MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

Sparse Summaries of Complex Covariance Structures

A thesis submitted in partial fulfilment of the requirements for the

degree of

Doctor of Philosophy

in

Statistics

Amir Bashir

School of Natural & Computational Sciences

Massey University

Auckland, New Zealand

May 2020

In the loving memory of my mother

Abstract

A matrix that has most of its elements equal to zero is called a sparse matrix. The zero elements in a sparse matrix reduce the number of parameters for its potential interpretability. Bayesians desiring a sparse model frequently formulate priors that enhance sparsity. However, in most settings, this leads to sparse posterior samples, not to a sparse posterior mean. A decoupled shrinkage and selection posterior - variable selection approach was proposed by (Hahn & Carvalho, 2015) to address this problem in a regression setting to set some of the elements of the regression coefficients matrix to exact zeros. Hahn & Carvallho (2015) suggested to work on a decoupled shrinkage and selection approach in a Gaussian graphical models setting to set some of the elements of a precision matrix (graph) to exact zeros. In this thesis, I have filled this gap and proposed decoupled shrinkage and selection approaches to sparsify the precision matrix and the factor loading matrix that is an extension of Hahn & Carvallho's (2015) decoupled shrinkage and selection approach. The decoupled shrinkage and selection approach proposed by me uses samples from the posterior over the parameter, sets a penalization criteria to produce progressively sparser estimates of the desired parameter, and then sets a rule to pick the final desired parameter from the generated parameters, based on the posterior distribution of fit. My proposed decoupled approach generally produced sparser graphs than a range of existing sparsification strategies such as thresholding the partial correlations, credible interval, adaptive graphical $Lasso$, and ratio selection, while maintaining a good fit based on the log-likelihood. In simulation studies, my decoupled shrinkage and selection approach had better sensitivity and specificity than the other strategies as

the dimension p and sample size n grew. For low-dimensional data, my decoupled shrinkage and selection approach was comparable with the other strategies.

Further, I have extended my proposed decoupled shrinkage and selection approach for one population to two populations by modifying the ADMM (alternating directions method of multipliers) algorithm in the JGL (joint graphical $Lasso$) R – package (Danaher et al, 2013) to find sparse sets of differences between two inverse covariance matrices. The simulation studies showed that my decoupled shrinkage and selection approach for two populations for the sparse case had better sensitivity and specificity than the sensitivity and specificity using JGL. However, sparse sets of differences were challenging for the dense case and moderate sample sizes. My decoupled shrinkage and selection approach for two populations was also applied to find sparse sets of differences between the precision matrices for cases and controls in a metabolomics dataset.

Finally, decoupled shrinkage and selection is used to post-process the posterior mean covariance matrix to produce a factor model with a sparse factor loading matrix whose expected fit lies within the upper 95% of the posterior over fits. In the Gaussian setting, simulation studies showed that my proposed DSS sparse factor model approach performed better than $\boldsymbol{fanc}$ (factor analysis using non-convex penalties) (Hirose and Yamamoto, 2015) in terms of sensitivity, specificity, and picking the correct number of factors. Decoupled shrinkage and selection is also easily applied to models where a latent multivariate normal underlies non-Gaussian marginals, e.g., multivariate probit models. I illustrate my findings with moderate dimensional data examples from modelling of food frequency questionnaires and fish abundance.

Acknowledgements

First of all, I thank the HEC (Higher Education Commission) of Pakistan for funding my PhD studies without which my PhD studies would not have been possible. I am grateful to my employer, the IBA (Institute of Business Administration) Pakistan as well to grant me paid study leaves to pursue my PhD studies.

I am grateful to my supervisor Dr. Adam Nicholas Howard Smith for his suggestions and constructive criticisms. I thank distinguished professor Marti Jane Anderson for being my co-supervisor. The completion of this PhD work would not have been possible without the help of my co-supervisor Dr. Mary Beatrix Jones, to whom I will always be indebted. Dr. Beatrix, you have always guided me really well whenever I was stuck at some point in my PhD research. I will always remember you throughout my life for your academic and moral support.

I would also like to add that this stage marks an important milestone in what has been a very long personal journey influenced by numerous people and events. A sincere thanks to all of them, especially my friend Ashar J. Malik. A special thanks to my parents, siblings and relatives for tolerating my insanity.

Finally, to Javeria, Wajeeha and Sheheryar, the three strongest and most influential people in my life, without whom I would truly be lost.

