $\boldsymbol{T}$ and a diagonal matrix $\boldsymbol{D}$ with positive diagonal entries as follows

$$\boldsymbol{\Omega} = \boldsymbol{T}^{\mathrm{T}}\boldsymbol{D}^{-1}\boldsymbol{T}. \tag{4}$$

The Cholesky factor matrices $(\boldsymbol{T}, \boldsymbol{D})$ are constructed with each variable X_j regressed on its previous variables $X_1, \ldots, X_{j-1}$. More specifically, let $X_1 = \epsilon_1$, and define

$$X_j = \sum_{q=1}^{j-1}\left(-t_{jq}\right)X_q + \epsilon_j = -\boldsymbol{Z}_j^{\mathrm{T}}\boldsymbol{t}_j + \epsilon_j, \quad j = 2, \ldots, p, \tag{5}$$

where $\boldsymbol{Z}_j = (X_1, \ldots, X_{j-1})^{\mathrm{T}}$, $\boldsymbol{t}_j = (t_{j1}, \ldots, t_{j,j-1})^{\mathrm{T}}$ and ϵ_j is the error term from the jth linear regression with $E(\epsilon_j) = 0$ and $Var(\epsilon_j) = d_j^2$. Write $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_p)^{\mathrm{T}}$ and $\boldsymbol{D} = Var(\boldsymbol{\epsilon}) = diag(d_1^2, \ldots, d_p^2)$. Then Equation (5) is written as

$$\boldsymbol{\epsilon} = \boldsymbol{T}\boldsymbol{X}, \tag{6}$$

where

$$\boldsymbol{T} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ t_{21} & 1 & 0 & \dots & 0 \\ t_{31} & t_{32} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ t_{p1} & t_{p2} & \dots & t_{p,p-1} & 1 \end{pmatrix}.$$

Taking $Var(\cdot)$ on both sides of (6) yields Equation (4) after a simple algebra. Therefore, the challenging precision estimation problem is converted into linear regression estimation through the MCD, which is relatively easy to solve and understand. Furthermore, the sparsity of the precision estimate can be induced by the sparsity in the Cholesky factor matrix $\boldsymbol{T}$, which can be easily encouraged via imposing regularization, e.g. Lasso penalty, on the linear regressions (5) as follows (Huang et al., 2006; Kang and Wang, 2021)

$$\hat{\boldsymbol{t}}_j = \arg\min_{\boldsymbol{t}_j} \sum_{i=1}^{n} \left[(\boldsymbol{x}_i)_j + (\boldsymbol{z}_i)_j^{\mathrm{T}} \boldsymbol{t}_j \right]^2 + \lambda_j \|\boldsymbol{t}_j\|_1, \quad j = 2, \dots, p, \tag{7}$$

where $(\boldsymbol{x}_i)_j$ is the jth element of $\boldsymbol{x}_i$, $(\boldsymbol{z}_i)_j = ((\boldsymbol{x}_i)_1, \dots, (\boldsymbol{x}_i)_{j-1})^{\mathrm{T}}$ and $\|\cdot\|_1$ stands for the vector L_1 norm.

It is seen from (7) that the variable ordering plays a critical role in the MCD approach, because different orderings would lead to regressions with different coefficients, which are entries of the Cholesky factor matrix $\boldsymbol{T}$, thus resulting in different MCD estimates. Enlightened by the model averaging idea, we uniformly generate M different permutations of $\{1, \dots, p\}$ from all the possible $p!$ with equal probability, denoted by $\sigma_1, \dots, \sigma_M$. Accordingly, an estimate of $\boldsymbol{\Omega}$ regarding the variable ordering σ_k is

$$\hat{\boldsymbol{\Omega}}_{\sigma_k} = \hat{\boldsymbol{T}}_{\sigma_k}^{\mathrm{T}} \hat{\boldsymbol{D}}_{\sigma_k}^{-1} \hat{\boldsymbol{T}}_{\sigma_k}, \quad k = 1, \dots, M.$$

Let $\boldsymbol{P}_{\sigma_k}$ be the permutation matrix corresponding to σ_k. Transforming back to the original ordering, we have

$$\begin{aligned} \hat{\boldsymbol{\Omega}}_k = \boldsymbol{P}_{\sigma_k} \hat{\boldsymbol{\Omega}}_{\sigma_k} \boldsymbol{P}_{\sigma_k}^{\mathrm{T}} &= \boldsymbol{P}_{\sigma_k} \hat{\boldsymbol{T}}_{\sigma_k}^{\mathrm{T}} \hat{\boldsymbol{D}}_{\sigma_k}^{-1} \hat{\boldsymbol{T}}_{\sigma_k} \boldsymbol{P}_{\sigma_k}^{\mathrm{T}} \\ &= (\boldsymbol{P}_{\sigma_k} \hat{\boldsymbol{T}}_{\sigma_k}^{\mathrm{T}} \boldsymbol{P}_{\sigma_k}^{\mathrm{T}})(\boldsymbol{P}_{\sigma_k} \hat{\boldsymbol{D}}_{\sigma_k}^{-1} \boldsymbol{P}_{\sigma_k}^{\mathrm{T}})(\boldsymbol{P}_{\sigma_k} \hat{\boldsymbol{T}}_{\sigma_k} \boldsymbol{P}_{\sigma_k}^{\mathrm{T}}) \\ &\triangleq \hat{\boldsymbol{T}}_k^{\mathrm{T}} \hat{\boldsymbol{D}}_k^{-1} \hat{\boldsymbol{T}}_k, \end{aligned}$$

where $\hat{\boldsymbol{\Omega}}_k$, $\hat{\boldsymbol{T}}_k$ and $\hat{\boldsymbol{D}}_k$ represent the estimates of $\boldsymbol{\Omega}$, $\boldsymbol{T}$ and $\boldsymbol{D}$ under the variable ordering σ_k. As a result, the MCD technique provides a set of different precision matrix estimates $\hat{\boldsymbol{\Omega}}_k$, which can be later used to construct the proposed ensemble LDA classifier.

Here we point out that a regularized estimate of $\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1$ can provide some further improvements on the individual LDA classifier. For example, using a hard-thresholding operator will produce a better estimator of $\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1$ than the sample estimate in (3). However, in this work, our primary focus is that the MCD provides a chance of constructing multiple LDA classifiers, and we investigate how to efficiently ensemble such LDA classifiers to conduct discriminant analysis. Besides, there will be more tuning parameters if we regularize the difference $\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1$ with their optimal values usually selected by cross validation, which may remarkably increase the computational burden.

2.3. Weighted voting

Ensemble learning has a long and successful history in machine learning. By combining multiple weak learners, the generalization ability of an ensemble learner is often much stronger than the individual learner. There are several combination strategies for different types of outputs in the ensemble learning. For classification problems considered in this paper, a commonly used method is the weighted voting. Denote by $h_i, i = 1, \dots, M$, the base classifier for the binary classification problem with the classes G_1 and G_2. Let $h_i^1(\boldsymbol{x})$ and $h_i^2(\boldsymbol{x})$ record the classification result from the classifier h_i for a new sample $\boldsymbol{x}$. Specifically, if the classifier h_i predicts the sample $\boldsymbol{x}$ to be G_1, then $h_i^1(\boldsymbol{x}) = 1$ and $h_i^2(\boldsymbol{x}) = 0$. Otherwise $h_i^1(\boldsymbol{x}) = 0$ and $h_i^2(\boldsymbol{x}) = 1$. Then the proposed ensemble classifier assigns the new sample $\boldsymbol{x}$ to G_1 if

$$\sum_{i=1}^{M} w_i h_i^1(\boldsymbol{x}) \geq \sum_{i=1}^{M} w_i h_i^2(\boldsymbol{x}),$$

where w_i is the weight of classifier h_i with $w_i \geq 0$ and $\sum_{i=1}^{M} w_i = 1$.

