High-dimensional covariance matrix regularization using more informative targets

RESEARCH SUPERVISOR
DR. SYED MUHAMMAD ASIM

SUBMITTED BY
ATIQ UR REHMAN
ROLL NO: 28
BS: STATISTICS

DEPARTMENT OF STATISTICS
UNIVERSITY OF PESHAWAR

SESSION: 2013-2017

High-dimensional covariance matrix regularization using more informative targets

A thesis submitted to Department of Statistics University of Peshawar in partial fulfillment of the requirement for the degree of BS in Statistics.

RESEARCH SUPERVISOR
DR. SYED MUHAMMAD ASIM

SUBMITTED BY
ATIQ UR REHMAN
ROLL NO: 28
BS: STATISTICS

DEPARTMENT OF STATISTICS
UNIVERSITY OF PESHAWAR
SESSION: 2013-2017

APPROVAL SHEET

This is to certify that the thesis submitted by Mr. ATIQ UR REHMAN, in partial fulfillment of the requirements for the award of the degree of BS in statistics has been approved by the supervisory committee.

RESEARCH SUPERVISOR

CHAIRMAN

EXTERNAL EXAMINER

December 2017

Abstract

In high-dimensional datasets, where the number of variables, p, are greater than the sample size, n, the non invertibility and ill-conditioning ($p \approx n$) of the sample covariance matrix provokes serious problems in many statistical applications. To overcome this problem a number of methods have been proposed in the literature. In this thesis, first we explore some well-known regularization methods. Second, we propose new method to regularize the sample covariance matrix, which depends on the penalty parameter and need to be chosen in the appropriate range of values. We make use of the likelihood function of multivariate normal distribution to choose an appropriate value of the penalty parameter. The new regularize estimator also dependent on the target matrix towards which we shrink the sample covariance matrix. A number of target matrices have been used in various methods. We use two more informative targets and shrink the sample covariance matrix towards them. These two targets matrix are the AR(1) and exchangeable covariance structures, which depends on the correlation parameter and needs to be estimated as well. We use the likelihood function of multivariate normal distribution to estimate the correlation parameter. To check the performance of the proposed method in comparison with the available shrinkage method, a simulation study has been conducted, which show that the proposed method is quite effective and perform better than the shrinkage method. Furthermore, the proposed method is analytically simpler and computationally less expensive in comparison to some of the available regularization methods.

Acknowledgements

Thanks to Almighty Allah for all the countless gifts you have offered me, and thanks to my family for their constant love and firm support.

It is a great pleasure to express my sincere thanks to my supervisor prof. Dr. Muhammad Asim, Chairman, Department of Statistics, University of Peshawar for his invaluable guidance and support to work on a topic that was of great interest to me and could not have been possible without his support. It was great honour to work under his supervision.

I would like to express my gratitude to all the faculty members of the Department of Statistics, University of Peshawar for their constant support and encouragement during the studies. I would also like to thank my class fellows and friends for their support.

I am extremely thankful to Dr. Insha Ullah, Kohat University of Science and Technology for providing me all necessary facilities throughout the studies and research in particular. I am extremely lucky to have such a brilliant brother as an experienced and trusted adviser.

I would like to dedicate this thesis to my family and friends...

Contents

Abstract i

Acknowledgements ii

List of Figures v

1 Introduction 1

2 Literature review 3
- 2.1 Introduction . . . 3
- 2.2 Shrinkage estimation . . . 4
- 2.3 Ridge regularization of the covariance matrix . . . 6
- 2.4 Covariance matrix regularization via lasso . . . 7

3 Informative targets and regularization of covariance matrix using informative targets 9
- 3.1 Introduction . . . 9
- 3.2 Estimation of Correlation parameter . . . 10
- 3.3 Estimation of regularization parameter using normal likelihood . . . 13
  - 3.3.1 Sample properties of γ . . . 15

4 Simulation study 17
- 4.1 Introduction . . . 17
- 4.2 Synthetic experiments . . . 17

5 Conclusion 26

A R codes for Figures 28

References 48

List of Figures

1.1 Sorted eigenvalues of the true and sample covariance matrices for a fixed p=50 and $n = \{25,\ 50,\ 100,\ 1000\}$ drawn from multivariate normal distribution with Identity matrix as a true covariance matrix. 2

3.1 Distribution of γ values for different samples of size $n = \{10, 30, 300\}$ from a multivariate normal distribution with $p = \{10, 30, 100\}$ for different choices of $\mathbf{\Sigma}$ simulated 1000 times. (a) identity matrix as a true covariance matrix and as target (b) AR(1) structure as a true covariance matrix and as a target (c) exchangeable structure as a true covariance matrix and as a target. 16

4.1 Distribution of the sum of absolute errors in estimated eigenvalues simulated 1000 times using proposed, shrinkage and maximum likelihood methods under different choices of covariance matrices (a, b) AR(1) with $t = 0.5$ and identity as true covariance matrix for all three methods and AR(1) as a target for the proposed method (c, d) exchangeable with $t = 0.5$ and identity as true covariance matrix for all three methods and exchangeable as a target for the proposed method (e) random covariance structure with 30% off-diagonal entries as non-zero and identity matrix as a target. The data are generated from multivariate normal distribution with $n = 50$ and $p = \{30, 50, 100\}$. 21

4.2 Comparison of MSE averaged over 1000 simulations of the estimated covariance matrices. The data are drawn from multivariate normal distribution with sample size $n = \{10, 20, 30, 40, 50, 60, 70, 90, 120, 200\}$ and $p = 10$ under different choices of covariance matrices (a, b) AR(1) with $t = 0.5$ and identity as true covariance matrix for all three methods and AR(1) as a target for the proposed method (c, d) exchangeable with $t = 0.5$ and identity as true covariance matrix for all three methods and exchangeable as a target for the proposed method (e) random covariance structure with 30% off-diagonal entries as non-zero and identity matrix as a target. 23

4.3 Comparison of estimated and true eigenvalues with $n = 30$ and $p = 50$ averaged over 1000 simulation. The data are drawn from multivariate normal distribution under different choices of covariance matrices (a, b) AR(1) with $t = 0.5$ and identity as true covariance matrix for all three methods and AR(1) as a target for the proposed method (c, d) exchangeable with $t = 0.3$ and identity as true covariance matrix for all three methods and exchangeable as a target for the proposed method (e) random covariance structure with 30% off-diagonal entries as non-zero and identity matrix as a target. 25

Chapter 1

Introduction

High-dimensional datasets, where the number of variables, p, is greater then the sample size, n, are increasingly becoming common in many feilds, particularly in genatics. For example, gene expresion dataset used by Eisen et al. (1998) has 2467 variables and 79 samples. Another study by Tamayo et al. (1999) has 6817 variables (human genes) obtained from 72 microarray images. A relatively recent cancer study by Beerenwinkel et al. (2007) which is based on a dataset with 78 genes (p) and only 35 samples (n).

In these types of high-dimensional datasets, the sample covariance matrix—maximum likelihood estimator and its unbiased version—perform poorly and are not considered a good approximation to the true covariance matrix (even if n is comparable to p). This is because the sample covariance matrix contains estimation error and their eigenvalues tends to be overdispersed; that is, the larger eigenvalues will contain a high amount of positive errors (overestimated) and smaller eigenvalues will contain a high amount of negative errors (underestimated). In addition, the inverse covariance matrix is fundamental to multivariate methods comprising regression, Gaussian graphical models, linear discriminant analysis and Mahalanobis distance. The sample covariance matrix loses its full rank and is not invertible if p exceeds n. The non-invertibility of the sample covariance matrix renders the above mentioned multivariate methods inapplicable.

To make things more clear, we conduct a small simulation study. We draw samples of size $n = \{25,\ 50,\ 100,\ 1000\}$ from a p-variate normal distribution with mean vector, $\boldsymbol{\mu} = \mathbf{0}$, and an identity covariance matrix, $\boldsymbol{\Sigma} = \boldsymbol{I}$. We fix the number of variables, $p = 50$. For each value of n we repeat the simulations 1000 times and the average estimated eigenvalues are portrayed in Figure 1.1. It is clear from the Figure that, due to estimation error, the larger eigenvalues are overestimated and the smaller eigenvalues are underestimated. The estimation error decreases as we increase the sample size. Moreover, if the number of observations are less then the number of variables, the sample covariance matrix becomes singular and is not invertible (product of the eigenvalues become zero).

To deal with high-dimensional covariance estimation problems, various methods have been proposed in the previous literature. In this Thesis, first we want to explore some of the well known shrinkage (regularization) methods. Further, these shrinkage methods rely on a tuning parameter whose value need to be chosen in a suitable range of values. An appropriate choice of the tuning parameter leads to improved estimate of the covariance matrix. Our second objective is to choose an appropriate value of the tuning parameter, which we achieve by maximizing a multivariate normal likelihood function. Third, we shrink the sample covariance matrix towards more informative target estimators rather than using identity matrix as a target estimator.

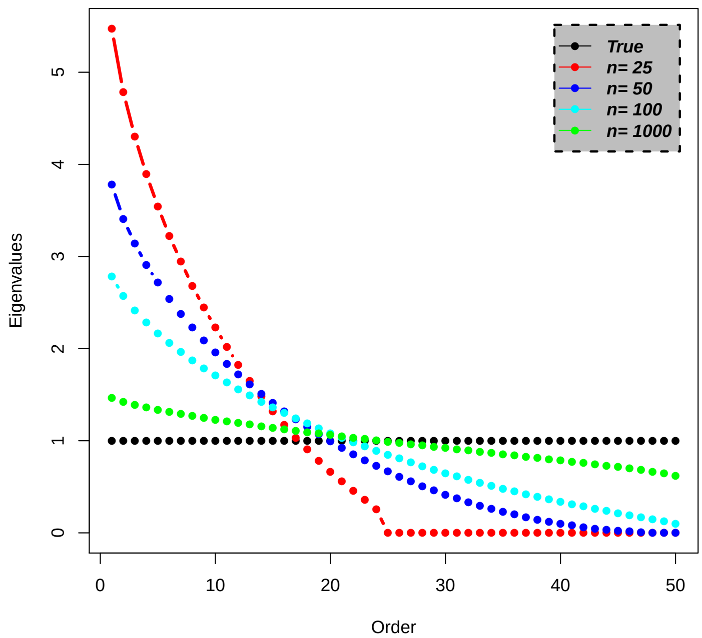

FIGURE 1.1: Sorted eigenvalues of the true and sample covariance matrices for a fixed p=50 and $n = \{25, 50, 100, 1000\}$ drawn from multivariate normal distribution with Identity matrix as a true covariance matrix.

Chapter 2

Literature review

2.1 Introduction

Under a large sample size, the population covariance matrix can be accurately estimated by the sample covariance matrix (maximum likelihood and related unbiased estimator). Consider a vector of random variables, $\mathbf{X} = (X_1, X_2, ..., X_p)$, drawn from a p-variate normal distribution with mean vector, $\boldsymbol{\mu}$, and covariance matrix, $\boldsymbol{\Sigma}$. The multivariate probability density function of $\mathbf{X}$ can be written as

$$f(\mathbf{X};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{p}{2}}\,|\boldsymbol{\Sigma}|^{\frac{1}{2}}}\exp[-\boldsymbol{(X-\mu)^t\Sigma^{-1}(X-\mu)}], \tag{2.1}$$

where $|\boldsymbol{A}|$ represent the determinant of a matrix $\boldsymbol{A}$ and $\boldsymbol{A^t}$ represent the transpose of a matrix $\boldsymbol{A}$. However, in practice, the true covariance matrix is unknown and we estimate it from the sample data. The most common approach is to use an estimate which maximize the following log likelihood function:

$$\log L(\boldsymbol{X};\boldsymbol{\Sigma}) = Const - \frac{n}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\boldsymbol{X^t\Sigma^{-1}X}. \tag{2.2}$$

Note that, in equation 2.2, without loss of generality, we assume the mean vector $\boldsymbol{\mu} = \mathbf{0}$. After differentiating equation 2.2 with respect to $\boldsymbol{\Sigma}$ and equating it to zero, we obtain the maximum likelihood estimate of covariance matrix given by $\widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\boldsymbol{X^tX}$. The related unbiased estimate is given by $\boldsymbol{S} = \frac{\boldsymbol{n}}{\boldsymbol{n-1}}\widehat{\boldsymbol{\Sigma}}$. It is noteworthy that when the number of samples is very large, both estimators become equal. Moreover, these estimators of the covariance matrix have some desirable properties When the sample size is large. First, their eigenvalues are closely related to their population counterpart. Second, both estimators are positive definite matrices so they can be inverted to obtain the estimate of the inverse covariance matrix, $\mathbf{\Sigma}^{-1}$.

However, in high-dimensional settings, the classical multivariate techniques which uses the sample covariance matrix or its inverse is a key ingredient either fails to work or becomes unreliable. Because of the two undesirable properties of the sample covariance matrix. First, the sample covariance matrix cannot be inverted. Second, the sample covariance matrix contains a massive amount of an estimation error, which can make considerable adverse impacts on the estimation accuracy (Fan et al., 2016).

To overcome this problem, a number of methods have been proposed in the literature. One method is the Moore-Penrose generalized inverse proposed by Penrose (1955), which is based on the singular value decomposition (SVD). In high-dimensional data, $(p \gg n)$ Moore-Penrose generalized inverse is often used to find the inverse of the sample covariance matrix $\hat{\mathbf{\Sigma}}$. To find the inverse of $\hat{\mathbf{\Sigma}}$ it is decomposed as $\hat{\mathbf{\Sigma}} = \boldsymbol{UDV^t}$, where $\boldsymbol{U}$ and $\boldsymbol{V}$ are the matrices of orthonormal eigenvectors and $\boldsymbol{D}$ is the diagonal matrix with diagonal elements equal to the square root of the eigenvalues of $\hat{\mathbf{\Sigma}}\hat{\mathbf{\Sigma}}^{\boldsymbol{t}}$. Moore-Penrose generalized can be achieved by using the equation

$$\hat{\mathbf{\Sigma}}^{\mathbf{-1}} = \boldsymbol{VD^{-1}U^t}, \tag{2.3}$$

where all the zero diagonal elements of $\boldsymbol{D}$ and the corresponding eigenvectors in $\boldsymbol{U}$ and $\boldsymbol{V}$ are removed before finding generalized inverse given in equation 2.3. It is interesting to note that Moore-Penrose generalized inverse reduces to the standard matrix inverse whenever $rank(\hat{\mathbf{\Sigma}}) \geq p$ (Golub & Kahan, 1965). Other regularization procedures closely related to our work are explored in the following sections.

2.2 Shrinkage estimation

Historically, the idea of shrinkage estimation is going back to Stein et al. (1956) who observed that the estimator can be improved through shrinking towards the structure target. The same idea is used by the Ledoit & Wolf (2004) who proposed the procedure to find an estimator by the convex combination of the sample covariance matrix and a target matrix. This convex combination is as follows:

$$\hat{\boldsymbol{\Sigma}}_{\gamma} = \gamma \boldsymbol{T} + (1-\gamma)\hat{\boldsymbol{\Sigma}}, \tag{2.4}$$

where $\hat{\boldsymbol{\Sigma}}_{\gamma}$ is the improved estimator and $\boldsymbol{T}$, $\hat{\boldsymbol{\Sigma}}$ are the target matrix and maximum likelihood estimator of the covariance matrix respectively. They provided a procedure to find shrinkage intensity γ by minimizing the expected square loss function, given by

$$R(\gamma) = E\left\|\hat{\boldsymbol{\Sigma}}_{\gamma} - \hat{\boldsymbol{\Sigma}}\right\|^2, \tag{2.5}$$

where expected square loss function is the measure of mean square error. Interestingly, there is no need to assume that the random variables p follows any specific distribution. But this procedure assumed to exist the first four moments (Schäfer & Strimmer, 2005). It can be shown that this improved estimator is well-conditioned (Ledoit & Wolf, 2004).