Contents

List of Figures

List of Tables

Chapter - 01

Introduction

1.1 Introduction

A covariance matrix quantifies the linear relationships among the variables in multivariate Gaussian data. The inverse covariance matrix quantifies the partial covariances between pairs $\boldsymbol{i}$ and $\boldsymbol{j}$, given the other variables in a multivariate normal context. The zero elements in the inverse covariance matrix denote the conditional independence between each pair of variables given the relationships among other variables. The non-zero elements in the inverse covariance matrix are called the edges. Sparsification of some of the elements of the inverse covariance matrix to exact zeros is of utmost importance since we have fewer elements in the inverse covariance matrix to interpret.

The estimation of the sample covariance matrix in a high-dimensional setting is challenging, particularly when the dimension $(\boldsymbol{p})$ is greater than the sample size $(\boldsymbol{n})$ i.e. $\boldsymbol{p} > \boldsymbol{n}$. In this case, the estimate will not be of full rank unless additional structure is imposed. To address this issue, a range of penalized likelihood and Bayesian methods discussed in chapter 2 have been proposed for fitting sparse models of covariance matrices that have fewer parameters than a full covariance matrix for its potential interpretability.

Model selection was performed on the inverse covariance matrix (precision matrix) using covariance selection modelling proposed by Dempster (1973). Frequentist methods such as the graphical $Lasso$ (Friedman et al, 2008), the adaptive graphical $Lasso$ (Choi et al, 2010) discussed in chapter 2 use different penalization techniques to shrink some of the off-diagonal elements of the precision matrix to exact zeros. Similarly, Bayesian approaches such as the Bayesian graphical $Lasso$ and the Bayesian adaptive graphical $Lasso$ (Wang, 2012), also, discussed in chapter 2 use different priors to generate the posterior samples of the covariance matrix, and then shrink some of the off-diagonal elements of the precision matrix to exact zeros. Peterson et al. (2013) obtained the posterior samples of inverse covariance matrices using Bayesian adaptive graphical

$Lasso$ (Wang, 2012) and set two strategies to sparsify the inverse covariance matrix. The first strategy suggested by (Wang, 2012) was setting to exact zeros the off-diagonal elements of the inverse covariance matrix that had absolute partial correlations less than 0.1, and setting to non-zero (edges) the off-diagonal elements of the inverse covariance matrix that had absolute partial correlations greater than or equal to 0.1. The second strategy, suggested by (Wang, 2012) was to set to zero the off-diagonal elements of the inverse covariance matrix for which the 90% credible interval included zeros. Most of the penalized likelihood and Bayesian approaches have arbitrary criterion for selection of final inverse covariance matrix. I have proposed a DSS (decoupled shrinkage and selection) approach summarized in subsection 1.1.1 to sparsify some of the elements of the inverse covariance matrix to exact zeros. My proposed DSS approach picks the final inverse covariance matrix from a set of generated inverse covariance matrices based on a certain rule explained in subsection 1.1.1.

1.1.1 My Main Contribution

Hahn & Carvalho (2015) proposed a posterior variable-selection summary approach in regression setting called DSS (decoupled shrinkage and selection) to set some of the elements of the regression coefficients matrix to zero, detailed in chapter 2. Hahn & Carvalho (2015) suggested to extend their DSS approach in regression setting to DSS approach in a Gaussian graphical models setting to set some of the elements of the precision matrix to zero. I have filled this gap and proposed a decoupled shrinkage and selection approach for estimation of edges in an inverse covariance matrix detailed in section 1.3. The Gaussian graphical models and factor analysis models are discussed in section 1.2.

My proposed DSS approach for sparsification of inverse covariance matrix and the factor analysis models is based on the following steps:

1. Generate the samples from the posterior over the parameter (precision matrix or factor loading matrix).

2. Set a criterion to produce progressively sparse estimates of the desired parameter.

3. Set a rule to pick the final desired parameter matrix from the generated desired parameter matrices, based on posterior distribution of fit.

Danaher et al. (2013) proposed a new joint graphical $Lasso$ (JGL) convex-optimization methodology discussed in chapter 3 for joint estimation of precision matrices in a high dimensional data setting for multiple classes, which had a faster computation time as compared to the proposal of Guo et al. (2011). I extended my proposed DSS-based method for one graph to DSS-based method for two graphs to find the sparse sets of differences between the two precision matrices by using a different penalty function in the JGL algorithm.