In the proposed method, the weight for each base classifier is determined according to the misclassification error from the corresponding classifier on the training data, which is denoted by $E_t, t = 1, \dots, M$. By the principle that a smaller misclassification error corresponds to a larger weight, there are three common ways of weight assignment as follows for $t = 1, \dots, M$

$$w_t = \frac{1}{E_t}\left(\sum_{i=1}^{M}\frac{1}{E_i}\right)^{-1}, \tag{8}$$

$$w_t = \frac{1}{E_t+\delta}\left(\sum_{i=1}^{M}\frac{1}{E_i+\delta}\right)^{-1} \tag{9}$$

and

$$w_t = e^{-E_t}\left(\sum_{i=1}^{M}e^{-E_i}\right)^{-1}, \tag{10}$$

where δ is a small positive value to handle the situation where $E_i = 0$ in the denominator of Equation (8). The numerical results indicate that there is not much difference in the classification performance between weights (9) and (10). In this article, we adopt (10) for the weight assignments.

Although there are a wealth of empirical evidences illustrating the advantages of ensemble classifier over a single classifier in literature, there are very few theoretical studies. For a general ensemble learning problem, let θ and θ_i represent the expectation of misclassification error rates for the ensemble classifier and the ith base classifier. Let η be the variance of the base classifiers. Then it is known that

$$\theta = \sum_i w_i\theta_i - \eta \le \sum_i w_i\theta_i. \tag{11}$$

It demonstrates that the expected misclassification error rate of the ensemble classifier is less than the weighted average of expected error of the base classifiers, and becomes smaller when the diversity of the base classifiers increases. The proposed ensemble algorithm for classification in this paper satisfies two principals of "accurate" and "diverse" in the ensemble learning. The accuracy of the base classifiers h_i results from that the MCD framework is able to provide a good sparse precision estimate in the high-dimensional settings. The diversity of the base classifiers h_i is fulfilled by a random selection scheme of the variable orderings in the MCD approach.

At the end of this section, we will remark that the computational complexity of the proposed method is $O(Mp^3)$. This is higher than $O(p^3)$ for some existing methods which improve the classifier's ability by proposing a novel precision estimate. In practice, we suggest to choose a relatively large value of M to ensure the classification accuracy if the computational resources are available. Otherwise, a moderate value of M is recommended to balance the accuracy and computation efficiency for the proposed method. In addition, the proposed classifier solves M precision estimates based on independently chosen variable orderings, which enables us to apply the parallel computing techniques. This means we can simultaneously solve precision estimates on multiple processors separately, which will greatly reduce the computational time.

Algorithm 2: Variable importance measures for the proposed classifier.

Input: The proposed classifier $H(\cdot)$, validation set W and corresponding class label vector y_W.
Step 1. Compute the misclassification error E_{PROP} of $H(\cdot)$.
Step 2. **for** $j = 1, \ldots, p$ **do**
  Generate W_{perm} by randomly permuting values of variable j in data W;
  Compute the misclassification error E_{perm} by comparing y_W and the predicted labels $H(W_{perm})$;
  Compute the variable importance $VI_j = E_{perm} - E_{PROP}$;
end
Step 3. Sort variables by descending VI_j.

2.4. Variable importance measures

The interpretability of ensemble learning models is as important as their prediction accuracy. In this section, we suggest a procedure to interpret our proposed method through measuring the variable importance via permuting the variable's values, which is enlightened by the random forest (Breiman, 2001). Specifically, we evaluate the importance of a variable by computing an increase in the misclassification error of the proposed classifier after permuting the variable. A variable is important if the misclassification error increases due to shuffling its values, since the proposed classifier relies on the variable for the discrimination. On the contrary, a variable is not important if the misclassification error unchanged after shuffling its values, as in this case the variable contributes little to the discrimination. The procedure of measuring the variable importance of the proposed classifier is summarized in Algorithm 2. Here the proposed classifier $H(\cdot)$ is already constructed from the training data. In practice, to implement such procedure, a validation set W is needed which is not used to build the proposed classifier.

3. Theoretical properties

In this section we establish the theoretical properties of the proposed classifier. For an observation $\boldsymbol{x}$ from one of the two classes G_1 and G_2, we define the misclassification error rate of a classification rule as the average of the probabilities for making two types of mistakes: classifying $\boldsymbol{x}$ to class G_1 when $\boldsymbol{x}$ is actually from G_2 and classifying $\boldsymbol{x}$ to class G_2 when $\boldsymbol{x}$ is from G_1.

We firstly focus on the property of an individual LDA classifier for problem (1). For convenience, we assume $\pi_0 = \pi_1$ throughout this section. Denote by $\boldsymbol{R}$ the Bayes classification error rate. From the properties of the normal distribution, we have

$$\boldsymbol{R} = \Phi(-\Delta_p/2), \quad \Delta_p = \sqrt{\boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{\Omega}\boldsymbol{\delta}},$$

where $\boldsymbol{\delta} = \boldsymbol{\mu}_0 - \boldsymbol{\mu}_1$ and $\Phi(\cdot)$ represents the cumulative distribution function of a standard normal variable. Given the training set, the conditional misclassification error rate of an individual LDA classifier constructed via MCD from Algorithm 1 is

$$\boldsymbol{R}_n = \frac{1}{2}\sum_{k=0}^{1}\Phi\left(\frac{(-1)^{k+1}\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\boldsymbol{\Omega}}\left(\boldsymbol{\mu}_k - \hat{\boldsymbol{\mu}}_k\right) - \hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\boldsymbol{\Omega}}\hat{\boldsymbol{\delta}}/2}{\sqrt{\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\boldsymbol{\Omega}}\boldsymbol{\Sigma}\hat{\boldsymbol{\Omega}}\hat{\boldsymbol{\delta}}}}\right),$$

where $\hat{\boldsymbol{\delta}} = \hat{\boldsymbol{\mu}}_0 - \hat{\boldsymbol{\mu}}_1$ and $\hat{\boldsymbol{\Omega}}$ denotes the Cholesky-based estimate of the precision matrix. To establish the theoretical results, we use the following definition from Shao et al. (2011).

Definition 1. Let T be a classification rule with conditional misclassification rate $\boldsymbol{R}_T$.

(i) T is asymptotically optimal if $\boldsymbol{R}_T/\boldsymbol{R} \xrightarrow{P} 1$.
(ii) T is asymptotically sub-optimal if $\boldsymbol{R}_T - \boldsymbol{R} \xrightarrow{P} 0$.