Schäfer & Strimmer (2005) followed the same procedure for computing the shrinkage parameter. To compute γ minimizing equation 2.5 with respect to γ we get,

$$\hat{\gamma} = \frac{\sum_{i=1}^{p}\sum_{j=1}^{p} var(\hat{\sigma}_{ij}) - cov(t_{ij},\hat{\sigma}_{ij}) - bias(\hat{\sigma}_{ij})E(t_{ij}-\sigma_{ij})^2}{\sum_{i=1}^{p}\sum_{j=1}^{p} E[t_{ij}-\sigma_{ij}]^2}, \tag{2.6}$$

they described some insights into how the γ should be chosen and derived this analytic equation to obtain shrinkage intensity for six commonly used targets for detailed discussion see (Schäfer & Strimmer, 2005). Note that if $\hat{\boldsymbol{\Sigma}}$ is an unbiased estimator then equation 2.6 reduces to

$$\hat{\gamma} = \frac{\sum_{i=1}^{p}\sum_{j=1}^{p} var(\hat{\sigma}_{ij}) - cov(t_{ij},\hat{\sigma}_{ij})}{\sum_{i=1}^{p}\sum_{j=1}^{p} E[t_{ij}-\sigma_{ij}]^2}. \tag{2.7}$$

Using the identity matrix where all the variables are normalized to have unit variance and its scalar multiple is relatively easy as target $\boldsymbol{T}$ from both analytical and computational perspective. Which is employed by Ledoit & Wolf (2003) and Ledoit & Wolf (2004). They also demonstrated that the improved estimator is well-conditioned and more accurate than the sample covariance matrix.

Another target matrix which was the main focus of Schäfer & Strimmer (2005) is the diagonal matrix $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{d}}$ with unequal variances on the main diagonal. This $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{d}}$ only shrinks the eignvalues and keeps the eigenvectors unchanged. In this case $\hat{\gamma}$

is given by

$$\hat{\gamma} = \frac{\sum_{i \neq j}^{p} var(s_{ij})}{\sum_{i \neq j}^{p} E(s_{ij}^2)}. \tag{2.8}$$

To compute $\hat{\gamma}$ in this case requares p parameters to be estimated which is complicated as compare to the identity matrix. Note that both identity matrix and $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{d}}$ are positive and sample covariance matrix is the semi-positive definite taking convex combination of one of these targets and sample covariance matrix would result in a positive definite matrix.

2.3 Ridge regularization of the covariance matrix

As described, in high dimensional settings ($p \gg n$) the maximum likelihood estimator of the covariance matrix become singular and ill-conditioned. A method so called ridge regularization, proposed by Warton (2008) resolve this problem by using

$$\hat{\boldsymbol{\Sigma}}_{\kappa} = \hat{\boldsymbol{\Sigma}} + \kappa \boldsymbol{I}, \tag{2.9}$$

where κ is the ridge parameter, $\boldsymbol{I}$ is the $p \times p$ identity matrix and $\hat{\boldsymbol{\Sigma}}_{\kappa}$ is the regularized estimator of the covariance matrix. When the variables are at different scales, it is more appropriate to regularize on the standard scale. In this case Warton (2008) regularize the sample estimator of the correlation matrix, $\boldsymbol{R}$, which can be obtained by rescaling equation 2.9 as

$$\hat{\boldsymbol{R}}_{\gamma} = \gamma \hat{\boldsymbol{R}} + (1-\gamma)\boldsymbol{I}, \tag{2.10}$$

where $\gamma = \frac{1}{1+\kappa} \in (0,1]$ is the ridge parameter and $\hat{\boldsymbol{R}}_{\gamma}$ is the regularized estimator of the correlation matrix. It is the shrinkage estimator as it shrinks $\hat{\boldsymbol{R}}$ toward the identity matrix and also guaranteed to be a positive definite matrix for any value of $\gamma \in (0,1]$. One interesting property of $\hat{\boldsymbol{R}}_{\gamma}$ is that it can be derived from the penalized likelihood function for multivariate normal data, with penalty term proportional to $\mathrm{tr}(\boldsymbol{R}^{-1})$ see (Warton, 2008) for analytical derivation. The penalize likelihood function is given by

$$\log L(\boldsymbol{X}; \boldsymbol{\Sigma}) = Const - \frac{n}{2} \log |\boldsymbol{\Sigma}| - \frac{1}{2} \boldsymbol{X}^{\boldsymbol{t}} \boldsymbol{\Sigma}^{-1} \boldsymbol{X} - \frac{c}{2} tr(\boldsymbol{R}^{-1}). \tag{2.11}$$

Using equation 2.10 the regularized estimator of $\mathbf{\Sigma}_\gamma$ can be obtained as

$$\hat{\mathbf{\Sigma}}_\gamma = \hat{\mathbf{\Sigma}}_d^{1/2}(\gamma\hat{\boldsymbol{R}} + (1-\gamma)\boldsymbol{I})\hat{\mathbf{\Sigma}}_d^{1/2}. \tag{2.12}$$

To estimate regularization parameter γ, Warton (2008) is using k-fold cross validation. In this case k-fold cross validation is done by dividing the whole sample of size n of a matrix $\mathbf{X}$ into $\mathbf{k}$ sub-samples denoted by $\mathbf{X} = [\mathbf{X}_1^T, \mathbf{X}_2^T, ..., \mathbf{X}_\mathbf{K}^T]$. In which the K-th sub-samples, that is, $\mathbf{X}_K$ is used as the validation data and the rest of the observations are used as training data. For example, a total sample size is 20 and we divide it into 5 equal parts in which each sub-sample consist of 4 observations. The training data, $\mathbf{X}_K$, is used to compute its mean, covariance matrix and correlation matrix denoted by $\boldsymbol{\mu}^{\setminus k}$, $\mathbf{\Sigma_\gamma}^{\setminus k}$ and $\boldsymbol{R}^{\setminus k}$ respectively. The observed likelihood is then calculated for each $\mathbf{X}_K$. And then estimate γ by maximizing the cross validation likelihood function which is given by

$$\begin{aligned} -2\log L(\boldsymbol{\mu}^{\setminus k}, \mathbf{\Sigma}^{\setminus k}; \boldsymbol{X}) = (n_k p)\log(2\pi) + n_k \log\left|\hat{\mathbf{\Sigma}}_\gamma^{\setminus k}\right| \\ + tr[(\boldsymbol{X_k} - \boldsymbol{\mu}^{\setminus \boldsymbol{k}})(\hat{\mathbf{\Sigma}}_\gamma^{\setminus \boldsymbol{k}})^{-\mathbf{1}}(\boldsymbol{X_k} - \boldsymbol{\mu}^{\setminus \boldsymbol{k}})]. \end{aligned} \tag{2.13}$$

To obtain an optimal value of γ we use the following equation.

$$\gamma = \underset{\gamma}{\operatorname{argmax}} \sum_{k=1}^{K} \log L(\boldsymbol{\mu}^{\setminus k}, \mathbf{\Sigma}^{\setminus k}; \boldsymbol{X_k}). \tag{2.14}$$

2.4 Covariance matrix regularization via lasso

The Inverse of a covariance matrix of the multivariate normal distribution is used to find out conditional independence relationship between two variables given the rest of $p-2$. These conditional dependencies can be visualized graphically called the Gaussian graphical model. However, the population inverse covariance matrix is unknown and we estimate it by the two well known estimators (maximum likelihood estimator and its unbiased version). These two estimators cannot produce estimated elements exactly eqaul to zero, no matter what the sample size is if they are zero in the true covariance matrix. Which makes the model unnecessarily more complex. This complexity and noise of the inverse covariance matrix can be reduced by setting some of the elements equal to zero, a technique called covariance selection proposed by Dempster (1972).

The lasso regularization was first introduced by Tibshirani (1996) in the regression context in order to enhance the accuracy and interpretability of the model by setting some of the coefficients exactly equal to zero and shrink important coefficient toward zero. This idea was used by Yuan & Lin (2007) and d'Aspremont et al. (2008) using the penalized log-likelihood method and derived different lasso algorithms for the sparse covariance selection. A fastest algorithm is the graphical lasso algorithm (Glasso) introduced by Friedman et al. (2008) for estimating the inverse covariance matrix by applying the lasso penalty. The lasso problem can be solved by using the coordinate decent algorithm (Friedman et al., 2007).

Chapter 3

Informative targets and regularization of covariance matrix using informative targets

3.1 Introduction

In this chapter, we use the steinian-class shrinkage estimation which is the convex linear combination of the sample covariance matix, $\hat{\mathbf{\Sigma}}$, and the target matrix, $\mathbf{T}$, given by

$$\hat{\mathbf{\Sigma}}_\gamma = \gamma \mathbf{T} + (1 - \gamma)\hat{\mathbf{\Sigma}}, \tag{3.1}$$

where $\gamma \in [0, 1]$ is the shrinkage parameter. The target matrix need to be pre-specified and an appropriate value of γ need to be chosen over a grid of values. Note that, when $\gamma = 0$ no shrinkage is applied and the sample covariance matrix is retained, and when $\gamma = 1$ full shrinkage is applied, which results $\mathbf{T}$ as an estimator of the covariance matrix. Ledoit & Wolf (2003) and Ledoit & Wolf (2004) uses identity matrix as a target estimator and the R package "corpcor" specify identity matrix as a target (Schaefer et al., 2013). But sometimes it may not be a good choice as explained by Schäfer & Strimmer (2005), the identity matrix shrinks all the diagonal and off-diagonal elements of the sample covariance matrix and consequently change the whole eigenstructure of the sample covariance matrix. Schäfer & Strimmer (2005) also discussed the six commonly used targets including identity matrix and their main focus was the diagonal matrix as a target with diagonal elements variances and off-diagonal elements zero pre-assuming that all the variables are independent, which only shrinks the eigenvalues and leave the eigenvectors intact.

We use two more informative target matrices, that are, first order auto-regressive AR(1) and exchangeable covariance structures for which the correlation parameter, $t \in [0,1]$, is the essential element. We maximize the likelihood function of the multivariate normal distribution to choose an appropriate value of the correlation parameter as described in the next section. Next, we calculate the appropriate value of the shrinkage parameter via maximizing the likelihood function of the multivariate normal distribution. Moreover, we also obtain the optimal shrinkage intensity for the above mentioned targets by minimizing the expected quadratic loss function.

3.2 Estimation of Correlation parameter

Correlation parameter is the essential element of the two covariance structures, namely AR(1) and exchangeable covariance structures which needs to be estimated. The AR(1) covariance structure can be defined as the first order autoregressive structure which considers the correlation systematically decreasing with increasing the distance between the time points given by

$$\sigma_{ij} = t^{|i-j|} \quad \text{for} \quad 1 \leq i, j \leq p, \tag{3.2}$$

whereas exchangeable covariance structure can be defined as the matrix with the same covariance between variables and the variances remains constant by rearranging (exchanging) the variables given by

$$\sigma_{ij} = \begin{cases} 1 & \text{when } i = j \\ t & \text{when } i \neq j \end{cases} \quad \text{for} \quad 1 \leq i, j \leq p, \tag{3.3}$$

where t is the constant correlation parameter and can be obtained by simply maximizing the log-likelihood function of the multivariate normal distribution for both covariance structures. Let's denote the AR(1) and exchangeable covariance structures by $\boldsymbol{\Sigma_t}$, the log-likelihood function can be written as

$$\log L(\boldsymbol{X};\boldsymbol{\Sigma_t}) = Const - \frac{n}{2}\log|\boldsymbol{\Sigma_t}| - \frac{1}{2}\boldsymbol{X^t\Sigma_t^{-1}X}. \tag{3.4}$$

Differentiating equation 3.4 with respect to t we get

$$\frac{\partial}{\partial t}\log L(\boldsymbol{X};\boldsymbol{\Sigma_t}) = -\frac{n}{2}\frac{\partial}{\partial t}\log|\boldsymbol{\Sigma_t}| - \frac{1}{2}\mathrm{tr}\big(\boldsymbol{X^t}\boldsymbol{\frac{\partial}{\partial t}}\boldsymbol{\Sigma_t^{-1}X}\big). \tag{3.5}$$

Solving equation 3.5 for AR(1) covariance structure the determinant of $\boldsymbol{\Sigma_t}$ is $|\boldsymbol{\Sigma_t}| = (1-t^2)^{p-1}$ and the derivative of $\log|\boldsymbol{\Sigma_t}|$ with respect to t gives

$$\frac{\partial}{\partial t}\log|\Sigma_t| = \frac{-2(p-1)t(1-t^2)^{p-2}}{(1-t^2)^{p-1}}. \tag{3.6}$$

The inverse of $\boldsymbol{\Sigma_t}$ is given by

$$\Sigma_t^{-1} = \frac{1}{(1-t^2)}\begin{pmatrix} 1 & -t & 0 & \dots & 0 & 0 \\ -t & 1+t^2 & -t & \dots & 0 & 0 \\ 0 & -t & 1+t^2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1+t^2 & -t \\ 0 & 0 & 0 & \dots & -t & 1 \end{pmatrix}.$$

The derivative of $\boldsymbol{\Sigma_t^{-1}}$ with respect to t is given by

$$\frac{\partial}{\partial t}\Sigma_t^{-1} = \frac{1}{(1-t^2)^2}\begin{pmatrix} 2t & -(1+t^2) & 0 & \dots & 0 & 0 \\ -(1+t^2) & 4t & -(1+t^2) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 4t & -(1+t^2) \\ 0 & 0 & 0 & \dots & -(1+t^2) & 2t \end{pmatrix}.$$

To find $\mathrm{tr}\big(X^t\frac{\partial}{\partial t}\Sigma_t^{-1}X\big)$ in equation 3.5, we can write $\boldsymbol{X^tX = n\Sigma}$, where $\boldsymbol{\Sigma}$ is true covariance matrix with entries $\sigma_{ij}, 1 \le i,j \le p$, then

$$\begin{aligned} n\,\mathrm{tr}\big(\frac{\partial}{\partial t}\Sigma_t^{-1}\Sigma\big) =& \frac{1}{(1-t^2)^2}[2t(\sigma_{11} + 2\sigma_{22} + 2\sigma_{33} + \dots + 2\sigma_{(p-1)(p-1)} + \sigma_{pp})] \\ &- t^2(\sigma_{12} + \sigma_{21} + \sigma_{23} + \dots + \sigma_{(p-1)(p)} + \sigma_{p(p-1)}) \\ &- (\sigma_{12} + \sigma_{21} + \sigma_{23} + \dots + \sigma_{(p-1)(p)} + \sigma_{p(p-1)}), \end{aligned} \tag{3.7}$$

since $\boldsymbol{\Sigma}$ is a symmetric matrix, i.e, $\sigma_{ij} = \sigma_{ji}$ and also diagonal elements are all equal to 1, we can write equation 3.7 as

$$n\,\mathrm{tr}\big(\frac{\partial}{\partial t}\Sigma_t^{-1}\Sigma\big) = \frac{2}{(1+t^2)^2}[t(p-1)-(t^2+1)\sum_{i=2}^{p}\sigma_{(i-1)i}]. \tag{3.8}$$