Here $\xrightarrow{p}$ represents convergence in probability. If $\lim_{n\to\infty}\boldsymbol{R} > 0$, the asymptotic sub-optimality is equivalent to the asymptotic optimality. In order to facilitate the expression of theoretical results, we introduce some notation. Let $\boldsymbol{\Omega} = \boldsymbol{T}^{\mathrm{T}}\boldsymbol{D}^{-1}\boldsymbol{T}$ be the MCD of the true precision matrix. Define set B as the collection of the nonzero elements in the lower triangular part of the matrix $\boldsymbol{T}$, and denote by the symbol φ the cardinality of B. Additionally, let the singular values of a matrix $\boldsymbol{A}$ be $sv_1(\boldsymbol{A}) \geq sv_2(\boldsymbol{A}) \geq \cdots \geq sv_p(\boldsymbol{A})$ in decreasing order. Besides, we assume the following regularity conditions.

(C1) The singular values of $\boldsymbol{\Omega}$ are bounded. That is, there exists a constant h_1 such that $0 < 1/h_1 < sv_p(\boldsymbol{\Omega}) \leq sv_1(\boldsymbol{\Omega}) < h_1 < \infty$.
(C2) The tuning parameter λ_j in (7) satisfies $\sum_{j=2}^{p}\lambda_j = O(\sqrt{\ln(p)/n})$.
(C3) For $\boldsymbol{\delta} = (\delta_j)_{1\leq j\leq p}$, there exists a constant h_2 such that $0 < 1/h_2 \leq \max_{1\leq j\leq p}\delta_j^2 \leq h_2 < \infty$.
(C4) $(\varphi + p)\ln(p)/n = o(1)$.

The conditions (C1) and (C2) are used in the literature to derive the consistency property of the Cholesky-based estimate of precision matrix, which is then used to construct the optimality property of the LDA rule. The conditions (C1) and (C3) indicate $\Delta_p \nrightarrow 0$, hence guaranteeing that the classes are separable. Now we present the theoretical results with detailed proofs deferred in the Appendix.

Theorem 1. *Suppose conditions (C1)-(C4) hold and $b_n = \sqrt{(\varphi + p)\ln(p)/n}$, then we have*

(i) *The conditional misclassification error rate of the Cholesky-based LDA rule is*

$$\boldsymbol{R}_n = \Phi(-[1 + O_p(b_n)]\Delta_p/2).$$

(ii) *If Δ_p is bounded, then the Cholesky-based LDA rule is asymptotically optimal and*

$$\frac{\boldsymbol{R}_n}{\boldsymbol{R}} - 1 = O_p(b_n).$$

(iii) *If $\Delta_p \to \infty$, then the Cholesky-based LDA rule is asymptotically sub-optimal.*
(iv) *If $\Delta_p \to \infty$ and $b_n\Delta_p^2 \to 0$, then the Cholesky-based LDA rule is asymptotically optimal.*

Theorem 1 establishes the convergency properties of the Cholesky-based LDA rule and demonstrates that the base classifier is able to asymptotically perform as well as the Bayes classifier under conditions (C1)-(C4). When Δ_p is bounded, the convergency rate b_n is guaranteed to converge to 0 if $p = O(n^\rho)$ with $0 \le \rho < 1/2$ since φ is at most $O(p^2)$. Moreover in the case of $\Delta_p \to \infty$, the condition $b_n\Delta_p^2 \to 0$ is stronger than $b_n \to 0$, which can be satisfied when $p = O(n^\rho)$ with $0 \le \rho < 1/4$ since $\Delta_p^2 = O(p)$ under conditions (C1) and (C3).

4. Simulation

In this section, we evaluate the finite performance of the proposed algorithm (PROP) by comparing the SAMPLE, GLASSO, MDA and EDDA methods. The SAMPLE refers to the LDA rule with its precision matrix replaced by the inverse of the sample covariance matrix. When it is singular in high dimensions, the generalized inverse is used. The Glasso refers to the LDA but where the graphical lasso estimate is used for the precision matrix in the rule. The tuning parameter is chosen by the BIC according to Yuan and Lin (2007). The MDA (Hastie and Tibshirani, 1996) extends the LDA by using a mixture norm distribution to model data in each class. The EDDA (Bensmail and Celeux, 1996) extends the LDA by estimating the covariance matrices for each class via the eigenvalue decomposition.

We consider the number of variables $p = 50, 100, 150$ as well as 30 samples in class G_1 and 30 samples in class G_2, that is, the size of the training set $n = 60$ with $n_0 = n_1 = 30$. The testing set contains 50 samples in class G_1 and 50 samples in class G_2, which is used to evaluate the prediction performance of each method. For the base classifiers of PROP, we consider $M = 50$ randomly generated variable orderings.

The data are drawn from two p-dimensional normal distributions with the same precision matrix $\boldsymbol{\Omega}$ and different mean vectors $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$. We set $\boldsymbol{\mu}_0 = \mathbf{0}$, and $\boldsymbol{\mu}_1 = (u_1, \ldots, u_p)^{\mathrm{T}}$ where each u_i is from uniform distribution U(0,1) independently. Five scenarios of the true precision matrix $\boldsymbol{\Omega} = (\omega_{ij})_{p\times p}$ are considered.

- Scenario 1. $\boldsymbol{\Omega}_1$ is a tridiagonal matrix with $\omega_{ij} = 1_{\{i=j\}} + 0.4_{\{|i-j|=1\}}$, $i, j = 1, \ldots, p$.
- Scenario 2. $\boldsymbol{\Omega}_2$ is generated by randomly permuting rows and corresponding columns of $\boldsymbol{\Omega}_1$.
- Scenario 3. $\boldsymbol{\Omega}_3$ is a loose banded matrix with $\omega_{ij} = 1_{\{i=j\}} + 0.4_{\{|i-j|=p/5\}}$. Compared to $\boldsymbol{\Omega}_1$, the nonzero entries of $\boldsymbol{\Omega}_3$ are less close to the diagonal.
- Scenario 4. The variances of variables are 1 and 1.25. The covariances of adjacent variables are -0.5.

$$\boldsymbol{\Omega}_4 = \begin{pmatrix} 1 & -0.5 & 0 & \ldots & 0 & 0 & 0 \\ -0.5 & 1.25 & -0.5 & \ldots & 0 & 0 & 0 \\ 0 & -0.5 & 1.25 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1.25 & -0.5 & 0 \\ 0 & 0 & 0 & \ldots & -0.5 & 1.25 & -0.5 \\ 0 & 0 & 0 & \ldots & 0 & -0.5 & 1 \end{pmatrix}.$$

- Scenario 5. $\boldsymbol{\Omega}_5 = \boldsymbol{B} + \alpha \boldsymbol{I}$ is a sparse matrix with no specific sparse structure. Nonzero entries have random positions and quantities. Here $\boldsymbol{B} = (b_{ij})_{p\times p}$ with $b_{ii} = 0, b_{ij} = b_{ji} = \mathrm{Ber}(1, 0.15) \times \mathrm{U}(-1, 1)$, where Ber represents Bernoulli distribution. About 15% of the non-diagonal entries are nonzeros and each non-diagonal entry is generated independently. We set $\alpha = \max(-\lambda_{min}(\boldsymbol{B}), 0) + 0.1$ to ensure the positive definiteness of $\boldsymbol{\Omega}_5$.