Substituting equation $\frac{\partial}{\partial t}\log|\Sigma_t|$ and $n\,\mathrm{tr}\big(\frac{\partial}{\partial t}\Sigma_t^{-1}\Sigma\big)$ in equation 3.5 and equating it to zero leads to

$$t = \frac{\sum_{i=2}^{p}\sigma_{(i-1)i}}{p-1}. \tag{3.9}$$

If $\boldsymbol{\Sigma_t}$ is the exchangeable covariance structure then $|\Sigma_t| = (1-t)^{p-1}\{1+(p-1)t\}$, differentiating $\log|\Sigma_t|$ with respect to t gives

$$\frac{\partial}{\partial t}\log|\Sigma_t| = \frac{(1-t)^{p-1}(p-1)+\{1+(p-1)t\}(p-1)(1-t)^{p-2}}{(1-t)^{p-1}\{1+(p-1)t\}}, \tag{3.10}$$

in this case the inverse of Σ_t is

$$\begin{aligned}\Sigma_t^{-1} =& \frac{1}{(1-t)\{1+(p-1)t\}}\begin{pmatrix}1+t & -t & \dots & -t\\ -t & 1+t & \dots & -t\\ \vdots & \vdots & \vdots & \vdots\\ -t & -t & \dots & 1+t\end{pmatrix}\\ =& \frac{1}{(1-t)}\big[\mathbf{I}-\frac{t}{\{1+(p-1)t\}}\mathbf{J}\big],\end{aligned}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{J}$ is the unit matrix, differentiating Σ_t^{-1} with respect to t gives

$$\frac{\partial}{\partial t}\Sigma_t^{-1} = \frac{1}{(1-t)^2}\mathbf{I} - \Big[\frac{1}{\{1+(p-1)t\}^2(1-t)}+\frac{t}{\{1+(p-1)t\}(1-t)^2}\Big]\mathbf{J}. \tag{3.11}$$

The

$$\begin{aligned}\mathrm{tr}(\frac{\partial}{\partial t}\Sigma_t^{-1}\Sigma) =& \frac{p}{(1-t)^2} - p\big[\frac{1}{\{1+(p-1)t\}^2(1-t)}+\frac{t}{\{1+(p-1)t\}(1-t)^2}\big]\\ & -\big[\frac{1}{\{1+(p-1)t\}^2(1-t)}+\frac{t}{\{1+(p-1)t\}(1-t)^2}\big]\sum_{i\neq j}^{p}\sigma_{ij}.\end{aligned} \tag{3.12}$$

Substituting the values of $n\,\mathrm{tr}(\frac{\partial}{\partial t}\Sigma_t^{-1}\Sigma)$ and $\frac{\partial}{\partial t}\log|\Sigma_t|$ in equation 3.5 and equating it to zero leads to

$$t = \frac{\sum_{i\neq j}^{p}\sigma_{ij}}{p(p-1)}. \tag{3.13}$$

3.3 Estimation of regularization parameter using normal likelihood

To obtain the regularized estimator of the true covariance matrix, we propose to maximize the multivariate normal likelihood function. Given a random sample of size n from p-variate normal distribution with mean vector, $\boldsymbol{\mu} = \mathbf{0}$, and covariance matrix, $\mathbf{\Sigma}$. The log-likelihood function is

$$\log L(\boldsymbol{X}; \mathbf{\Sigma}) = Const - \frac{n}{2} \log |\mathbf{\Sigma}| - \frac{1}{2} \boldsymbol{X}^{\boldsymbol{t}} \mathbf{\Sigma}^{\mathbf{-1}} \boldsymbol{X}. \tag{3.14}$$

In high-dimensional applications the sample estimator of the covariance matrix is not invertible, we use

$$\hat{\mathbf{\Sigma}}_{\kappa} = \hat{\mathbf{\Sigma}} + \kappa \boldsymbol{T}, \tag{3.15}$$

where $\boldsymbol{T}$ is a positive definite informative target matrix and $\kappa > 0$ is the regularization parameter. The expression in 3.15 incorporates additional information that we may have about the structure of the covariance matrix. If the sample size is very small, $\hat{\mathbf{\Sigma}}_{\kappa}$ becomes similar (if not equal) to $\boldsymbol{T}$ that is

$$\boldsymbol{T} \approx \hat{\mathbf{\Sigma}} + \kappa \boldsymbol{T}.$$

This leads to

$$\kappa \boldsymbol{T} \approx \boldsymbol{T} - \hat{\mathbf{\Sigma}}, \tag{3.16}$$

which can be exploited to find the value of κ as we do later in this section.

The expression in 3.16 can also be obtained by assuming that some scaled version of $\boldsymbol{T}$ is the true covariance and replace $\mathbf{\Sigma}$ by $(1+\kappa)\boldsymbol{T}$ in equation 3.14, which then becomes

$$\log L(\boldsymbol{X}; \mathbf{\Sigma}_{\kappa}) = Const - \frac{n}{2} \log |\boldsymbol{T} + \kappa \boldsymbol{T}| - \frac{1}{2} \boldsymbol{X}^{\boldsymbol{t}} \frac{\mathbf{1}}{\mathbf{1} + \boldsymbol{\kappa}} \boldsymbol{T}^{\mathbf{-1}} \boldsymbol{X}. \tag{3.17}$$

Differentiating 3.17 with respect to κ we get

$$\frac{\partial}{\partial \kappa} \log L(\boldsymbol{X}; \mathbf{\Sigma}_{\kappa}) = -\frac{n}{2} (\boldsymbol{T} + \kappa \boldsymbol{T})^{-1} \boldsymbol{T} + \frac{1}{2} (\boldsymbol{T} + \kappa \boldsymbol{T})^{-1} X^t X (\boldsymbol{T} + \kappa \boldsymbol{T})^{-1}, \tag{3.18}$$

where $\frac{\partial}{\partial \boldsymbol{\kappa}} \log |\boldsymbol{A}| = \frac{1}{|\boldsymbol{A}|} |\boldsymbol{A}| \boldsymbol{A}^{-t}$ and $\frac{\partial}{\partial \boldsymbol{\kappa}} \boldsymbol{A}^{-1} = -\boldsymbol{A}^{-1} \frac{\partial A}{\partial \boldsymbol{\kappa}} \boldsymbol{A}^{-1}$. Equating equation 3.18 to zero leads to

$$\kappa \boldsymbol{T} = \boldsymbol{S} - \boldsymbol{T},$$

which is similar to the expression in 3.16 up to a sign which does not make difference when we ignore the sign (take the absolute) as we do in what follows. Taking $\|.\|_1$ on both sides which is the sum of the absolute of all entries of the matrix, we have

$$\|\kappa \boldsymbol{T}\|_1 = \|\boldsymbol{S} - \boldsymbol{T}\|_1 , \tag{3.19}$$

If target $\boldsymbol{T}$ is the identity matrix then the value of κ becomes

$$\kappa = \frac{\|\boldsymbol{S} - \boldsymbol{I}\|_1}{p}, \tag{3.20}$$

for AR(1) covariance structure as a target κ is

$$\kappa = \frac{\|\boldsymbol{S} - \boldsymbol{T}\|_1}{p + 2\sum_{k=0}^{p-1} k t^{p-k}}, \tag{3.21}$$

furthermore, if target $\boldsymbol{T}$ is the exchangeable covariance structure then

$$\kappa = \frac{\|\boldsymbol{S} - \boldsymbol{T}\|_1}{p + p(p-1)t}. \tag{3.22}$$

In order to restrict the range of the regularization parameter between zero and one, it is more appropriate to use the correlation scale rather than the covariance scale in equation 3.19. The correlation matrix can be obtained as

$$\hat{\boldsymbol{R}} = \hat{\boldsymbol{\Sigma}}_{\boldsymbol{d}}^{-\frac{1}{2}} \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Sigma}}_{\boldsymbol{d}}^{-\frac{1}{2}},$$

where $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{d}}$ is the diagonal matrix with corresponding diagonal elements of $\hat{\boldsymbol{\Sigma}}$. To get the regularization parameter at correlation scale we follow Warton (2008) who rather regularized the correlation matrix (not the covariance matrix) and rescale κ as

$$\gamma = \frac{1}{1+\kappa}, \tag{3.23}$$

where $\gamma \in [0, 1]$. The corresponding regularized estimator of the correlation matrix can be obtained as

$$\boldsymbol{R}_\gamma = \gamma \hat{\boldsymbol{R}} + (1-\gamma)\boldsymbol{T}. \tag{3.24}$$

3.3.1 Sample properties of γ

We present some useful properties of the penalty term γ. First, for a fixed p as n increases γ on the average decreases which indicates that the regularized estimator, $\mathbf{\Sigma}_\gamma$, is the consistent estimator of the true covariance matrix. In contrast, as p increases the penalty term also decreases and give more weight to the target matrix. Second, for a fixed value of p as we n increases the variance of γ tends to decrease.

For making the above properties of γ clear, we conduct a simulation study. We draw a sample of size $n = \{10, 30, 300\}$ from multivariate normal distribution with $p = \{10, 30, 100\}$ and mean vector, $\boldsymbol{\mu} = \mathbf{0}$, and three types of $\mathbf{\Sigma}$, that is, the first order auto-regressive AR(1) covariance structure, exchangeable covariance structure and identity matrix. For both AR(1) and exchangeable covariance structures we consider $t = 0.5$ and For each combination of n and p we simulate the data 1000 times and compute penalty parameter γ for each combination of n and p. Note that we use three types of targets mentioned in section 3.3 in case of AR(1) and exchangeable covariance structures as a target, γ depends on the correlation parameter t, which we estimate using 3.9 and 3.13 also note that the the penalty term changes with changing the target matrix. The aforementioned properties can be clearly seen in Figure 3.1.

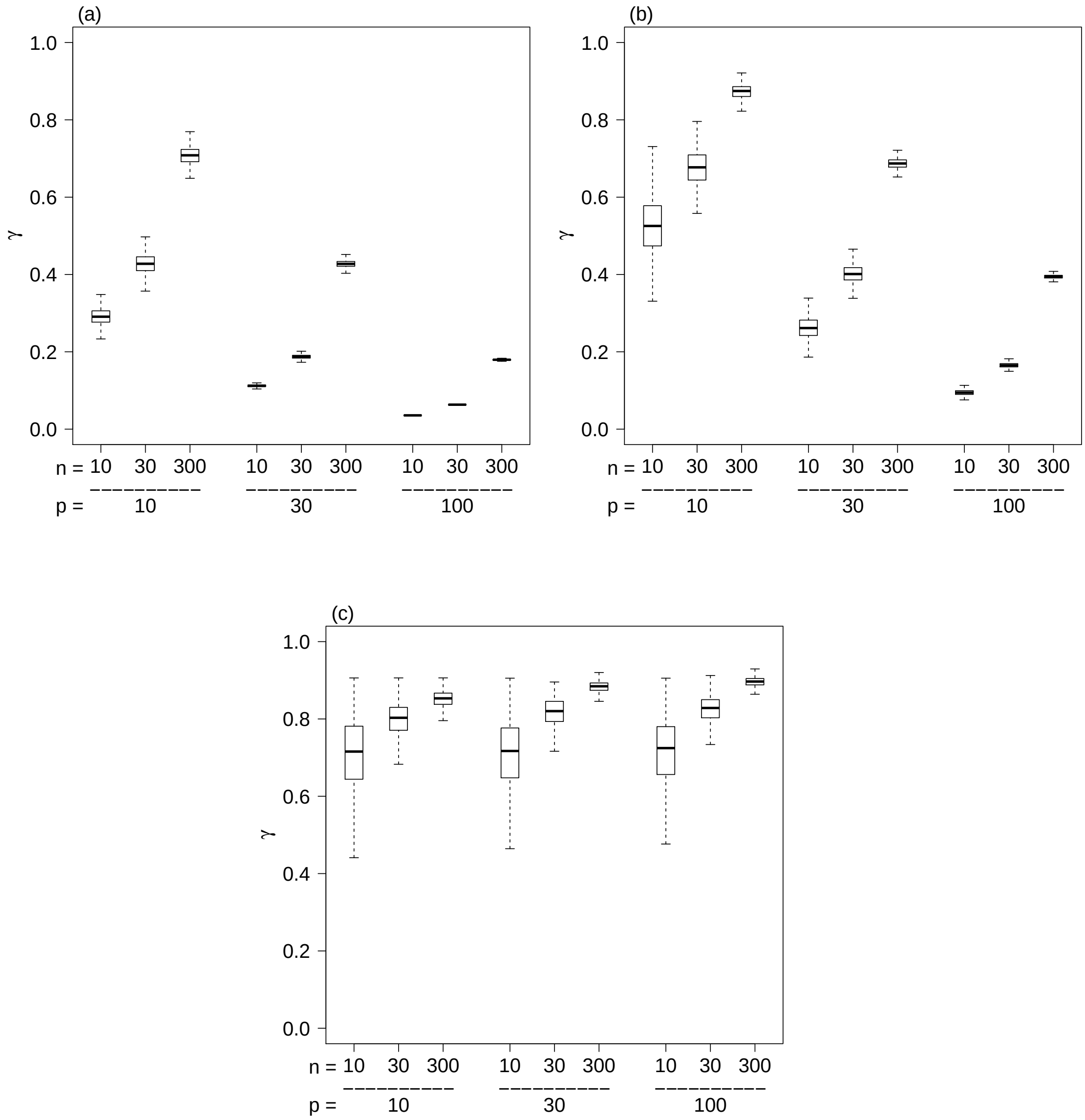

FIGURE 3.1: Distribution of γ values for different samples of size $n = \{10, 30, 300\}$ from a multivariate normal distribution with $p = \{10, 30, 100\}$ for different choices of $\boldsymbol{\Sigma}$ simulated 1000 times. (a) identity matrix as a true covariance matrix and as target (b) AR(1) structure as a true covariance matrix and as a target (c) exchangeable structure as a true covariance matrix and as a target.

Chapter 4

Simulation study

4.1 Introduction

In this chapter, we conduct extensive simulation study to numerically demonstrate the performance of the proposed method and also compare it with the shrinkage method proposed by Schäfer & Strimmer (2005) which is also implemented in R package "corpcor" (Schaefer et al., 2013).

4.2 Synthetic experiments

To examine the behavior of the proposed method under different simulation settings, we generate various datasets taking into account a number of different parameters. These parameters include varying sample sizes, number of variables, and different covariance structures.

We draw a sample of size n from a p-variate normal distribution with mean vector, $\boldsymbol{\mu} = \mathbf{0}$, and covariance matrix, $\boldsymbol{\Sigma}$. In order to evaluate the performance in a variety of situations, we consider three different covariance structures for $\boldsymbol{\Sigma}$. These include AR(1) covariance structure given by

$$\sigma_{ij} = t^{|i-j|} \quad \text{for} \quad 1 \leq i, j \leq p \quad \& \quad t \in [0, 1],$$

the exchangeable coavraince structure given by

$$\sigma_{ij} = \begin{cases} 1 & \text{when } i = j \\ t & \text{when } i \neq j \end{cases} \quad \text{for} \quad 1 \leq i, j \leq p \quad \& \quad t \in [0, 1],$$

and the covariance matrix generated by the algorithm presented in Schäfer & Strimmer (2004), which we will refer to random structure in the rest of the thesis. The random covariance structure is guaranteed to be a positive definite and allows to control for the number of zeros and the non-zeros entries in the off-diagonal positions of the inverse covariance matrix. Note that the off-diagonal entries of the inverse covariance matrix are the partial covariances and are interpretable in the context of Gaussian graphical models (Dempster, 1972). The algorithm to generate this covariance matrix is as follows:

- Start with an empty $p \times p$ matrix.
- Select randomly a suitable number of off-diagonal positions and fill it with random numbers drawn from uniform distribution between -1 and 1.
- Set the diagonal elements equal to the absolute sum of the columns of matrix generated in step 2 plus a small positive constant to ensure positive definiteness. This gives us the inverse covariance matrix.
- The inverse of the matrix obtained in step-3 is the desired covariance matrix.