To evaluate the performance of afore mentioned methods, we use the criteria of the misclassification error rate and the area under curve (AUC). The AUC is commonly used for examining the ability of a binary classifier. It measures the area under the receiver operating characteristic (ROC) curve, which is a graph created by plotting the true positive rate against the false positive rate at various threshold settings. Table 1 presents the averages and corresponding standard errors (in parentheses) of misclassification error rates in percentage on testing set based on 50 replications, as well as the averaged AUC for each method. Dashed lines in the tables represent the corresponding values not available possibly due to the matrix singularity in high dimensions. The standard errors of AUC are omitted since they are all equal to zero except a few scenarios for SAMPLE in which they are a little bit larger. We observe that the PROP overall produces the best classification performance in almost all the settings. More precisely, the PROP is better than or at least comparable with GLASSO, and much better than the MDA, EDDA and SAMPLE in terms of error rate. Since the MDA assumes data from a mixture normal distribution, it gives inferior performance than the PROP as expected. The EDDA encounters computational issues in high dimensions since its construction is based on the sample covariance. When the number of variables p increases from 50 to 150, the misclassification error rates of SAMPLE rise sharply, even over 50% in some cases. While the error rates of PROP are relatively stable, which shows the superiority of PROP in tackling the high-dimensional binary classification problem.

Furthermore, to investigate the impact of the values of M on the classification performance of the proposed method, we report in Fig. 1 the misclassification error rates in percentage for five scenarios of precision matrices with respect to different values of $M = 5, 10, 20, 30, 40$ and 50. The solid line, dashed line and dotted line represent three situations where $p = 50, 100$ and 150, respectively. We observe that almost all the lines significantly decrease in the range of $M = (5, 20)$, and

Table 1
The averaged misclassification error rates in percentage and AUC with their standard errors (in parenthesis) of each method for simulated data.

2*Scenario	p	PROP		GLASSO		MDA		EDDA		SAMPLE	
		error rate	AUC	error rate	AUC	error rate	AUC	error rate	AUC	error rate	AUC
	50	**1.62 (0.23)**	1.00	2.62 (0.30)	1.00	10.60 (0.48)	0.89	6.44 (0.50)	0.94	17.62 (1.08)	0.87
	100	**0.24 (0.09)**	1.00	0.34 (0.09)	1.00	3.96 (0.35)	0.96	- (-)	-	48.96 (2.00)	0.48
	150	**0.20 (0.06)**	1.00	0.42 (0.10)	1.00	1.52 (0.20)	0.98	- (-)	-	48.90 (1.79)	0.51
2	50	**1.90 (0.24)**	0.99	2.66 (0.28)	1.00	7.66 (0.50)	0.92	4.34 (0.41)	0.96	18.46 (1.18)	0.85
	100	**0.28 (0.09)**	1.00	0.46 (0.10)	1.00	2.42 (0.22)	0.97	- (-)	-	51.12 (1.89)	0.47
	150	**0.08 (0.04)**	1.00	**0.08 (0.04)**	1.00	0.90 (0.15)	0.98	- (-)	-	49.76 (1.69)	0.51
3	50	**2.16 (0.34)**	0.99	3.06 (0.41)	1.00	10.28 (0.64)	0.90	6.78 (0.55)	0.93	16.46 (0.99)	0.90
	100	**0.32 (0.10)**	1.00	0.56 (0.12)	1.00	3.92 (0.39)	0.96	- (-)	-	50.06 (1.85)	0.53
	150	**0.10 (0.04)**	1.00	0.42 (0.09)	1.00	1.64 (0.24)	0.98	- (-)	-	52.78 (1.82)	0.47
4	50	44.60 (0.80)	0.51	44.74 (0.87)	0.52	44.50 (1.01)	0.56	**43.34 (0.97)**	0.57	47.62 (0.76)	0.53
	100	**41.82 (0.87)**	0.61	45.00 (0.92)	0.57	42.56 (0.69)	0.58	- (-)	-	50.06 (0.68)	0.51
	150	**42.28 (0.79)**	0.70	43.76 (0.63)	0.67	43.04 (0.76)	0.57	- (-)	-	49.96 (0.85)	0.50
5	50	**5.16 (0.48)**	0.96	5.34 (0.50)	0.95	7.68 (0.61)	0.92	6.06 (0.57)	0.94	21.52 (1.02)	0.83
	100	**2.62 (0.29)**	0.99	3.94 (0.40)	1.00	10.74 (0.72)	0.90	- (-)	-	48.90 (1.66)	0.50
	150	**1.66 (0.18)**	0.99	1.68 (0.22)	1.00	6.58 (0.78)	0.93	- (-)	-	51.12 (1.48)	0.47

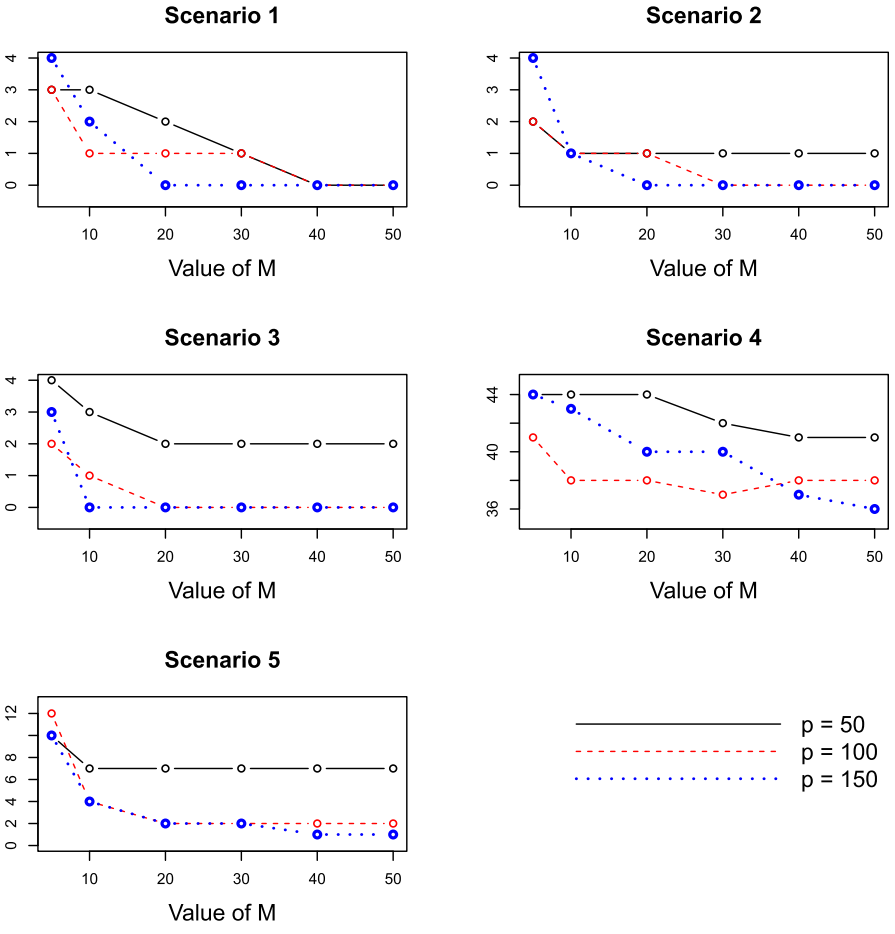

Fig. 1. Plot of misclassification error rates in percentage of the proposed classifier against the values of M. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