For instance, to generate a coavriance matrix whose inverse is sparse we fill only a small proportion of non-zero random numbers in the off-diagonal positions of the inverse covariance matrix. On the other hand filling all the off-diagonal positions with non-zero entries will result in a covariance matrix whose inverse is dense. Furthermore, the inverse of the AR(1) covariance structure is spares and the inverse of the exchangeable covariance structure is dense. These three covariance structures allows to test the method in a range of situations.

We use the following three diffenet matrics to compare the accuracy of the proposed method with the maximum likelihood estimate and the shrinkage method of Schäfer & Strimmer (2005):

1. Sum of absolute errors in estimated eigenvalues.

2. Sum of element-wise squared errors of the estimated covariance matrices.

3. Visual comparison of estimated and true eigenvalues.

In our first type of experiments, we show the results for $n = 50$ and $p = 30, 50, 100$ to demonstrate the effect of increasing number of variables for a fixed value of n. The data is simulated from multivariate normal distribution using all three covariance structures mentioned above. For AR(1) and exchangeable covariance structures, we shrink the estimated covariance matrix towards the correct targets that are, respectively, AR(1) and exchangeable. We also examine the performance in which case the target is incorrectly specified as AR(1) and exchangeable while the true covariance matrix is identity. However, for random covariance structure we use only identity matrix as a target, which although is incorrect but have been used extensively as a shrinkage target to regularize the covariance matrix Ledoit & Wolf (2003); Schäfer & Strimmer (2005). Note that although we have conducted the experiments for a range of values of t, we show here the results only for $t = 0.5$ for both AR(1) and exchangeable structures. Similarly, for random covariance structure we show results for a covariance matrix whose inverse contains 30% of the off-diagonal positions as being non-zero.

The covariance matrix is estimated using the proposed method and shrinkage method of Schäfer & Strimmer (2005). For the proposed method whenever the target is correctly specified we estimate t using Gaussian estimating equations as described in chapter 3. Note that the target is correctly specified (but not always) only when we use AR(1) and exchangeable covariance structures. The sum of absolute errors in estimated eigenvalues averaged over 1000 simulated datasets are presented in Figure 4.1. The eigenvalues of sample covariance are also plotted not only for comparison purpose but also as a warning that how much our analysis can be unreliable under a high-dimensional setting.

Results:

From the simulation results, it is clear that whenever the target is correctly specified in case of both AR(1) and exchangeable covariance structures, the proposed method performs better than the shrinkage method as it maintains the smallest estimation error and is also much precise than the competing methods. Moreover, as p increases the estimates obtained by using the proposed method becomes more accurate and precise. Its performance becomes slightly weaker than the shrinkage method if the target is incorrectly specified as AR(1) or exchangeable while the true true covariance matrix is identity matrix. In case of random covariance structure when we use the identity matrix as a target, our proposed method also outperform than the shrinkage method in terms of accuracy and precision.

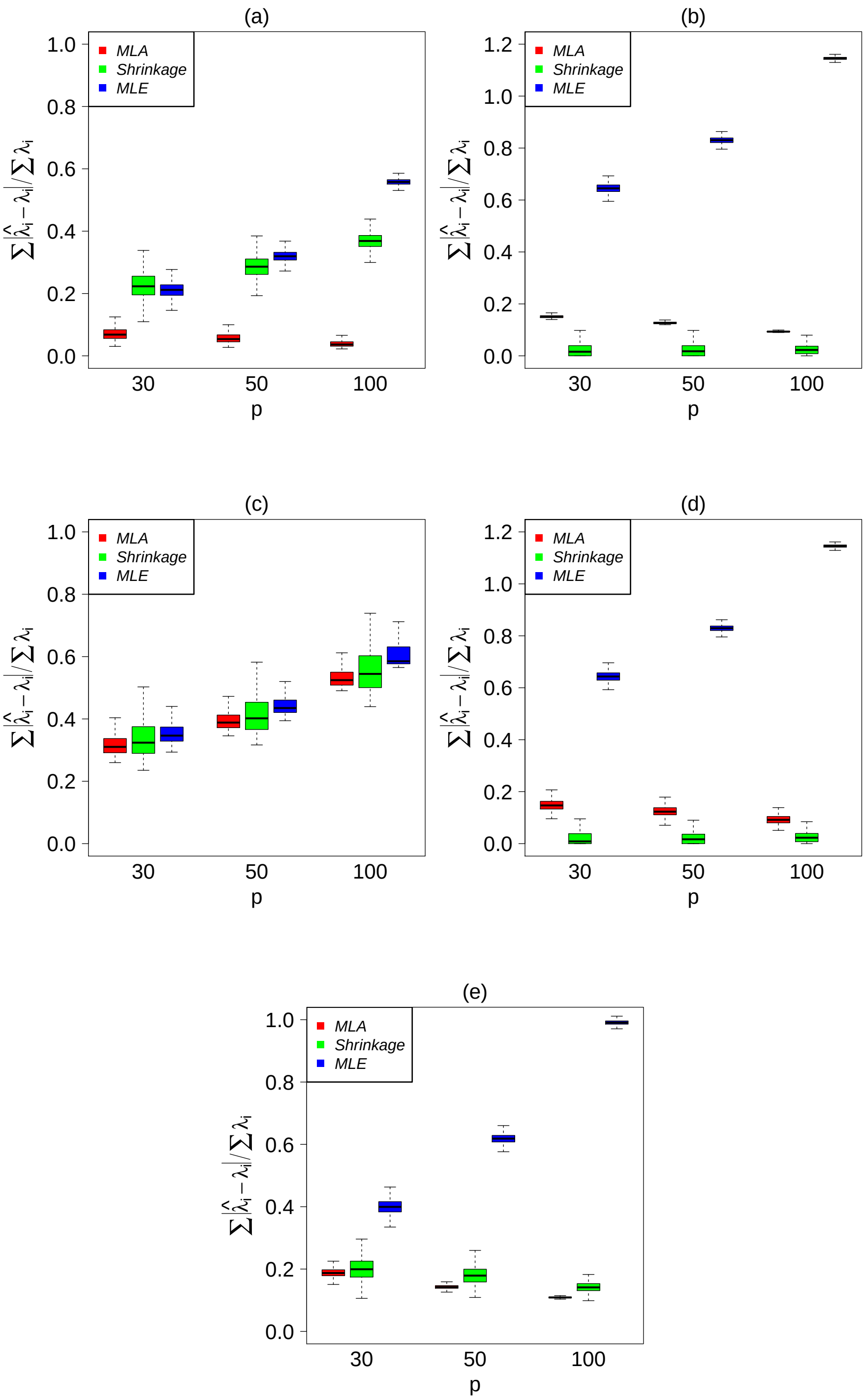

FIGURE 4.1: Distribution of the sum of absolute errors in estimated eigenvalues simulated 1000 times using proposed, shrinkage and maximum likelihood methods under different choices of covariance matrices (a, b) AR(1) with $t = 0.5$ and identity as true covariance matrix for all three methods and AR(1) as a target for the proposed method (c, d) exchangeable with $t = 0.5$ and identity as true covariance matrix for all three methods and exchangeable as a target for the proposed method (e) random covariance structure with 30% off-diagonal entries as non-zero and identity matrix as a target. The data are generated from multivariate normal distribution with $n = 50$ and $p = \{30, 50, 100\}$.

For the second type of experiments, we show the results for increasing value of $n = 10, 20, 30, 40, 50, 60, 70, 90, 120, 200$ while keeping $p = 10$. In this case we show the asymptotic performance of the competing methods. We use all three aforementioned covariance structures and simulate the data from multivariate normal distribution as we did in the previous case. For this case, we also consider $t = 0.5$ for both AR(1) and exchangeable covariance structures and for random covariance structure we take 30% non-zero off-diagonal elements. When the true covariance matrices are AR(1) and exchangeable we shrink the sample covariance matrix, respectively, towards AR(1) and exchangeable targets, which are correct targets for AR(1) and exchangeable covariance structures. For identity as a true covariance matrix we also use both AR(1) and exchangeable targets to shrink the sample covariance matrix towards them, which is incorrect. In case of random covariance structure we shrink the sample covariance matrix towards the identity matrix (commonly used target to regularized sample covariance matrix whenever $p \gg n$). We then calculate sum of element-wise squared errors (denoted by MSE in the rest of the thesis) of the estimated covariance matrices given by $\left\|\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}\right\|_F^2$, where $\|.\|_F^2$ denotes the sum of element-wise squared errors also known as squared Frobenius norm. Figure 4.2 shows the MSE of three different methods averaged over 1000 simulations.

Results:

From the simulation results, it is clear that the best estimator in terms of MSE is the one obtained from the proposed method when the target is correctly specified in case of AR(1) and exchangeable covariance structures. However, as can be expected, the performance of the other methods become similar when the sample size is very large. However, the proposed method perform slightly poor than the shrinkage method when the target is incorrectly specified as AR(1) or exchangeable while the true covariance matrix is identity matrix. For the random covariance structure the proposed method has minimum mean squared error than the shrinkage method when $n \approx p$, but as n increases its mean squared error is also increases indicating that the proposed estimator is asymptotically more biased (the case when the maximum likelihood estimate is valid).

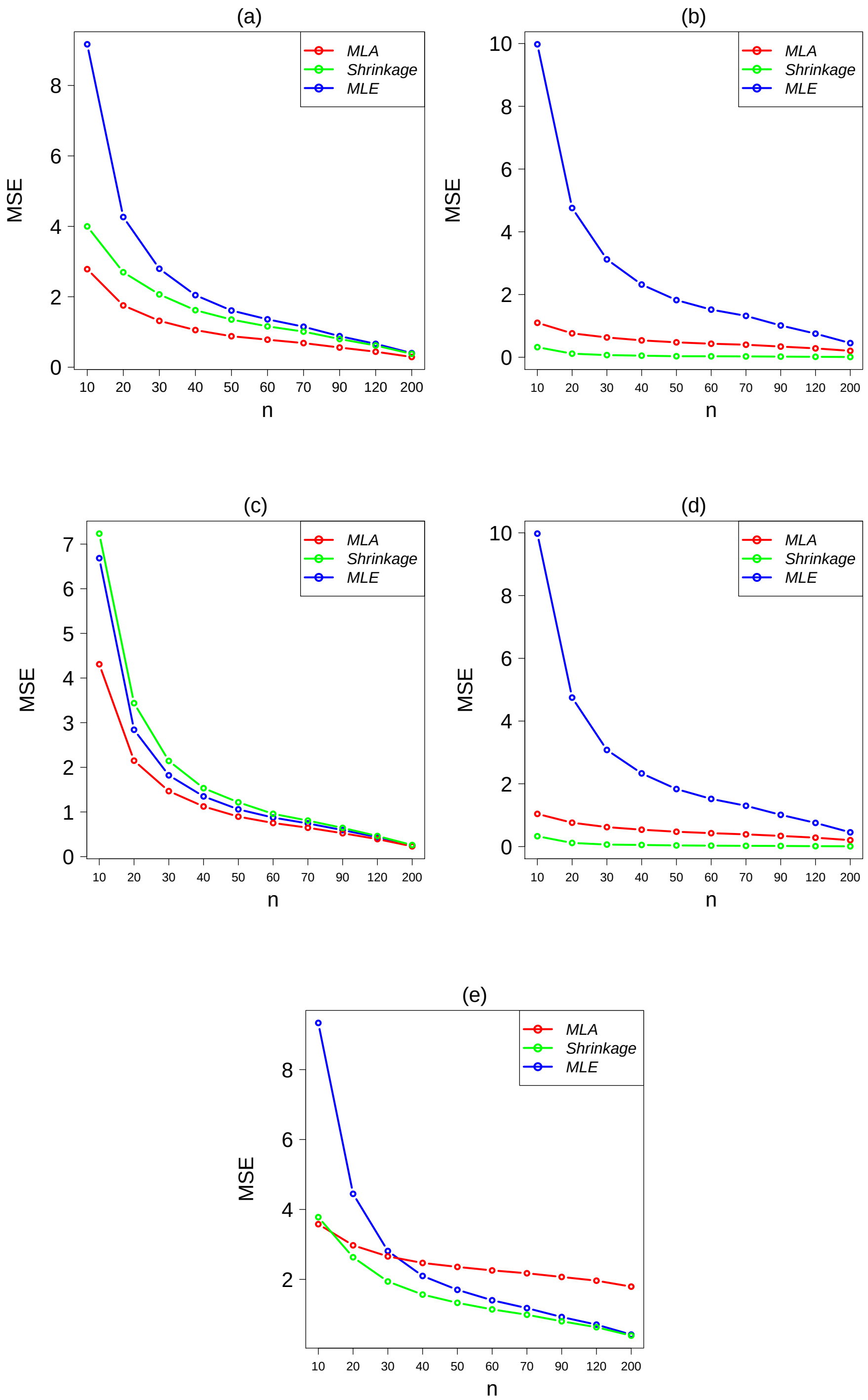

FIGURE 4.2: Comparison of MSE averaged over 1000 simulations of the estimated covariance matrices. The data are drawn from multivariate normal distribution with sample size $n = \{10, 20, 30, 40, 50, 60, 70, 90, 120, 200\}$ and $p = 10$ under different choices of covariance matrices (a, b) AR(1) with $t = 0.5$ and identity as true covariance matrix for all three methods and AR(1) as a target for the proposed method (c, d) exchangeable with $t = 0.5$ and identity as true covariance matrix for all three methods and exchangeable as a target for the proposed method (e) random covariance structure with 30% off-diagonal entries as non-zero and identity matrix as a target.

In the previous case, we demonstrated the estimation error in eigenvalues. Here we visually compare the estimated eigenvalues with the true eigenvalue. Although we have conducted a range of simulation experiments, we show the results for $n = 30$ and $p = 50$ and generate the data from multivariate normal distribution using all three covariance structures mentioned above. For AR(1) and exchangeable covariance structures we consider $t = 0.5$ and $t = 0.3$, respectively. For random covariance structure we take 30% of the off-diagonal elements as being non-zero. Furthermore, when the true covariance matrices are AR(1) and exchangeable we shrink the sample covariance matrix, respectively, towards AR(1) and exchangeable targets, which are correct targets for AR(1) and exchangeable covariance structures. For identity as a true covariance matrix we also use both AR(1) and exchangeable targets to shrink the sample covariance matrix towards them, which are incorrect targets. However, for random covariance structure we use only identity matrix as a target. We then estimate the covariance matrix using the proposed and shrinkage methods and calculate the eigenvalues of the two competing estimators along with the eigenvalues of sample covariance and true covariance matrix (gold standard). These eigenvalues are presented in Figure 4.3 averaged over 1000 simulations for all three the estimated covariance matrices.

Results:

From Figure 4.3 it can be seen that the sample eigenvalues are highly inaccurate that is the large eigenvalues overestimated and the small eigenvalues are under estimated. The proposed method and shrinkage method of Schäfer & Strimmer (2005) overcome this problem. But the eigenvalues of the proposed method recover the true eigenvalues more accurately compare to the shrinkage method of Schäfer & Strimmer (2005) if the target is correctly specified in case of AR(1) and exchangeable covariance structures. When the target is incorrectly specified in that case the eigenvalues estimated using shrinkage method are slightly closer to the true eigenvalues as compare to the proposed method. For random covariance structure the eigenvalues obtained from the proposed estimator are more accurate.