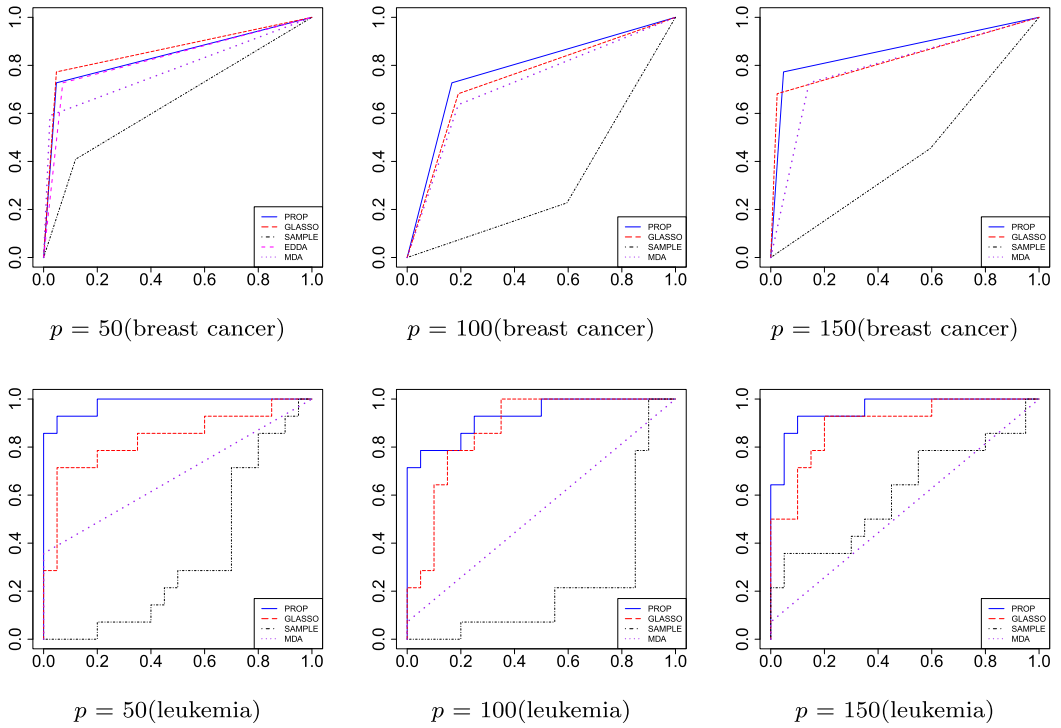

Fig. 2. ROC curves of each method for the real data.

Table 2
The averaged misclassification error rates in percentage and AUC with their standard errors (in parenthesis) of each method for the real data.

dataset	p	PROP		GLASSO		MDA		EDDA		SAMPLE	
		error rate	AUC	error rate	AUC	error rate	AUC	error rate	AUC	error rate	AUC
breast cancer	50	13.6 (0.46)	0.83 (0.003)	14.7 (0.45)	0.82 (0.002)	14.4 (0.48)	0.82 (0.007)	13.6 (0.41)	0.85 (0.005)	31.0 (1.05)	0.76 (0.021)
	100	14.2 (0.41)	0.81 (0.003)	15.2 (0.47)	0.80 (0.001)	14.7 (0.59)	0.82 (0.007)	- (-)	- (-)	50.8 (1.46)	0.53 (0.029)
	150	14.7 (0.52)	0.90 (0.002)	14.8 (0.45)	0.91 (0.002)	14.8 (0.47)	0.81 (0.006)	- (-)	- (-)	52.4 (1.06)	0.50 (0.037)
leukemia	50	5.88	0.93	26.5	0.74	26.4	0.68	- (-)	- (-)	58.8	0.39
	100	11.8	0.86	17.7	0.82	38.2	0.54	- (-)	- (-)	61.8	0.34
	150	8.82	0.90	26.5	0.70	38.3	0.54	- (-)	- (-)	41.2	0.61

become stable over other values of M, which imply that the performance of the proposed classifier tends to be consistent when M exceeds 20 in the simulation study.

5. Application

In this section, we apply the proposed algorithm to analyze the medical data, including a breast cancer dataset and a leukemia dataset. To examine the performance of the proposed algorithm, we compare the GLASSO, MDA, EDDA and SAMPLE methods.

The breast cancer dataset from Glaab et al. (2012) contains 128 samples including 84 luminal samples and 44 non-luminal samples with 47293 gene expressions. We randomly choose half of the samples in each class to build the training set, and the rest are used as the testing set. The analysis results for breast cancer data displayed in Fig. 2 and Table 2 are based on 50 such independent random splits. The leukemia dataset from Golub et al. (1999) has 72 samples along with 7129 genes. All the samples are taken from acute leukemia patients, who are either lymphoblastic leukemia (ALL) or acute myelogenous leukemia (AML). The dataset is provided with 38 training samples (27 ALL patients and 11 AML patients) and 34 testing samples (20 ALL patients and 14 AML patients).

Before analysis, we preprocess the data as follows. First, the leukemia data are scaled by 1/100 for the analysis reason. Then the variable screening procedure is conducted on two datasets via the two sample t-test to select the most significant variables that are more informative for the discrimination (Rothman et al., 2009; Xue et al., 2012; Li et al., 2021). Specifically, for each variable, a t-test is performed against the two classes of the training data such that the variables with large absolute values of test statistics are considered as the significant variables. The top 50, 100 and 150 significant variables are used for the subsequent data classification.

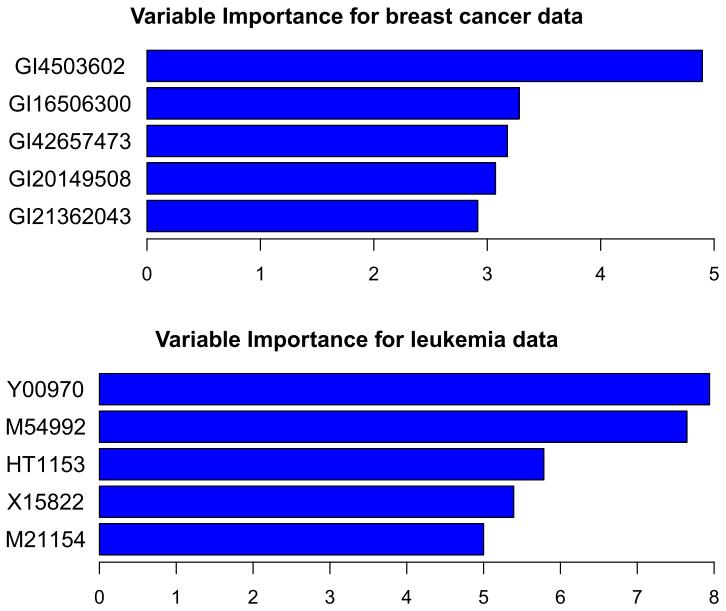

Fig. 3. The variable importance ranking obtained from the proposed classifier for the real data.

Fig. 2 depicts the ROC curves of compared methods. For the breast cancer dataset, it randomly selects a split among 50 replications and plots the ROC curves for each method. Table 2 reports the averaged misclassification error rates in percentage and corresponding standard errors (in parentheses) as well as AUC values for each method. From these results, it is seen that for the breast cancer data, the PROP slightly outperforms GLASSO as well as MDA, and is much better than SAMPLE. As p increases from 50 to 150, the performance of SAMPLE becomes worse while others remain stable. For the leukemia dataset, the PROP shows a great superiority over other methods in each setting of p. The EDDA performs very well in predicting the class labels when $p = 50$, but it fails in high dimensions when p is larger than the sample size.