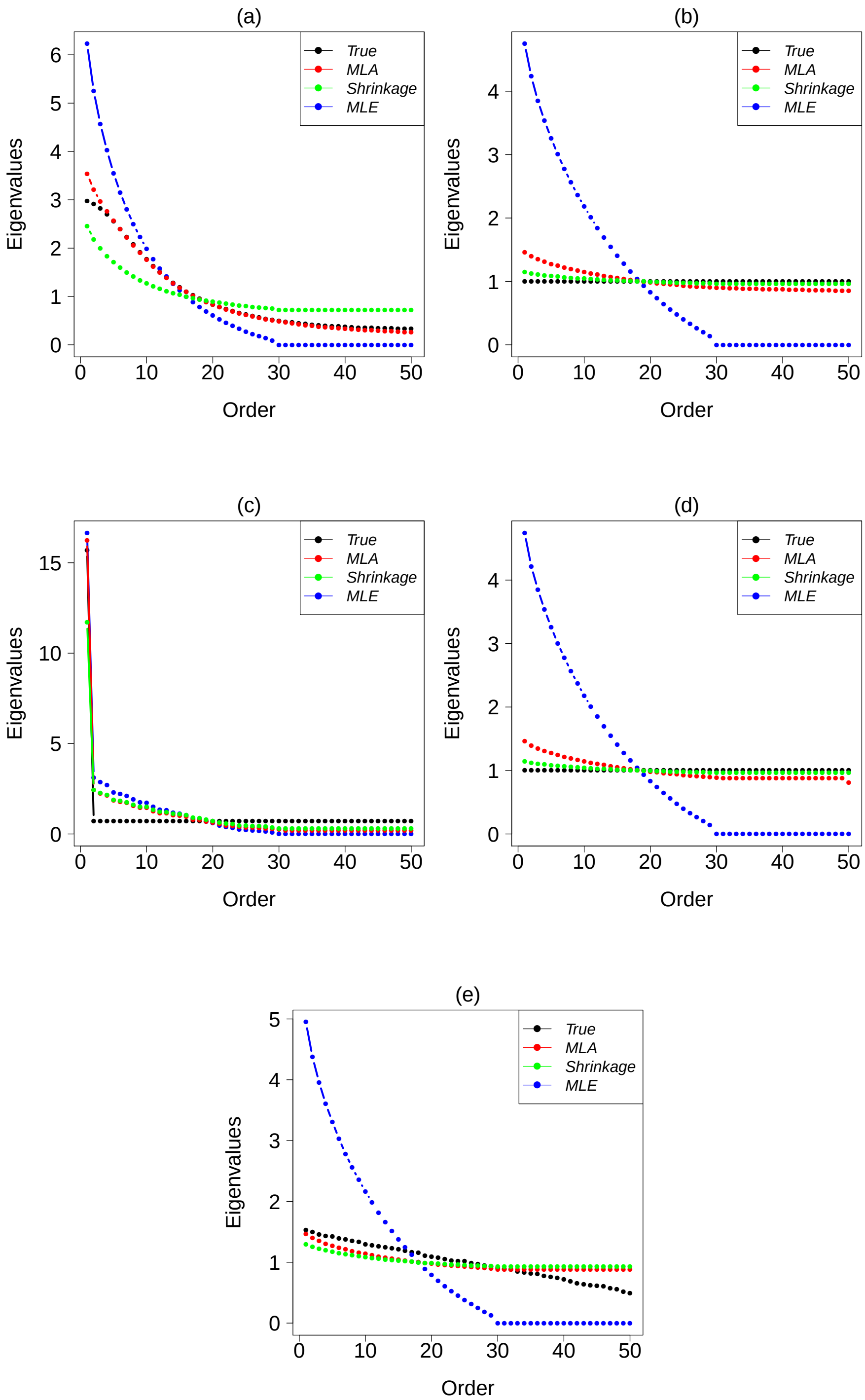

FIGURE 4.3: Comparison of estimated and true eigenvalues with $n = 30$ and $p = 50$ averaged over 1000 simulation. The data are drawn from multivariate normal distribution under different choices of covariance matrices (a, b) AR(1) with $t = 0.5$ and identity as true covariance matrix for all three methods and AR(1) as a target for the proposed method (c, d) exchangeable with $t = 0.3$ and identity as true covariance matrix for all three methods and exchangeable as a target for the proposed method (e) random covariance structure with 30% off-diagonal entries as non-zero and identity matrix as a target.

Chapter 5

Conclusion

In high-dimensional data where sample size is smaller than the number of variables, the sample estimator of the covariance matrix is not invertible and contains a large amount of estimation error. In these situations the classical multivariate techniques, which rely on the covariance matrix or its inverse, either fails to work or becomes unreliable (when n is larger but comparable to p). To overcome this problem various methods have been proposed in the literature to improve upon the sample estimator such as Moore-Penrose generalized inverse and other shrinkage estimation methods.

In this study a new regularized method of the covariance matrix is developed, which is also like shrinkage estimation is the linear convex combination of the sample covariance matrix and a target matrix. This new regularized estimator depends on the penalty parameter whose value need to be chosen in the appropriate range of values. The value of the penalty parameter is achieved by maximizing the log-likelihood function of the multivariate normal distribution. The proposed estimator is not only invertible but also well-conditioned. Furthermore, two new more informative targets have been used to shrink the sample covariance matrix towards them. These targets are the AR(1) and exchangeable covariance structures, which depends on the correlation parameter and need to be estimated in the appropriate range of values. To choose an appropriate value of the correlation parameter maximum log-likelihood function of the multivariate normal distribution is used.

The behaviour of the proposed method compare to the shrinkage method is explored through large simulations. The simulation experiments show that the proposed estimator perform better than the shrinkage estimator whenever the target is correctly specified in case of AR(1) and exchangeable covariance structures. However, its performance becomes slightly weaker than the shrinkage estimator when the target is incorrectly specified as AR(1) or exchangeable while the true covariance matrix is identity matrix. In case of random covariance structure our proposed estimator perform better than the shrinkage estimator.

It is worth noticing that the proposed estimator is also analytically much simpler and computationally inexpensive procedure compare to the shrinkage method.

Appendix A

R codes for Figures

Functions to be used in some Figures R codes.

```
t.ar1 <-  function(x) { # The function that estimate "t" in case of AR(1)
  n <- nrow(x)           # covariance structure given in equation 3.19.
  p <- ncol(x)           # We will mention it in a comment wherever it is
  cor.psi <- cor(x)      # required
  sum <- 0
  for(t in 2:p){
    sum <- sum + cor.psi[t,t-1]
  }
  hat.t <- sum/(p-1)
}

t.exch <-  function(x) { # The function that estimate "t" in case of
  n <- nrow(x)            # exchangeable covariance structure given in
  p <- ncol(x)            # equation 3.13. We will mention it in a
  cc <- cor(x)            # comment wherever it is required
  sum <- 0
  for(i in 1:(p-1)){
    for(j in (i+1):p){
      sum <- sum + cc[i,j]
    }
  }
  hat.t <- 2*sum/(p*(p-1))
}

un.str.cov <- function(n.var, prop.non.zero, const){ # The function to compute the
  AA <- matrix(0, n.var, n.var)      # random covariance structure of Schafer &
  SS <- matrix(0, n.var, n.var)      # Strimmmer (2004) We will mention it in a
                                     # comment wherever it is required
  AA[upper.tri(AA)] <- c(rbinom(n.var*(n.var-1)/2,1,prop.non.zero)) # proportion of non-zero elements

  BB <- AA + t(AA)
  for (i in 1:(n.var-1)){
    for (j in (i+1):n.var){
      if (AA[i,j]==1){
        SS[i,j] <- runif(1,-1,1) # Replace 1s with random values from the uniform distribution
      }
    }
  }
```

```
  
  SS <- SS + t(SS)
  ABS.SS <- abs(SS)
  ColSum <- apply(ABS.SS, 2, sum)

  for (i in 1:n.var){
    SS[i,i] <- ColSum[i] + const # fill diagonal entries with column sum plus small
      positive constant
  }

  cov2cor(SS)
}
```

Figure 4.1.

```
library(MASS) # Library "Modern Applied Statistics with S" abbreviated as MASS must be
    installed before running these R codes and will be used for all Figures R codes.

# Different sample sizes "n" and fixed "p"
n <- c(25,50,100,1000)
p <- 50

# Null matrix needed for results inside for loop to be used for further analysis
mat <- matrix(NA, nrow=length(n), ncol=p)

# True covariance and its eigenvalues
sigma <- diag(1, nrow = p, ncol = p)
e.tre <- eigen(sigma)$values

# Calculating sample covariance matrix eigenvalues for different samples "n"
for(i in 1:length(n)){
  # Repeat the results 1000 times. We will repeat the results for all Figures in R
    codes.
  res <- replicate(1000, {
  # Generating "n * p" X matrix from multivariate normal distribution
  x <- mvrnorm(n=n[i], mu=rep(0,p), Sigma=sigma)
  # Sample covarince matrix and its eigenvalues
  S <- cov(x)
  S.eigenvalues <- eigen(S)$values
  })
  # Averaging eigenvalues repeated 1000 times
  mat[i,] <- apply(res,1,mean)
}
# y limit for plot
yl <- max(rbind(mat,e.tre))

pdf(file = "screeplot.pdf")
# plotting true and sample eigenvalues
plot(e.tre,type="b",lwd=3,pch=16, ylim = c(0,yl), xlab="Order", ylab = "Eigenvalues")
points(mat[1,], type="b", lwd=3,pch=16 ,col="red")
points(mat[2,], type="b", lwd=3,pch=16, col="blue")
points(mat[3,], type="b", lwd=3, pch=16,col="5")
points(mat[4,], type="b", lwd=3, pch=16,col="green")

legend("topright",inset = 0.03, legend = c("True","n= 25","n= 50","n= 100","n= 1000"),
    cex = 1,box.lwd = 2,box.lty = 2,text.font = 4, bg="gray",col = c("black","red","
    blue","5","green"), lty = 1,  pch = 16)

dev.off()
```

Figure (3.1)a

```
library(MASS)

# Various "n" and "p"
n <- c(10, 30, 300)
p <- c(10, 30, 100)

# Null matrix (gamma matrix) needed for gamma values inside for loops to be used for
    futher analysis
gamma <- matrix(NA, length(n), length(p))
res<- replicate(1000,{
  for(i in 1:length(p)){
    for(j in 1:length(n)){
      # Identity matrix as a true covariance matrix
      sigma <- diag(p[i])
      x <- mvrnorm(n=n[j], mu= rep(0,p[i]), Sigma=sigma)

      # Calculate gamma values and store them in null matrix
      gamma[i,j] <- 1/(1+sum(abs(cor(x) - diag(p[i])))/(p[i]))
    }
  }
  gamma
})

# Make the resulting gamma matrix as a vector repeated 1000 times
vec <- as.vector(res)
# Boxes positions in the boxplot
arr <- as.vector(array(c(1,4,7,2,5,8,3,6,9),dim = c(3,3,1000)))

pdf(file="boxId.pdf")
# Margin
par(mar=c(7,5,1.5,1))
boxplot(vec~arr, outline=FALSE, ylab= expression(gamma), ylim=c(0,1), las=1, xaxt="n",
    at=c(1,3,5,8,10,12,15,17,19),cex.axis=1.5, cex.lab=1.5)
axis(1, at=c(1,3,5,8,10,12,15,17,19), labels = rep(c(10, 30, 300),3),cex.axis=1.5)
mtext("(a)",at=0.5, line = 0.2, cex= 1.5)
mtext("n =",at=-0.4, line = -28.6, cex= 1.5)
mtext("--",at= 1:5, line = -30, cex= 1.5)
mtext("--",at= 8:12, line = -30, cex= 1.5)
mtext("--",at= 15:19, line = -30, cex= 1.5)
mtext("p =",at= -0.4, line = -31, cex= 1.5)
mtext("10",at= 3, line = -31, cex= 1.5)
mtext("30",at= 10, line = -31, cex= 1.5)
mtext("100",at= 17, line = -31, cex= 1.5)

dev.off()
```

Figure (3.1)b

```
library(MASS)
    # t.ar1 function is required here to estimate "t"

n <- c(10, 30, 300)
p <- c(10, 30, 100)
t <- 0.5 # True t value for AR(1) covariance structure as a true covariance matrix

gamma <- matrix(NA, 3,3)
res <- replicate(1000,{

  for(i in 1:length(p)){
    for(j in 1:length(n)){
```

```
      sigma <-  t ^ outer(1:p[i], 1:p[i], function(aa, bb) abs(aa - bb))
      x <- mvrnorm(n=n[j], mu= rep(0,p[i]), Sigma=sigma)
      # Estimate t using t.ar1() function
      t.hat <- t.ar1(x)
      # Estimate true covariance matrix
      AR1.hat <- t.hat ^ outer(1:p[i], 1:p[i], function(aa, bb) abs(aa - bb))
      # Calculate gamma values see section 3.3
      k <- seq(0,p[i]-1)
      gamma[i,j] <- 1/(1+sum(abs(cor(x) - AR1.hat))/(p[i]+sum(2*k*(t.hat)^(p[i]-k))))
    }
  }
  gamma
})
vec <- as.vector(res)
arr <- as.vector(array(c(1,4,7,2,5,8,3,6,9),dim = c(3,3,1000)))

pdf(file="boxAR.pdf")
par(mar=c(7,5,1.5,1))
boxplot(vec~arr, outline=FALSE, ylab= expression(gamma),
        ylim=c(0,1), las=1, xaxt="n", at=c(1,3,5,8,10,12,15,17,19),
        cex.axis=1.5, cex.lab=1.5)
axis(1, at=c(1,3,5,8,10,12,15,17,19), labels = rep(c(10, 30, 300),3),
     cex.axis=1.5)
mtext("(b)",at=0.5, line = 0.2, cex= 1.5)
mtext("n =",at=-0.4, line = -28.6, cex= 1.5)
mtext("--",at= 1:5, line = -30, cex= 1.5)
mtext("--",at= 8:12, line = -30, cex= 1.5)
mtext("--",at= 15:19, line = -30, cex= 1.5)
mtext("p =",at= -0.4, line = -31, cex= 1.5)
mtext("10",at= 3, line = -31, cex= 1.5)
mtext("30",at= 10, line = -31, cex= 1.5)
mtext("100",at= 17, line = -31, cex= 1.5)
dev.off()
```

Figure (3.1)c

```
library(MASS)
    # t.exch function is required here to estimate "t"

n <- c(10, 30, 100)
p <- c(10, 30, 80)
t <- 0.5 # True t value for exchangeable covariance structure as a true covariance matrix

gamma <- matrix(NA, 3,3)
res<- replicate(1000,{
  for(i in 1:length(p)){
    for(j in 1:length(n)){
      sigma <- matrix(t,p[i],p[i])
      diag(sigma) = 1
      x <- mvrnorm(n=n[j], mu= rep(0,p[i]), Sigma=sigma)
      # Estimate t value using t.exch() function
      t.hat <- t.exch(x)
      # Estimate true covariance matrix
      hat.exch <- matrix(t.hat,p[i],p[i])
      diag(hat.exch) = 1
      # Calculate gamma values see section 3.3
      gamma[i,j] <- 1/(1+sum(abs(cor(x) - hat.exch))/(p[i]+ (p[i]*(p[i]-1)*t.hat)))
    }
  }
  gamma
})
```

```r
vec <- as.vector(res)
arr <- as.vector(array(c(1,4,7,2,5,8,3,6,9),dim = c(3,3,1000)))

pdf(file="boxex.pdf")
par(mar=c(7,5,1.5,1))
boxplot(vec~arr, outline=FALSE, ylab= expression(gamma),
        ylim=c(0,1), las=1, xaxt="n", at=c(1,3,5,8,10,12,15,17,19),
        cex.axis=1.5, cex.lab=1.5)
axis(1, at=c(1,3,5,8,10,12,15,17,19), labels = rep(c(10, 30, 300),3),
     cex.axis=1.5)
mtext("(c)",at=0.5, line = 0.2, cex= 1.5)
mtext("n =",at=-0.4, line = -28.6, cex= 1.5)
mtext("--",at= 1:5, line = -30, cex= 1.5)
mtext("--",at= 8:12, line = -30, cex= 1.5)
mtext("--",at= 15:19, line = -30, cex= 1.5)
mtext("p =",at= -0.4, line = -31, cex= 1.5)
mtext("10",at= 3, line = -31, cex= 1.5)
mtext("30",at= 10, line = -31, cex= 1.5)
mtext("100",at= 17, line = -31, cex= 1.5)

dev.off()
```