Additionally, to better interpret the model, we evaluate the variable importance for two data sets by Algorithm 2. Since the purpose here is to examine the ability of Algorithm 2 in measuring the variable importance, we simply take the testing set as the validate set for a convenience of illustration. Fig. 3 shows the measures of the variable importance from the proposed method with the top five most important variables when $p = 150$. The labels on the vertical axis are the gene names from the data. We see that genes "GI4503602" and "Y00970" are the most informative variables in the classification in the study for two data respectively.

6. Discussion

In this work, we have proposed an ensemble classification algorithm based on the MCD estimate of matrices. The proposed algorithm has two innovations. First, it incorporates the MCD technique to accommodate the traditional LDA for high-dimensional situations. Second, it utilizes the capacity of MCD, which can provide multiple precision estimates, to construct a set of base LDA classifiers, such that the ensemble learning technique can be applied to accurately predict class labels.

The proposed ensemble classifier can be easily extended to solve the multi-group classification problems, since the LDA is able to classify multiple labels by comparing the posterior probabilities of a new observation belonging to each class given the training set. To accommodate the multi-group classification, the weighted voting is modified accordingly as follows. Let $h_i, i = 1, \ldots, M$, be the base classifier in the N-dimensional classification problem, where the class labels are denoted by $\{c_1, \ldots, c_N\}$. Let $(h_i^1(\boldsymbol{x}), h_i^2(\boldsymbol{x}), \ldots, h_i^N(\boldsymbol{x}))^{\mathrm{T}}$ be the N-dimensional output vector of the classifier h_i for a new sample $\boldsymbol{x}$. That is, if the classifier h_i predicts the class label of sample $\boldsymbol{x}$ to be c_j, then $h_i^j(\boldsymbol{x}) = 1$ and $h_i^k(\boldsymbol{x}) = 0$ for $k \neq j$. Consequently, the ensemble classifier $H(\boldsymbol{x}) = c_{j^\star}$, where $j^\star = \arg\max_j \sum_{i=1}^M w_i h_i^j(\boldsymbol{x})$ with w_i as the weight of classifier h_i.

In the proposed algorithm, we randomly generate and use the variable orderings, which may not ensure the diversity of base classifiers. An interesting research direction is to investigate an efficient mechanism for choosing appropriate variable orderings, which can lead to a set of more representative base LDA classifiers.

Acknowledgements

The authors are grateful to the AE and the reviewers for their insightful comments that have significantly improved the paper. Zhenguo Gao's research is supported by National Natural Science Foundation of China (Grant No. NSFC-12001365)

and the Research Startup Foundation of Shanghai Jiao Tong University (Grant No. WF220407116). Xiaoning Kang's research is supported by Natural Science Foundation of Liaoning Province (Grant No. 2022-MS-179), Department of Education of Liaoning Province (Grant No. LJKMZ20221565), and National Natural Science Foundation of China (Grant No. NSFC-72232001).

Appendix A

A.1. Proof of Theorem 1

To prove Theorem 1, we need Lemma 1 from Kang and Deng (2020) and Lemma 2 from Shao et al. (2011). We present two Lemmas here for completeness, but omit their proof to save space.

Lemma 1. *Assume data are from normal distribution $N(0, \mathbf{\Omega}^{-1})$. Under conditions (C1) and (C2), we have*

$$\|\hat{\mathbf{\Omega}}_k - \mathbf{\Omega}\|_F = O_p(\sqrt{\frac{(\varphi + p)\ln(p)}{n}}),\ k = 1, \ldots, M.$$

Lemma 2. *Let ξ_n and τ_n be two sequences of positive numbers such that $\xi_n \to \infty$ and $\tau_n \to 0$ when $n \to \infty$. If $\lim_{n\to\infty} \tau_n \xi_n = \gamma$, where γ may be 0, positive or ∞, then we have*

$$\lim_{n\to\infty} \frac{\Phi\left(-\sqrt{\xi_n}(1 - \tau_n)\right)}{\Phi\left(-\sqrt{\xi_n}\right)} = e^{\gamma}.$$

Lemma 1 from Kang and Deng (2020) shows the convergence rate of the Cholesky-based estimate of precision matrix for any variable ordering. Lemma 2 from Shao et al. (2011) demonstrates the convergence property of $\Phi(\cdot)$, which will be used to prove part (iv) of Theorem 1. Now we prove Theorem 1.

Proof. (i) The conditional misclassification rate $\boldsymbol{R}_n$ is

$$\frac{1}{2}\sum_{k=0}^{1} \Phi\left(\frac{(-1)^{k+1}\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\mathbf{\Omega}}\left(\boldsymbol{\mu}_k - \hat{\boldsymbol{\mu}}_k\right) - \hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\mathbf{\Omega}}\hat{\boldsymbol{\delta}}/2}{\sqrt{\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\mathbf{\Omega}}\mathbf{\Sigma}\hat{\mathbf{\Omega}}\hat{\boldsymbol{\delta}}}}\right). \tag{12}$$

By Lemma 1, we have $\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\mathbf{\Omega}}\mathbf{\Sigma}\hat{\mathbf{\Omega}}\hat{\boldsymbol{\delta}} = \hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\mathbf{\Omega}}\hat{\boldsymbol{\delta}}[1 + O_P(b_n)] = \hat{\boldsymbol{\delta}}^{\mathrm{T}}\mathbf{\Omega}\hat{\boldsymbol{\delta}}[1 + O_P(b_n)]$. We decompose the term $\hat{\boldsymbol{\delta}}^{\mathrm{T}}\mathbf{\Omega}\hat{\boldsymbol{\delta}}$ as

$$\hat{\boldsymbol{\delta}}^{\mathrm{T}}\mathbf{\Omega}\hat{\boldsymbol{\delta}} = \boldsymbol{\delta}^{\mathrm{T}}\mathbf{\Omega}\boldsymbol{\delta} + (\hat{\boldsymbol{\delta}} - \boldsymbol{\delta})^{\mathrm{T}}\mathbf{\Omega}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}) + 2\boldsymbol{\delta}^{\mathrm{T}}\mathbf{\Omega}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}).$$

Now we bound these three terms separately. The first term $\boldsymbol{\delta}^{\mathrm{T}}\mathbf{\Omega}\boldsymbol{\delta} = \Delta_p^2$. By the spectral decomposition of $\mathbf{\Omega}$, the second term $(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta})^{\mathrm{T}}\mathbf{\Omega}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}) = (\hat{\boldsymbol{\delta}} - \boldsymbol{\delta})^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\mathbf{\Lambda}\mathbf{Q}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta})$, where $\mathbf{Q}$ is an orthogonal matrix and $\mathbf{\Lambda}$ is a diagonal matrix with eigenvalues of $\mathbf{\Omega}$ on its diagonal. Accordingly, from condition (C1) we have

$$\begin{aligned}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta})^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\mathbf{\Lambda}\mathbf{Q}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}) &\in (\frac{\|\mathbf{Q}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta})\|_2^2}{h}, h\|\mathbf{Q}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}\|_2^2) \\ &= (\frac{\|\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}\|_2^2}{h}, h\|\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}\|_2^2).\end{aligned} \tag{13}$$

Based on the properties of the multivariate normal distribution, it is easy to derive $\hat{\boldsymbol{\delta}} - \boldsymbol{\delta} \sim N(\mathbf{0}, (\frac{1}{n_0} + \frac{1}{n_1})\mathbf{\Sigma})$. Therefore we have

$$E\|\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}\|_2^2 = (\frac{1}{n_0} + \frac{1}{n_1})\mathrm{tr}(\mathbf{\Sigma}) \in (\frac{p}{hn_0} + \frac{p}{hn_1}, \frac{ph}{n_0} + \frac{ph}{n_1}). \tag{14}$$

From (13) and (14), we obtain that $E[(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta})^{\mathrm{T}}\mathbf{\Omega}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta})] = O(p/n)$. The third term is bounded through the Cauchy-Schwarz inequality, resulting in $E[\boldsymbol{\delta}^{\mathrm{T}}\mathbf{\Omega}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta})]^2 \le \Delta_p^2 E[(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta})^{\mathrm{T}}\mathbf{\Omega}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta})]$, which indicates that $\boldsymbol{\delta}^{\mathrm{T}}\mathbf{\Omega}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}) = \Delta_p O_P(\sqrt{p/n})$. Consequently, we have

$$\begin{aligned}\hat{\boldsymbol{\delta}}^{\mathrm{T}}\mathbf{\Omega}\hat{\boldsymbol{\delta}} &= \Delta_p^2 + O_P\left(\frac{p}{n}\right) + O_P\left(\frac{\sqrt{p}\Delta_p}{\sqrt{n}}\right) \\ &= \Delta_p^2\left[1 + O_P\left(\frac{\sqrt{p}}{\sqrt{n}\Delta_p}\right) + O_P\left(\frac{p}{n\Delta_p^2}\right)\right] \\ &= \Delta_p^2\left[1 + O_P(\frac{\sqrt{p}}{\sqrt{n}\Delta_p})\right].\end{aligned}$$

Now we consider

$$\frac{\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\boldsymbol{\Omega}}\left(\hat{\boldsymbol{\mu}}_0-\boldsymbol{\mu}_0\right)-\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\boldsymbol{\Omega}}\hat{\boldsymbol{\delta}}/2}{\sqrt{\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\boldsymbol{\Omega}}\boldsymbol{\Sigma}\hat{\boldsymbol{\Omega}}\hat{\boldsymbol{\delta}}}}=-\frac{\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\boldsymbol{\Omega}}\hat{\boldsymbol{\delta}}}{2\sqrt{\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\boldsymbol{\Omega}}\boldsymbol{\Sigma}\hat{\boldsymbol{\Omega}}\hat{\boldsymbol{\delta}}}}+\frac{\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\boldsymbol{\Omega}}\left(\hat{\boldsymbol{\mu}}_0-\boldsymbol{\mu}_0\right)}{\sqrt{\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\boldsymbol{\Omega}}\boldsymbol{\Sigma}\hat{\boldsymbol{\Omega}}\hat{\boldsymbol{\delta}}}}. \tag{15}$$

For the first term of Equation (15),

$$\begin{aligned}-\frac{\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\boldsymbol{\Omega}}\hat{\boldsymbol{\delta}}}{2\sqrt{\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\boldsymbol{\Omega}}\boldsymbol{\Sigma}\hat{\boldsymbol{\Omega}}\hat{\boldsymbol{\delta}}}}=-\frac{\sqrt{\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\boldsymbol{\Omega}}\hat{\boldsymbol{\delta}}}}{2\sqrt{1+O_P\left(b_n\right)}}&=-\frac{\Delta_p\sqrt{1+O_P\left(\frac{\sqrt{p}}{\sqrt{n}\Delta_p}\right)}}{2\sqrt{1+O_P\left(b_n\right)}}\\&=-\frac{\Delta_p}{2}\left[1+O_P\left(b_n\right)+O_P\left(\frac{\sqrt{p}}{\sqrt{n}\Delta_p}\right)\right]\\&=-\frac{\Delta_p}{2}\left[1+O_P\left(b_n\right)\right],\end{aligned}$$

where the last equality is from $\sqrt{p}/(b_n\sqrt{n}\Delta_p)=\sqrt{p}/(\sqrt{(s+p)\ln p}\Delta_p)=O(1)$ because $\Delta_p\neq 0$ under condition (C3). For the second term of Equation (15),

$$\frac{\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\boldsymbol{\Omega}}\left(\hat{\boldsymbol{\mu}}_0-\boldsymbol{\mu}_0\right)}{\sqrt{\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\boldsymbol{\Omega}}\boldsymbol{\Sigma}\hat{\boldsymbol{\Omega}}\hat{\boldsymbol{\delta}}}}=\frac{\sqrt{\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\boldsymbol{\Omega}}\hat{\boldsymbol{\delta}}}\,O_P(\sqrt{p/n})}{\sqrt{\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\boldsymbol{\Omega}}\hat{\boldsymbol{\delta}}}\sqrt{1+O_P(b_n)}}=O_P\left(\sqrt{\frac{p}{n}}\right),$$

where the first equality applies the Cauchy-Schwarz inequality for the numerator. As a result, we have

$$\begin{aligned}\frac{\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\boldsymbol{\Omega}}\left(\hat{\boldsymbol{\mu}}_0-\boldsymbol{\mu}_0\right)-\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\boldsymbol{\Omega}}\hat{\boldsymbol{\delta}}/2}{\sqrt{\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\boldsymbol{\Omega}}\boldsymbol{\Sigma}\hat{\boldsymbol{\Omega}}\hat{\boldsymbol{\delta}}}}&=-\frac{\Delta_p}{2}\left[1+O_P\left(b_n\right)\right]+O_P\left(\sqrt{\frac{p}{n}}\right)\\&=-\frac{\Delta_p}{2}\left[1+O_P\left(b_n\right)+O_P\left(\frac{\sqrt{p}}{\sqrt{n}\Delta_p}\right)\right]\\&=-\frac{\Delta_p}{2}\left[1+O_P(b_n)\right].\end{aligned}$$

Similarly, we can show that

$$\frac{\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\boldsymbol{\Omega}}\left(\boldsymbol{\mu}_1-\hat{\boldsymbol{\mu}}_1\right)-\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\boldsymbol{\Omega}}\hat{\boldsymbol{\delta}}/2}{\sqrt{\hat{\boldsymbol{\delta}}^{\mathrm{T}}\hat{\boldsymbol{\Omega}}\boldsymbol{\Sigma}\hat{\boldsymbol{\Omega}}\hat{\boldsymbol{\delta}}}}=-\frac{\Delta_p}{2}\left[1+O_P(b_n)\right].$$

These results together with formula (12) imply the result in (i).

(ii) Let ϕ be the density of Φ. By the result in (i), we have

$$\boldsymbol{R}_n-\boldsymbol{R}=\phi(\omega_n)O_P(b_n),$$

where ω_n is between $-\Delta_p/2$ and $-[1+O_P(b_n)]\Delta_p/2$. Since $\phi(\omega_n)$ is bounded by a constant, the result follows from the fact that $\boldsymbol{R}$ is bounded away from 0 when Δ_p is bounded.

(iii) When $\Delta_p\to\infty$, $\boldsymbol{R}\to 0$, and by the result in (i), $\boldsymbol{R}_n\xrightarrow{P}0$.