Figure (4.1)a

```r
# Comparison of the proposed, shrinkage of Schafer & Strimmer (2005) and maximum
    likelihood methods sum of absolute errors in estimated eigenvalues.
library(MASS)
library(corpcor) # An R package "correlations and partial correlations" abbreviated as
    corpcor required for the method of shrinkage estimation of Schafer & Strimmer (2005
    ). We will use this library for the remaining all Figures R codes.
                 # t.ar1 function is required here to estimate "t"

p <- c(30, 50 , 100)
n <- 50
t <- 0.5

gamma <- rep(NA,length(p))
ALL.EIGEN <- matrix(NA, length(p), length(p)) # Null matrix for all three competing
    estimated covariance matrices eigenvalues to be stored in it and use it for further
     analysis

res <- replicate(1000,{
  for(i in 1:length(p)) {

  sigma <-  t ^ outer(1:p[i], 1:p[i], function(aa, bb) abs(aa - bb))
  e.tre <- eigen(sigma)$values
  x <- mvrnorm(n=n, mu= rep(0,p[i]), Sigma=sigma)
  t.hat <- t.ar1(x)
  # compute target matrix, i.e, AR(1)
  TAR.AR1 <- t.hat ^ outer(1:p[i], 1:p[i], function(aa, bb) abs(aa - bb))
  S <- cor(x)
  # Gamma values needed for the proposed estimator
  k <- seq(0,p[i]-1)
  gamma[i] <- 1/(1+sum(abs(S - TAR.AR1))/(p[i]+sum(2*k*(t.hat)^(p[i]-k))))
  # Proposed estimator
  sigma.gamma <- gamma[i]*S + (1-gamma[i])*TAR.AR1

  # Compute eigenvalues of all three competing estimators
  MLA.eigen_values <- eigen(sigma.gamma)$values
  shrink.eigen_values <- eigen(cor.shrink(x,verbose = FALSE))$values
  MLE.eigen_values <- eigen(S)$values

```

```
  # Compute sum of absolute errors in estimated eigenvalues of all three competing estimators
  sum.eigen.MLA <-  sum(abs(MLA.eigen_values - e.tre ))/sum(e.tre)
  sum.eigen.shrink <- sum(abs(shrink.eigen_values - e.tre ))/sum(e.tre)
  sum.eigen.MLE <- sum(abs(MLE.eigen_values - e.tre ))/sum(e.tre)

  # store the resulting sum of absolute errors in estimated eigenvalues of all three competing estimators
  ALL.EIGEN[,i] <- c(sum.eigen.MLA, sum.eigen.shrink, sum.eigen.MLE)
  }
  ALL.EIGEN
})

vec <- as.vector(res)
arr <- as.vector(array(c(1:9),dim = c(3,3,1000)))

pdf(file="FIG4_2a.pdf")
par(mar=c(6,7,2,1), mgp = c(4,1,0))
boxplot(vec~arr, outline=FALSE, ylab= expression(sum(abs(hat(lambda[i]) - lambda[i]))/sum(lambda[i])),
        ylim=c(0,1), las=2, xaxt="n",at=c(1,2,3,5,6,7,9,10,11),
        cex.axis=2, cex.lab=2, col=c("red","green", "blue"))
mtext("(a)", side = 3, line = 0.5, cex = 2)
mtext("30",at=2, line = -29, cex= 2)
mtext("50",at=6, line = -29, cex= 2)
mtext("100",at=10, line = -29, cex= 2)
mtext("p",at=6, line = -31, cex= 2)

legend("topleft", legend = c("MLA","Shrinkage","MLE"),cex = 1.5,box.lwd = 2,box.lty = 1,
       text.font = 3,col = c("red","green", "blue"), pch = 15)
dev.off()
```

Figure (4.1)b

```
# Comparison of the proposed, shrinkage of Schafer & Strimmer (2005) and maximum likelihood methods sum of absolute errors in estimated eigenvalues.
library(MASS)
library(corpcor)
       # t.ar1 function is required here to estimate "t"

p <- c(30, 50 , 100)
n <- 50
t <- 0.5

gamma <- rep(NA,length(p))
ALL.EIGEN <- matrix(NA, length(p), length(p))

res <- replicate(1000,{
  for(i in 1:length(p)) {
    sigma <- diag(p[i]) # Identity as a true covariance matrix
    e.tre <- eigen(sigma)$values
    x <- mvrnorm(n=n, mu= rep(0,p[i]), Sigma=sigma)
    # AR(1) as a target matrix
    # compute target matrix
    t.hat <- t.ar1(x)
    TAR.AR1 <- t.hat ^ outer(1:p[i], 1:p[i], function(aa, bb) abs(aa - bb))
    S <- cor(x)
    k <- seq(0,p[i]-1)
    # Gamma values for the proposed method
    gamma[i] <- 1/(1+sum(abs(S - TAR.AR1))/(p[i]+sum(2*k*(t.hat)^(p[i]-k))))
    # Compute proposed estimator
```

```
    sigma.gamma <- gamma[i]*cor(x) + (1-gamma[i])*TAR.AR1

    # Compute eigenvalues of all three competing estimators
    MLA.eigen_values <- eigen(sigma.gamma)$values
    shrink.eigen_values <- eigen(cor.shrink(x,verbose = FALSE))$values
    MLE.eigen_values <- eigen(S)$values

    # Compute sum of absolute errors in estimated eigenvalues of all three competing
     estimators
    sum.eigen.MLA <-  sum(abs(MLA.eigen_values - e.tre ))/sum(e.tre)
    sum.eigen.shrink <- sum(abs(shrink.eigen_values - e.tre ))/sum(e.tre)
    sum.eigen.MLE <- sum(abs(MLE.eigen_values - e.tre ))/sum(e.tre)

    ALL.EIGEN[,i] <- c(sum.eigen.MLA, sum.eigen.shrink, sum.eigen.MLE)

    }
  ALL.EIGEN
})

vec <- as.vector(res)
arr <- as.vector(array(c(1:9),dim = c(3,3,1000)))

pdf(file="FIG4_2b.pdf")
par(mar=c(6,7,2,1), mgp = c(4,1,0))
boxplot(vec~arr, outline=FALSE, ylab= expression(sum(abs(hat(lambda[i]) - lambda[i]))/
    sum(lambda[i])),
        ylim=c(0,1.2), las=2, xaxt="n",at=c(1,2,3,5,6,7,9,10,11),
        cex.axis=2, cex.lab=2, col=c("red","green", "blue"))
mtext("(b)", side = 3, line = 0.5, cex = 2)
mtext("30",at=2, line = -29, cex= 2)
mtext("50",at=6, line = -29, cex= 2)
mtext("100",at=10, line = -29, cex= 2)
mtext("p",at=6, line = -31, cex= 2)

legend("topleft", legend = c("MLA","Shrinkage","MLE"),cex = 1.5,box.lwd = 2,box.lty = 1
    ,
       text.font = 3,col = c("red","green", "blue"), pch = 15)
dev.off()
```

Figure (4.1)c

```
# Comparison of the proposed, shrinkage of Schafer & Strimmer (2005) and maximum
    likelihood methods sum of absolute errors in estimated eigenvalues.
library(MASS)
library(corpcor)
         # t.exch function is required here to estimate "t"

p <- c(30, 50 , 100)
n <- 50
t <- 0.5

gamma <- rep(NA,length(p))
ALL.EIGEN <- matrix(NA, length(p), length(p))

res <- replicate(1000,{

  for(i in 1:length(p)) {

    sigma <-  matrix(t, p[i], p[i])
    diag(sigma) = 1
    e.tre <- eigen(sigma)$values
    x <- mvrnorm(n=n, mu= rep(0,p[i]), Sigma=sigma)
```

```r
    # Exchangeable as a target matrix
    # compute target matrix
    t.hat <- t.exch(x)
    TAR.EXCH <- matrix(t.hat, p[i], p[i])
    diag(TAR.EXCH) = 1
    S <- cor(x)

    # Gamma values for the proposed method
    gamma[i] <- 1/(1+sum(abs(S - TAR.EXCH))/(p[i] + (p[i]*(p[i]-1)*t.hat)))
    # Compute proposed estimator
    sigma.gamma <- gamma[i]*S + (1-gamma[i])*TAR.EXCH

    # Compute eigenvalues of all three competing estimators
    MLA.eigen_values <- eigen(sigma.gamma)$values
    shrink.eigen_values <- eigen(cor.shrink(x,verbose = FALSE))$values
    MLE.eigen_values <- eigen(S)$values

    # Compute sum of absolute errors in estimated eigenvalues of all three competing estimators
    sum.eigen.MLA <-  sum(abs(MLA.eigen_values - e.tre ))/sum(e.tre)
    sum.eigen.shrink <- sum(abs(shrink.eigen_values - e.tre ))/sum(e.tre)
    sum.eigen.MLE <- sum(abs(MLE.eigen_values - e.tre ))/sum(e.tre)

    ALL.EIGEN[,i] <- c(sum.eigen.MLA, sum.eigen.shrink, sum.eigen.MLE)
  }
  ALL.EIGEN
})

vec <- as.vector(res)
arr <- as.vector(array(c(1:9),dim = c(3,3,1000)))

pdf(file="FIG4_2c.pdf")
par(mar=c(6,7,2,1), mgp = c(4,1,0))
boxplot(vec~arr, outline=FALSE, ylab= expression(sum(abs(hat(lambda[i]) - lambda[i]))/sum(lambda[i])),
       ylim=c(0,1), las=2, xaxt="n",at=c(1,2,3,5,6,7,9,10,11),
       cex.axis=2, cex.lab=2, col=c("red","green", "blue"))
mtext("(c)", side = 3, line = 0.5, cex = 2)
mtext("30",at=2, line = -29, cex= 2)
mtext("50",at=6, line = -29, cex= 2)
mtext("100",at=10, line = -29, cex= 2)
mtext("p",at=6, line = -31, cex= 2)

legend("topleft", legend = c("MLA","Shrinkage","MLE"),cex = 1.5,box.lwd = 2,box.lty = 1,
      text.font = 3,col = c("red","green", "blue"), pch = 15)
dev.off()
```

Figure (4.1)d

```r
# Comparison of the proposed, shrinkage of Schafer & Strimmer (2005) and maximum likelihood methods sum of absolute errors in estimated eigenvalues.
library(MASS)
library(corpcor)
          # t.exch function is required here to estimate "t"

p <- c(30, 50 , 100)
n <- 50
t <- 0.5

gamma <- rep(NA,length(p))
ALL.EIGEN <- matrix(NA, length(p), length(p))
```

```
res <- replicate(1000,{
  for(i in 1:length(p)) {
    # Identity as a true covariance matrix
    sigma <- diag(p[i])
    e.tre <- eigen(sigma)$values
    x <- mvrnorm(n=n, mu= rep(0,p[i]), Sigma=sigma)
    # Exchangeable as a target matrix
    # compute target matrix
    t.hat <- t.exch(x)
    TAR.EXCH <- matrix(t.hat, p[i], p[i])
    diag(TAR.EXCH) = 1
    S <- cor(x)
    # Gamma values for the proposed method
    gamma[i] <- 1/(1+sum(abs(S - TAR.EXCH))/(p[i] + (p[i]*(p[i]-1)*t.hat)))
    # Compute proposed estimator
    sigma.gamma <- gamma[i]*S + (1-gamma[i])*TAR.EXCH

    # Compute eigenvalues of all three competing estimators
    MLA.eigen_values <- eigen(sigma.gamma)$values
    shrink.eigen_values <- eigen(cor.shrink(x,verbose = FALSE))$values
    MLE.eigen_values <- eigen(S)$values

    # Compute sum of absolute errors in estimated eigenvalues of all three competing estimators
    sum.eigen.MLA <-  sum(abs(MLA.eigen_values - e.tre ))/sum(e.tre)
    sum.eigen.shrink <- sum(abs(shrink.eigen_values - e.tre ))/sum(e.tre)
    sum.eigen.MLE <- sum(abs(MLE.eigen_values - e.tre ))/sum(e.tre)

    ALL.EIGEN[,i] <- c(sum.eigen.MLA, sum.eigen.shrink, sum.eigen.MLE)
  }
  ALL.EIGEN
})


vec <- as.vector(res)
arr <- as.vector(array(c(1:9),dim = c(3,3,1000)))

pdf(file="FIG4_2d.pdf")
par(mar=c(6,7,2,1), mgp = c(4,1,0))
boxplot(vec~arr, outline=FALSE, ylab= expression(sum(abs(hat(lambda[i]) - lambda[i]))/sum(lambda[i])),
        ylim=c(0,1.2), las=2, xaxt="n",at=c(1,2,3,5,6,7,9,10,11),
        cex.axis=2, cex.lab=2, col=c("red","green", "blue"))
mtext("(d)", side = 3, line = 0.5, cex = 2)
mtext("30",at=2, line = -29, cex= 2)
mtext("50",at=6, line = -29, cex= 2)
mtext("100",at=10, line = -29, cex= 2)
mtext("p",at=6, line = -31, cex= 2)

legend("topleft", legend = c("MLA","Shrinkage","MLE"),cex = 1.5,box.lwd = 2,box.lty = 1,
       text.font = 3,col = c("red","green", "blue"), pch = 15)
dev.off()
```

Figure (4.1)e

```
# Comparison of the proposed, shrinkage of Schafer & Strimmer (2005) and maximum likelihood methods sum of absolute errors in estimated eigenvalues.
library(MASS)
library(corpcor)
           # Function of random covariance structure is required here
```

```
p <- n.var <- c(30, 50 , 100)
n <- 50

gamma <- rep(NA, length(p))
ALL.EIGEN <- matrix(NA, length(p), length(p))

res <- replicate(1000,{
  for(i in 1:length(p)) {
    prop.non.zero <- 0.3
    const <- 0.01
    # random covariance structure as a ture covariance matrix
    sigma <- un.str.cov(n.var[i], prop.non.zero, const)
    e.tre <- eigen(sigma)$values
    x <- mvrnorm(n=n, mu= rep(0,p[i]), Sigma=sigma)
    S <- cor(x)

    # Compute gamma values using Identity as a target
    gamma[i] <- 1/(1+sum(abs(S - diag(p[i])))/(p[i]))
    # Compute proposed estimator
    sigma.gamma <- gamma[i]*S + (1-gamma[i])*diag(p[i])

    # Compute eigenvalues of all three competing estimators
    MLA.eigen_values <- eigen(sigma.gamma)$values
    shrink.eigen_values <- eigen(cor.shrink(x,verbose = FALSE))$values
    MLE.eigen_values <- eigen(S)$values

    # Compute sum of absolute errors in estimated eigenvalues of all three competing
     estimators
    sum.eigen.MLA <-  sum(abs(MLA.eigen_values - e.tre ))/sum(e.tre)
    sum.eigen.shrink <- sum(abs(shrink.eigen_values - e.tre ))/sum(e.tre)
    sum.eigen.MLE <- sum(abs(MLE.eigen_values - e.tre ))/sum(e.tre)

    ALL.EIGEN[,i] <- c(sum.eigen.MLA, sum.eigen.shrink, sum.eigen.MLE)
  }
  ALL.EIGEN
})

vec <- as.vector(res)
arr <- as.vector(array(c(1:9),dim = c(3,3,1000)))

pdf(file="FIG4_2e.pdf")
par(mar=c(6,7,2,1), mgp = c(4,1,0))
boxplot(vec~arr, outline=FALSE, ylab= expression(sum(abs(hat(lambda[i]) - lambda[i]))/
    sum(lambda[i])),
        ylim=c(0,1), las=2, xaxt="n",at=c(1,2,3,5,6,7,9,10,11),
        cex.axis=2, cex.lab=2, col=c("red","green", "blue"))
mtext("(e)", side = 3, line = 0.5, cex = 2)
mtext("30",at=2, line = -29, cex= 2)
mtext("50",at=6, line = -29, cex= 2)
mtext("100",at=10, line = -29, cex= 2)
mtext("p",at=6, line = -31, cex= 2)

legend("topleft", legend = c("MLA","Shrinkage","MLE"),cex = 1.5,box.lwd = 2,box.lty = 1
    ,
       text.font = 3,col = c("red","green", "blue"), pch = 15)
dev.off()
```