(iv) If $\Delta_p\to\infty$, then by Lemma 2 and the condition $b_n\Delta_p^2\to 0$, we conclude that $\boldsymbol{R}_n/\boldsymbol{R}\xrightarrow{P}1$. □

A.2. Derivation of formula (11)

Suppose the data and its distribution are $x\sim p(x)$, $y(x)$ is the class label of the sample x. For M base classifiers $h_i(x)$, the ensemble classifier is

$$H(x)=\sum_{i=1}^{M}w_ih_i(x),$$

where $w_i\geq 0$ and $\sum_{i=1}^{M}w_i=1$. The expectation of misclassification error of the ith base classifier and the ensemble classifier are

$$\theta_i = \int_x p(x)\big(h_i(x) - y(x)\big)^2 dx \quad \text{and} \quad \theta = \int_x p(x)\big(H(x) - y(x)\big)^2 dx.$$

Now consider

$$\begin{aligned}
\sum_{i=1}^{M} w_i\theta_i &= \sum_{i=1}^{M} w_i \int_x p(x)\Big(h_i(x) - y(x)\Big)^2 dx \\
&= \sum_{i=1}^{M} w_i \int_x p(x)\Big(h_i(x) - H(x) + H(x) - y(x)\Big)^2 dx \\
&= \sum_{i=1}^{M} w_i \int_x p(x)\Big(\Big[h_i(x) - H(x)\Big]^2 + \Big[H(x) - y(x)\Big]^2 \\
&\quad + 2\Big[h_i(x) - H(x)\Big]\Big[H(x) - y(x)\Big]\Big)dx \\
&= \sum_{i=1}^{M} w_i \int_x p(x)\Big(\Big[h_i(x) - H(x)\Big]^2 dx + \sum_{i=1}^{M} w_i \int_x p(x)\Big[H(x) - y(x)\Big]^2 dx \\
&\quad + 2\sum_{i=1}^{M} w_i \int_x p(x)\Big[h_i(x) - H(x)\Big]\Big[H(x) - y(x)\Big]dx. \\
&= \eta + \sum_{i=1}^{M} w_i\theta + 0.
\end{aligned}$$

That is, $\sum_{i=1}^{M} w_i\theta_i = \theta + \eta$. Consequently, the expectation of the misclassification error rate of ensemble classifier is

$$\theta = \sum_{i=1}^{M} w_i\theta_i - \eta \le \sum_{i=1}^{M} w_i\theta_i.$$

References

Bensmail, H., Celeux, G., 1996. Regularized gaussian discriminant analysis through eigenvalue decomposition. J. Am. Stat. Assoc. 91, 1743–1748.
Breiman, L., 1996. Bagging predictors. Mach. Learn. 24, 123–140.
Breiman, L., 2001. Random forests. Mach. Learn. 45, 5–32.
Cai, J., Huang, X., 2018. Modified sparse linear-discriminant analysis via nonconvex penalties. IEEE Trans. Neural Netw. Learn. Syst. 29, 4957–4966.
Cai, T., Liu, W., 2011. A direct estimation approach to sparse linear discriminant analysis. J. Am. Stat. Assoc. 106, 1566–1577.
Chang, C., Tsay, R.S., 2010. Estimation of covariance matrix via the sparse cholesky factor with lasso. J. Stat. Plan. Inference 140, 3858–3873.
Clemmensen, L., Hastie, T., Witten, D., Ersbøll, B., 2011. Sparse discriminant analysis. Technometrics 53, 406–413.
Dellaportas, P., Pourahmadi, M., 2012. Cholesky-garch models with applications to finance. Stat. Comput. 22, 849–855.
Durrant, R.J., Kabán, A., 2015. Random projections as regularizers: learning a linear discriminant from fewer observations than dimensions. Mach. Learn. 99, 257–286.
Freund, Y., Schapire, R.E., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. 55, 119–139.
Friedman, J., Hastie, T., Tibshirani, R., 2007. Sparse inverse covariance estimation with the graphical lasso. Biostatistics 9, 432–441.
Glaab, E., Bacardit, J., Garibaldi, J.M., Krasnogor, N., 2012. Using rule-based machine learning for candidate disease gene prioritization and sample classification of cancer gene expression data. PLoS ONE 7, e39932.
Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., Lander, E.S., 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286, 531–537.
Hastie, T., Tibshirani, R., 1996. Discriminant analysis by gaussian mixtures. J. R. Stat. Soc., Ser. B, Methodol. 58, 155–176.
Ho, T.K., 1998. The random subspace method for constructing decision forests. IEEE Trans. Pattern Anal. Mach. Intell. 20, 832–844.
Huang, J., Liu, N., Pourahmadi, M., Liu, L., 2006. Covariance matrix selection and estimation via penalised normal likelihood. Biometrika 93, 85–98.
Kang, X., Deng, X., 2020. An improved modified cholesky decomposition approach for precision matrix estimation. J. Stat. Comput. Simul. 90, 443–464.
Kang, X., Deng, X., Tsui, K., Pourahmadi, M., 2020. On variable ordination of modified cholesky decomposition for estimating time-varying covariance matrices. Int. Stat. Rev. 88, 616–641.
Kang, X., Wang, M., 2021. Ensemble sparse estimation of covariance structure for exploring genetic disease data. Comput. Stat. Data Anal. 159, 107220.
Li, C., Yang, M., Wang, M., Kang, H., Kang, X., 2021. A cholesky-based sparse covariance estimation with an application to genes data. J. Biopharm. Stat. 31, 603–616.
Liu, J., Yu, G., Liu, Y., 2019. Graph-based sparse linear discriminant analysis for high-dimensional classification. J. Multivar. Anal. 171, 250–269.
Mai, Q., Zou, H., Yuan, M., 2012. A direct approach to sparse discriminant analysis in ultra-high dimensions. Biometrika 99, 29–42.
Pourahmadi, M., 1999. Joint mean-covariance models with applications to longitudinal data: unconstrained parameterisation. Biometrika 86, 677–690.
Rajaratnam, B., Salzman, J., 2013. Best permutation analysis. J. Multivar. Anal. 121, 193–223.
Rodriguez, J., Kuncheva, L., Alonso, C., 2006. Rotation forest: a new classifier ensemble method. IEEE Trans. Pattern Anal. Mach. Intell. 28, 1619–1630.
Rothman, A.J., Levina, E., Zhu, J., 2009. Generalized thresholding of large covariance matrices. J. Am. Stat. Assoc. 104, 177–186.

Shao, J., Wang, Y., Deng, X., Wang, S., 2011. Sparse linear discriminant analysis by thresholding for high dimensional data. Ann. Stat. 39, 1241–1265.
Wang, S., Xie, C., Kang, X., 2023. A novel robust estimation for high-dimensional precision matrices. Stat. Med. 42, 656–675.
Wu, M.C., Zhang, L., Wang, Z., Christiani, D.C., Lin, X., 2009. Sparse linear discriminant analysis for simultaneous testing for the significance of a gene set/pathway and gene selection. Bioinformatics 25, 1145–1151.
Xue, L., Ma, S., Zou, H., 2012. Positive-definite l_1-penalized estimation of large covariance matrices. J. Am. Stat. Assoc. 107, 1480–1491.
Yuan, M., Lin, Y., 2007. Model selection and estimation in the Gaussian graphical model. Biometrika 94, 19–35.
Zhang, R., Fattahi, S., Sojoudi, S., 2018. Large-scale sparse inverse covariance estimation via thresholding and max-det matrix completion. In: Proceedings of the 35th International Conference on Machine Learning, pp. 5766–5775.
Zheng, H., Tsui, K.W., Kang, X., Deng, X., 2017. Cholesky-based model averaging for covariance matrix estimation. Stat. Theory Relat. Fields 1, 48–58.