Figure (4.2)a

```
# Comparison of MSE of all three competing procedures: Proposed, shrinkage and maximum
    likelihood     method
```

```
library(MASS)
library(corpcor)
              # t.ar1 function is required here to estimate "t"

p <- n.var <- 10
n <- c(10, 20,30,40,50,60,70,90,120,200)
t <- 0.5
sigma <-  t ^ outer(1:p, 1:p, function(aa, bb) abs(aa - bb)) # true covariance matrix
    AR(1)
gamma <- rep(NA, length(n))
MSE <- matrix(NA, 3, length(n)) # Null matrix for the MSE of all three competing
    procedures to be store in it and use for further analysis

res <- replicate(1000,{
  for(i in 1:length(n)) {
  x <- mvrnorm(n=n[i], mu= rep(0,p), Sigma=sigma)
  # Compute target matrix, i.e, AR(1)
  t.hat <- t.ar1(x)
  TAR.AR1 <- t.hat ^ outer(1:p, 1:p, function(aa, bb) abs(aa - bb))
  S <- cor(x)

  # Compute gamma for the proposed estimator
  k <- seq(0,p-1)
  gamma[i] <- 1/(1+sum(abs(S - TAR.AR1))/(p+sum(2*k*(t.hat)^(p-k))))
  # Compute proposed estimator of the covariance matrix
  sigma.gamma = gamma[i]*S + (1-gamma[i])*TAR.AR1
  # Compute shrinkage estimator of the covariance matrix
  sigma.shrink <- cor.shrink(x,verbose = FALSE)

  # Compute the MSE of all three estimators
  MSE_OF_MLE <- sum((S - sigma)^2)
  MSE_OF_SIGMA.GAMMA <- sum((sigma.gamma - sigma)^2)
  MSE_OF_SIGMA.SHRINK <- sum((sigma.shrink - sigma)^2)

  # Store the resulting MSE in the null matrix
  MSE[,i] <- c(MSE_OF_MLE,MSE_OF_SIGMA.GAMMA,MSE_OF_SIGMA.SHRINK)
  }
  MSE
})
# Average the MSE repeated 1000 times
ave_MSE <- apply(res, c(1,2), mean)

y.min <- min(ave_MSE)
y.max <- max(ave_MSE)

pdf(file="FIG4_4a.pdf")
par(mar=c(6,6,2,1), mgp = c(4,1,0))
plot(ave_MSE[1,], type="b",lwd=3,pch=1,ylim = c(y.min,y.max),
     cex.lab=2, ylab = "MSE", xlab = "",las=2, xaxt='n',
     cex.axis=2, col=20)
axis(1, at=c(1,2,3,4,5,6,7,8,9,10),
     labels = c(10, 20,30,40,50,60,70,90,120,200),
     cex.axis=1.3)
mtext("n",at=6, line = -31, cex= 2)
points(ave_MSE[2,], type="b", lwd=3,pch=1 ,col= "red")
points(ave_MSE[3,], type="b", lwd=3,pch=1 ,col= "green")
mtext("(a)", side = 3, line = 0.5, cex = 2)

legend("topright", legend = c("MLA", "Shrinkage", "MLE"),
       cex = 1.5,box.lwd = 1,box.lty = 1,
       text.font = 3,col = c("red", "green", 20),
       lwd=3, lty = 1, pch = 1)
dev.off()
```

Figure (4.2)b

```
# Comparison of MSE of all three competing procedures: Proposed, shrinkage and maximum
    likelihood   method
library(MASS)
library(corpcor)
              # t.ar1 function is required here to estimate "t"

p <- n.var <- 10
n <- c(10, 20,30,40,50,60,70,90,120,200)
t <- 0.5
sigma <- diag(p) # Identity as a true covariance matrix
gamma <- rep(NA, length(n))
MSE <- matrix(NA, 3, length(n))

res <- replicate(1000,{
  for(i in 1:length(n)) {
    x <- mvrnorm(n=n[i], mu= rep(0,p), Sigma=sigma)
    # Compute target matrix, i.e, AR(1)
    t.hat <- t.ar1(x)
    # Compute the target matrix, i.e, AR(1)
    TAR.AR1 <- t.hat ^ outer(1:p, 1:p, function(aa, bb) abs(aa - bb))
    S <- cor(x)

    k <- seq(0,p-1)
    # Compute gamma for the proposed estimator
    gamma[i] <- 1/(1+sum(abs(S - TAR.AR1))/(p+sum(2*k*(t.hat)^(p-k))))
    # Compute proposed estimator of the covariance matrix
    sigma.gamma = gamma[i]*S + (1-gamma[i])*TAR.AR1
    # Compute shrinkage estimator of the covariance matrix
    sigma.shrink <- cor.shrink(x,verbose = FALSE)

    # Compute the MSE of all three competing estimators
    MSE_OF_MLE <- sum((S - sigma)^2)
    MSE_OF_SIGMA.GAMMA <- sum((sigma.gamma - sigma)^2)
    MSE_OF_SIGMA.SHRINK <- sum((sigma.shrink - sigma)^2)

    # Store the resulting MSE in the null matrix
    MSE[,i] <-  c(MSE_OF_MLE,MSE_OF_SIGMA.GAMMA,MSE_OF_SIGMA.SHRINK)
  }
  MSE
})
# Average the MSE repeated 1000 times
ave_MSE <- apply(res, c(1,2), mean)
y.min <- min(ave_MSE)
y.max <- max(ave_MSE)

pdf(file="FIG4_4b.pdf")
par(mar=c(6,7,2,1), mgp = c(5,1,0))
plot(ave_MSE[1,], type="b",lwd=3,pch=1,ylim = c(y.min,y.max),
     cex.lab=2, ylab = "MSE", xlab = "",las=2, xaxt='n',
     cex.axis=2, col=20)
axis(1, at=c(1,2,3,4,5,6,7,8,9,10),
     labels = c(10, 20,30,40,50,60,70,90,120,200),
     cex.axis=1.2)
points(ave_MSE[2,], type="b", lwd=3,pch=1 ,col= "red")
points(ave_MSE[3,], type="b", lwd=3,pch=1 ,col= "green")
mtext("n",at=6, line = -31, cex= 2)
mtext("(b)", side = 3, line = 0.5, cex = 2)

legend("topright", legend = c("MLA", "Shrinkage", "MLE"),
        cex = 1.5,box.lwd = 1,box.lty = 1,
        text.font = 3,col = c("red", "green", 20),
        lwd=3, lty = 1, pch = 1)
dev.off()
```

Figure (4.2)c

```
# Comparison of MSE of all three competing procedures: Proposed, shrinkage and maximum
    likelihood    method
library(MASS)
library(corpcor)
                # t.exch function is required here to estimate "t"

p <- n.var <- 10
n <- c(10, 20,30,40,50,60,70,90,120,200)
t <- 0.5
sigma <-  matrix(t, p, p) # Exchangeable covariance structure as a true covariance
    matrix
diag(sigma) = 1
gamma <- rep(NA, length(n))
MSE <- matrix(NA, 3, length(n))

res <- replicate(1000,{
  for(i in 1:length(n)) {
    x <- mvrnorm(n=n[i], mu= rep(0,p), Sigma=sigma)
    # Compute the target matrix, i.e, exchangeable
    t.hat <- t.exch(x)
    TAR.EXCH <- matrix(t.hat, p, p)
    diag(TAR.EXCH) = 1
    S <- cor(x)

    # Compute the gamma values for the proposed method
    gamma[i] <- 1/(1+sum(abs(S - TAR.EXCH))/(p + (p*(p-1)*t.hat)))
    # Compute proposed estimator of the covariance matrix
    sigma.gamma <- gamma[i]*S + (1-gamma[i])*TAR.EXCH
    # Compute shrinkage estimator of the covariance matrix
    sigma.shrink <- cor.shrink(x,verbose = FALSE)

    # Computing MSE for all three competing estimators
    MSE_OF_MLE <- sum((S - sigma)^2)
    MSE_OF_SIGMA.GAMMA <- sum((sigma.gamma - sigma)^2)
    MSE_OF_SIGMA.SHRINK <- sum((sigma.shrink - sigma)^2)

    MSE[,i] <- c(MSE_OF_MLE,MSE_OF_SIGMA.GAMMA,MSE_OF_SIGMA.SHRINK)
  }
  MSE
})
# Average the resulting MSE repeated 1000 times
ave_MSE <- apply(res, c(1,2), mean)
y.min <- min(ave_MSE)
y.max <- max(ave_MSE)

pdf(file="FIG4_4c.pdf")
par(mar=c(6,7,2,1), mgp = c(4,1,0))
plot(ave_MSE[1,], type="b",lwd=3,pch=1,ylim = c(y.min,y.max),
     cex.lab=2, ylab = "MSE", xlab = "",las=2, xaxt='n',
     cex.axis=2, col=20)
axis(1, at=c(1,2,3,4,5,6,7,8,9,10),
     labels = c(10, 20,30,40,50,60,70,90,120,200),
     cex.axis=1.2)
mtext("n",at=6, line = -31, cex= 2)
points(ave_MSE[2,], type="b", lwd=3,pch=1 ,col= "red")
points(ave_MSE[3,], type="b", lwd=3,pch=1 ,col= "green")
mtext("(c)", side = 3, line = 0.5, cex = 2)

```

```r
legend("topright", legend = c("MLA", "Shrinkage", "MLE"),
        cex = 1.5,box.lwd = 1,box.lty = 1,
        text.font = 3,col = c("red", "green", 20),
        lwd=3, lty = 1, pch = 1)
dev.off()
```

Figure (4.2)d

```r
# Comparison of MSE of all three competing procedures: Proposed, shrinkage and maximum
    likelihood      method
library(MASS)
library(corpcor)
             # t.exch function is required here to estimate "t"

p <- n.var <- 10
n <- c(10, 20,30,40,50,60,70,90,120,200)
t <- 0.5
sigma <-  diag(p) # Identity as a true covariance matrix
gamma <- rep(NA, length(n))
MSE <- matrix(NA, 3, length(n))

res <- replicate(1000,{
  for(i in 1:length(n)) {
    x <- mvrnorm(n=n[i], mu= rep(0,p), Sigma=sigma)
    # Compute target matrix, i.e, exchangeable
    t.hat <- t.exch(x)
    TAR.EXCH <- matrix(t.hat, p, p)
    diag(TAR.EXCH) = 1
    S <- cor(x)

    # Calculate gamma values for the proposed estimator
    gamma[i] <- 1/(1+sum(abs(S - TAR.EXCH))/(p + (p*(p-1)*t.hat)))
    # Calculate proposed and shrinkage estimator
    sigma.gamma <- gamma[i]*S + (1-gamma[i])*TAR.EXCH
    sigma.shrink <- cor.shrink(x,verbose = FALSE)

    # Calculate MSE of all three competing procedures
    MSE_OF_MLE <- sum((S - sigma)^2)
    MSE_OF_SIGMA.GAMMA <- sum((sigma.gamma - sigma)^2)
    MSE_OF_SIGMA.SHRINK <- sum((sigma.shrink - sigma)^2)

    MSE[,i] <- c(MSE_OF_MLE,MSE_OF_SIGMA.GAMMA,MSE_OF_SIGMA.SHRINK)
  }
  MSE
})
# Average the resulting MSE over 1000 simulations
ave_MSE <- apply(res, c(1,2), mean)
y.min <- min(ave_MSE)
y.max <- max(ave_MSE)

pdf(file="FIG4_4d.pdf")
par(mar=c(6,7,2,1), mgp = c(4,1,0))
plot(ave_MSE[1,], type="b",lwd=3,pch=1,ylim = c(y.min,y.max),
     cex.lab=2, ylab = "MSE", xlab = "",las=2, xaxt='n',
     cex.axis=2, col=20)
axis(1, at=c(1,2,3,4,5,6,7,8,9,10),
     labels = c(10, 20,30,40,50,60,70,90,120,200),
     cex.axis=1.2)
mtext("n",at=6, line = -31, cex= 2)
points(ave_MSE[2,], type="b", lwd=3,pch=1 ,col= "red")
points(ave_MSE[3,], type="b", lwd=3,pch=1 ,col= "green")
mtext("(d)", side = 3, line = 0.5, cex = 2)
```

```r

legend("topright", legend = c("MLA", "Shrinkage", "MLE"),
        cex = 1.5,box.lwd = 1,box.lty = 1,
        text.font = 3,col = c("red", "green", 20),
        lwd=3, lty = 1, pch = 1)
dev.off()
```

Figure (4.2)e

```r
# Comparison of MSE of all three competing procedures: Proposed, shrinkage and maximum likelihood method
library(MASS)
library(corpcor)
                    # Function of random covariance structure is required here

p <- n.var <- 10
n <- c(10, 20,30,40,50,60,70,90,120,200)

gamma <- rep(NA, length(n))
MSE <- matrix(NA, 3, length(n))

prop.non.zero <- 0.3
const <- 0.01
sigma <- un.str.cov(n.var, prop.non.zero, const) # Random covariance structure as a true covariance matrix

res <- replicate(1000,{
  for(i in 1:length(n)) {
    x <- mvrnorm(n=n[i], mu= rep(0,p), Sigma=sigma)
    S <- cor(x)

    # Calculate gamma values for the proposed estimator
    gamma[i] <- 1/(1+sum(abs(S - diag(p)))/(p))
    # Calculate proposed and shrinkage estimator
    sigma.gamma <- gamma[i]*S + (1-gamma[i])*diag(p)
    sigma.shrink <- cor.shrink(x,verbose = FALSE)

    # Calculate MSE of all three competing procedures
    MSE_OF_MLE <- sum((S - sigma)^2)
    MSE_OF_SIGMA.GAMMA <- sum((sigma.gamma - sigma)^2)
    MSE_OF_SIGMA.SHRINK <- sum((sigma.shrink - sigma)^2)

    MSE[,i] <-  c(MSE_OF_MLE,MSE_OF_SIGMA.GAMMA,MSE_OF_SIGMA.SHRINK)

  }
  MSE
})
# Average the resulting MSE over 1000 simulations
ave_MSE <- apply(res, c(1,2), mean)
y.min <- min(ave_MSE)
y.max <- max(ave_MSE)

pdf(file="FIG4_4e.pdf")
par(mar=c(6,7,2,1), mgp = c(4,1,0))
plot(ave_MSE[1,], type="b",lwd=3,pch=1,ylim = c(y.min,y.max),
      cex.lab=2, ylab = "MSE", xlab = "",las=2, xaxt='n',
      cex.axis=2, col=20)
axis(1, at=c(1,2,3,4,5,6,7,8,9,10),
      labels = c(10, 20,30,40,50,60,70,90,120,200),
      cex.axis=1.2)
mtext("n",at=6, line = -31, cex= 2)
points(ave_MSE[2,], type="b", lwd=3,pch=1 ,col= "red")
```

```
points(ave_MSE[3,], type="b", lwd=3,pch=1 ,col= "green")
mtext("(e)", side = 3, line = 0.5, cex = 2)

legend("topright", legend = c("MLA", "Shrinkage", "MLE"),
       cex = 1.5,box.lwd = 1,box.lty = 1,
       text.font = 3,col = c("red", "green", 20),
       lwd=3, lty = 1, pch = 1)
dev.off()
```

Figure (4.3)a

```
# Comparison of eigenvalues of the proposed, shrinkage and maximum likelihood method
    along with the eigenvalues of true covariance matrix
library(MASS)
library(corpcor)
                # t.ar1 function is required here to estimate "t"

p <- n.var <- 50
n <- 30
t <- 0.5
sigma <-  t ^ outer(1:p, 1:p, function(aa, bb) abs(aa - bb))
e.tre <- eigen(sigma)$values

eigen.matrix <- matrix(NA,3,n.var) # null matrix for the eigenvalues to be store in it.

res <- replicate(1000,{
  x <- mvrnorm(n=n, mu= rep(0,p), Sigma=sigma)
  # Calculate target matrix, i.e, AR(1)
  t.hat <- t.ar1(x)
  TAR.AR1 <- t.hat ^ outer(1:p, 1:p, function(aa, bb) abs(aa - bb))
  S <- cor(x)

  # Calculate gamma values for the proposed method
  k <- seq(0,p-1)
  gamma <- 1/(1+sum(abs(S - TAR.AR1))/(p+sum(2*k*(t.hat)^(p-k))))

  # Calculate the proposed estimator
  sigma.gamma = gamma*S + (1-gamma)*TAR.AR1

  # Calculate the eigenvalues of all three competing procedures
  MLA.eigen_values <- eigen(sigma.gamma)$values
  eigen_values.shrink <- eigen(cor.shrink(x,verbose = FALSE))$values
  MLE.eigen_values <- eigen(S)$values

  # Store the resulting eigenvalues in the null matrix
  eigen.matrix[1,] <- MLE.eigen_values
  eigen.matrix[2,] <- MLA.eigen_values
  eigen.matrix[3,] <- eigen_values.shrink
  eigen.matrix
})
# Average the resulting eigenvalues over 1000 simulations
ave_eigen <- apply(res, c(1,2), mean)
yl <- max(rbind(ave_eigen, e.tre))

pdf(file="FIG4_3a.pdf")
par(mar=c(7,6,2,1), mgp = c(4,1,0))
plot(e.tre, type="b",lwd=3,pch=16,ylim = c(0,yl), xlab="Order",
     ylab = "Eigenvalues", cex.axis=2, yaxt="n", cex.lab=2)
axis(2, las=2, cex.axis=2)
points(ave_eigen[1,], type="b", lwd=3,pch=16 ,col= "20")
points(ave_eigen[2,], type="b", lwd=3,pch=16 ,col= "red")
points(ave_eigen[3,], type="b", lwd=3,pch=16 ,col= "green")
```

```
mtext("(a)", side = 3, line = 0.5, cex = 2)

legend("topright", legend = c("True", "MLA", "Shrinkage", "MLE"),
       cex = 1.5,box.lwd = 1,box.lty = 1,
       text.font = 3,col = c("black","red", "green", 20),
       lty = 1, pch = 16)
dev.off()
```

Figure (4.3)b

```
library(MASS)
library(corpcor)
                 # t.ar1 function is required here to estimate "t"

p <- n.var <- 50
n <- 30
t <- 0.5

sigma <-  diag(p) # Identity as a true covariance matrix
e.tre <- eigen(sigma)$values
eigen.matrix <- matrix(NA,3,n.var)

res <- replicate(1000,{
  x <- mvrnorm(n=n, mu= rep(0,p), Sigma=sigma)
  # Calculate target matrix, i.e, AR(1)
  t.hat <- t.ar1(x)
  TAR.AR1 <- t.hat ^ outer(1:p, 1:p, function(aa, bb) abs(aa - bb))
  S <- cor(x)

  # Calculate gamma values for the proposed method
  k <- seq(0,p-1)
  gamma <- 1/(1+sum(abs(S - TAR.AR1))/(p+sum(2*k*(t.hat)^(p-k))))
  # Calculate the proposed estimator of the true covariance matrix
  sigma.gamma = gamma*S + (1-gamma)*TAR.AR1

  # Calculate the eigenvalues of all three proposed estimators
  MLA.eigen_values <- eigen(sigma.gamma)$values
  eigen_values.shrink <- eigen(cor.shrink(x,verbose = FALSE))$values
  MLE.eigen_values <- eigen(S)$values

  # Store the eigenvalues in the null matrix
  eigen.matrix[1,] <- MLE.eigen_values
  eigen.matrix[2,] <- MLA.eigen_values
  eigen.matrix[3,] <- eigen_values.shrink
  eigen.matrix
})
# Average the resulting eigenvalue repeated 1000 times
ave_eigen <- apply(res, c(1,2), mean)
yl <- max(rbind(ave_eigen, e.tre))

pdf(file="FIG4_3b.pdf")
par(mar=c(7,6,2,1), mgp = c(4,1,0))
plot(e.tre, type="b",lwd=3,pch=16,ylim = c(0,yl), xlab="Order",
     ylab = "Eigenvalues", cex.axis=2, yaxt="n", cex.lab=2)
axis(2, las=2, cex.axis=2)
points(ave_eigen[1,], type="b", lwd=3,pch=16 ,col= "20")
points(ave_eigen[2,], type="b", lwd=3,pch=16 ,col= "red")
points(ave_eigen[3,], type="b", lwd=3,pch=16 ,col= "green")
mtext("(b)", side = 3, line = 0.5, cex = 2)

legend("topright", legend = c("True", "MLA", "Shrinkage", "MLE"),
       cex = 1.5,box.lwd = 1,box.lty = 1,
```

```
        text.font = 3,col = c("black","red", "green", 20),
        lty = 1, pch = 16)
dev.off()
```

Figure (4.3)c

```
library(MASS)
library(corpcor)

# t.exch function is required here to estimate "t"

p <- n.var <- 50
n <- 30
t <- 0.3
sigma <-  matrix(t, p, p) # Exchangeable covariance structure as a true covariance
     matrix
diag(sigma) = 1
e.tre <- eigen(sigma)$values

eigen.matrix <- matrix(NA,3,n.var)
x <- mvrnorm(n=n, mu= rep(0,p), Sigma=sigma)

res <- replicate(1000,{
  # Calculate the target matrix, i.e, exchangeable
  t.hat <- t.exch(x)
  TAR.EXCH <- matrix(t.hat, p, p)
  diag(TAR.EXCH) = 1
  S <- cor(x)

  # Calculate the gamma values for the proposed method
  gamma <- 1/(1+sum(abs(S - TAR.EXCH))/(p + (p*(p-1)*t.hat)))
  # Calculate the proposed estimator
  sigma.gamma = gamma*S + (1-gamma)*TAR.EXCH

  # calculate the eigenvalues of all three competing procedures
  MLA.eigen_values <- eigen(sigma.gamma)$values
  eigen_values.shrink <- eigen(cor.shrink(x,verbose = FALSE))$values
  MLE.eigen_values <- eigen(S)$values

  # Store the resulting eigenvalues in the null matrix
  eigen.matrix[1,] <- MLE.eigen_values
  eigen.matrix[2,] <- MLA.eigen_values
  eigen.matrix[3,] <- eigen_values.shrink
  eigen.matrix
})
# Average the resulting eigenvalues repeated 1000 times
ave_eigen <- apply(res, c(1,2), mean)
yl <- max(rbind(ave_eigen, e.tre))

pdf(file="FIG4_3c.pdf")
par(mar=c(7,6,2,1), mgp = c(4,1,0))
plot(e.tre, type="b",lwd=3,pch=16,ylim = c(0,yl), xlab="Order",
     ylab = "Eigenvalues", cex.axis=2, yaxt="n", cex.lab=2)
axis(2, las=2, cex.axis=2)
points(ave_eigen[1,], type="b", lwd=3,pch=16 ,col= "20")
points(ave_eigen[2,], type="b", lwd=3,pch=16 ,col= "red")
points(ave_eigen[3,], type="b", lwd=3,pch=16 ,col= "green")
mtext("(c)", side = 3, line = 0.5, cex = 2)

legend("topright", legend = c("True", "MLA", "Shrinkage", "MLE"),
        cex = 1.5,box.lwd = 1,box.lty = 1,
        text.font = 3,col = c("black","red", "green", 20),
```

```
        lty = 1, pch = 16)
dev.off()
```

Figure (4.3)d

```
library(MASS)
library(corpcor)
                # t.exch function is required here to estimate "t"

p <- n.var <- 50
n <- 30
t <- 0.3
sigma <-  diag(p) # Identity as a true covariance matrix
e.tre <- eigen(sigma)$values

eigen.matrix <- matrix(NA,3,n.var)

res <- replicate(1000,{
  x <- mvrnorm(n=n, mu= rep(0,p), Sigma=sigma)
  # Calculate target matrix, i.e, exchangeable covariance structure
  t.hat <- t.exch(x)
  TAR.EXCH <- matrix(t.hat, p, p)
  diag(TAR.EXCH) = 1
  S <- cor(x)

  # Calculate gamma values for the proposed method
  gamma <- 1/(1+sum(abs(S - TAR.EXCH))/(p + (p*(p-1)*t.hat)))
  # Calculate the proposed estimator
  sigma.gamma = gamma*S + (1-gamma)*TAR.EXCH

  # Calculate the eigenvalues of all the competing estimators of the true covariance
     matrix
  MLA.eigen_values <- eigen(sigma.gamma)$values
  eigen_values.shrink <- eigen(cor.shrink(x,verbose = FALSE))$values
  MLE.eigen_values <- eigen(S)$values

  eigen.matrix[1,] <- MLE.eigen_values
  eigen.matrix[2,] <- MLA.eigen_values
  eigen.matrix[3,] <- eigen_values.shrink
  eigen.matrix
})
# Average the resulting eigenvalues simulated 1000 times
ave_eigen <- apply(res, c(1,2), mean)
yl <- max(rbind(ave_eigen, e.tre))

pdf(file="FIG4_3d.pdf")
par(mar=c(7,6,2,1), mgp = c(4,1,0))
plot(e.tre, type="b",lwd=3,pch=16,ylim = c(0,yl), xlab="Order",
     ylab = "Eigenvalues", cex.axis=2, yaxt="n", cex.lab=2)
axis(2, las=2, cex.axis=2)
points(ave_eigen[1,], type="b", lwd=3,pch=16 ,col= "20")
points(ave_eigen[2,], type="b", lwd=3,pch=16 ,col= "red")
points(ave_eigen[3,], type="b", lwd=3,pch=16 ,col= "green")
mtext("(d)", side = 3, line = 0.5, cex = 2)

legend("topright", legend = c("True", "MLA", "Shrinkage", "MLE"),
        cex = 1.5,box.lwd = 1,box.lty = 1,
        text.font = 3,col = c("black","red", "green", 20),
        lty = 1, pch = 16)
dev.off()
```

Figure (4.3)e

```
library(MASS)
library(corpcor)
               # Function of random covariance structure is required here

n.var <- 50
n <- 30
prop.non.zero <- 0.30
const <- 0.01
sigma <- un.str.cov(n.var, prop.non.zero, const) # random covariance structure as a
    true covariance matrix

e.tre <- eigen(sigma)$values
eigen.matrix <- matrix(NA,3,n.var)

res <- replicate(1000,{
  x <- mvrnorm(n=n, mu= rep(0,n.var), Sigma=sigma)
  S <- cor(x)

  # Calculate gamma values for the proposed method
  gamma <- 1/(1+sum(abs(S - diag(n.var)))/(n.var))
  # Calculate teh proposed estimator of the true covariance matrix
  sigma.gamma = gamma*S + (1-gamma)*diag(n.var)

  # Calculate the eigenvalues of all three competing estimators
  MLA.eigen_values <- eigen(sigma.gamma)$values
  eigen_values.shrink <- eigen(cor.shrink(x,verbose = FALSE))$values
  MLE.eigen_values <- eigen(S)$values

  eigen.matrix[1,] <- MLE.eigen_values
  eigen.matrix[2,] <- MLA.eigen_values
  eigen.matrix[3,] <- eigen_values.shrink
  eigen.matrix
})
# Average the resulting eigenvalues simulated 1000 times
ave_eigen <- apply(res, c(1,2), mean)
yl <- max(rbind(ave_eigen, e.tre))

pdf(file="FIG4_3e.pdf")
par(mar=c(7,6,2,1), mgp = c(4,1,0))
plot(e.tre, type="b",lwd=3,pch=16,ylim = c(0,yl), xlab="Order",
     ylab = "Eigenvalues", cex.axis=2, yaxt="n", cex.lab=2)
axis(2, las=2, cex.axis=2)
points(ave_eigen[1,], type="b", lwd=3,pch=16 ,col= "20")
points(ave_eigen[2,], type="b", lwd=3,pch=16 ,col= "red")
points(ave_eigen[3,], type="b", lwd=3,pch=16 ,col= "green")
mtext("(e)", side = 3, line = 0.5, cex = 2)

legend("topright", legend = c("True", "MLA", "Shrinkage", "MLE"),
       cex = 1.5,box.lwd = 1,box.lty = 1,
       text.font = 3,col = c("black","red", "green", 20),
       lty = 1, pch = 16)
dev.off()
```

References

Beerenwinkel, N., Antal, T., Dingli, D., Traulsen, A., Kinzler, K. W., Velculescu, V. E., … Nowak, M. A. (2007). Genetic progression and the waiting time to cancer. *PLoS computational biology*, *3*(11), e225.

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

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

Eisen, M. B., Spellman, P. T., Brown, P. O., & Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, *95*(25), 14863–14868.

Fan, J., Liao, Y., & Liu, H. (2016). An overview of the estimation of large covariance and precision matrices. *The Econometrics Journal*, *19*(1).

Friedman, J., Hastie, T., Höfling, H., Tibshirani, R., et al. (2007). Pathwise coordinate optimization. *The Annals of Applied Statistics*, *1*(2), 302–332.

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

Golub, G., & Kahan, W. (1965). Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, *2*(2), 205–224.

Ledoit, O., & Wolf, M. (2003). Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *Journal of empirical finance*, *10*(5), 603–621.

Ledoit, O., & Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of multivariate analysis*, *88*(2), 365–411.

Penrose, R. (1955). A generalized inverse for matrices. In *Mathematical proceedings of the cambridge philosophical society* (Vol. 51, pp. 406–413).

Schaefer, J., Opgen-Rhein, R., Zuber, V., Ahdesmäki, M., Duarte Silva, A., & Strimmer, K. (2013). corpcor: Efficient estimation of covariance and (partial) correlation. *R package version*, *1*(6).

Schäfer, J., & Strimmer, K. (2004). An empirical bayes approach to inferring large-scale gene association networks. *Bioinformatics*, *21*(6), 754–764.

Schäfer, J., & Strimmer, K. (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical applications in genetics and molecular biology*, *4*(1).

Stein, C., et al. (1956). Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In *Proceedings of the third berkeley symposium on mathematical statistics and probability* (Vol. 1, pp. 197–206).

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., … Golub, T. R. (1999). Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proceedings of the National Academy of Sciences*, *96*(6), 2907–2912.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–288.

Warton, D. I. (2008). Penalized normal likelihood and ridge regularization of correlation and covariance matrices. *Journal of the American Statistical Association*, *103*(481), 340–349.

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