A factor loading matrix represents correlations between the observed quantitative variables and unobserved latent variables called factors. A zero correlation between an observed variable and a factor means that the variable and the factor are independent from each other. I also propose a DSS sparse factor model to shrink some of the elements of the factor loading matrix to exact zeros so that the dimension is reduced and we are left with fewer variables and factors for easy interpretation. My proposed DSS approach is also an arbitrary approach that provides a framework to compare different selection methods with comparable thresholds such as thresholding the partial correlations, credible interval, the adaptive graphical $Lasso$, and ratio selection methods which all have arbitrary thresholds as well to select the precision matrix.

Some matrix norms and performance measures are explained in section 1.4. Section 1.5 gives the outline of the thesis.

1.2 Gaussian Graphical and Factor Analysis Models

The details about the Gaussian graphical models and factor analysis models are given in subsections 1.2.1 and 1.2.2 respectively.

1.2.1 Gaussian Graphical Models

A graph is represented by vertices (variables) and the non-zero relationships among variables are denoted by the edges $\boldsymbol{E}$. There are several types of graphs, including null

or empty graphs (having 0 edges), directed graphs and undirected graphs. In a directed graph, the relationships among variables have specific directions, represented by arrows (Figure 1.2). With undirected graphs, relationships have no specific direction (Figure 1.1).

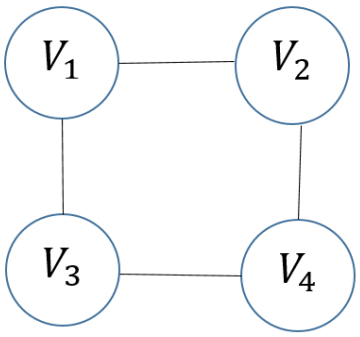


Figure 1.1: An Undirected Graph

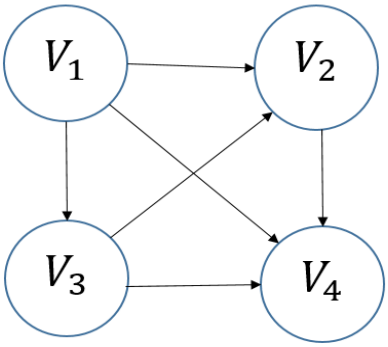


Figure 1.2: A Directed Graph

In Gaussian graphical models (GGMs), we assume that the data follow a multivariate normal distribution. We can represent the precision matrix, denoted by $\boldsymbol{\Omega}$, as a graph with vertices $\boldsymbol{V}$ corresponding to variables and $\boldsymbol{E}$ corresponding to edges (the non-zero elements in $\boldsymbol{\Omega}$). The zero elements in $\boldsymbol{\Omega}$ reduce the number of parameters. Sparsification of a graph is desirable for its potential interpretability. Graphical models have vast applications such as inferring protein interaction networks, gene regulatory networks and co-expression networks in genomics and proteomics (Dobra et al, 2004; Friedman, 2004; Mukherjee and Speed, 2008; Stingo et al, 2010). These models are also used to infer international financial flows (Giudici and Spelta, 2016). The graphical models also have wide applications in fault diagnosis (Adel Aloraini and Moamar Sayed-Mouchaweh, 2014).

1.2.2 Factor Analysis Models

The covariance/correlation structure among the variables can be analysed by a statistical technique called factor analysis that is an alternative to the Gaussian graphical model representing a different sort of structure. The relationship between the factors and the variables is represented by a factor loading matrix, denoted here by $\boldsymbol{\Lambda}$. The factor loading matrix obtained using standard factor analysis does not have any zero loadings, which

can be considered a drawback. The interpretation of the correlations among the observed quantitative variables and the factors could be simplified by shrinking some of the correlations in the factor loading matrix to exact zeros. Where a factor and a variable have no relationship, this is represented by a factor loading equal to zero. If $\boldsymbol{k}$ is the number of factors, then the Gaussian factor model is expressed as follows:

$$\underset{(p\times 1)}{y_i} = \underset{(p\times k)}{\boldsymbol{\Lambda}}\ \underset{(k\times 1)}{\eta_i} + \underset{(p\times 1)}{\epsilon_i} \qquad (1.1)$$

The observed variables vector is denoted by y_i , the factor scores vector is denoted by η_i , and the idiosyncratic noise is represented by $\epsilon_i \sim N(\mathbf{0}, \boldsymbol{\Psi})$ with $\boldsymbol{\Psi} = diag\left(\sigma_1^2, \ldots, \sigma_p^2\right)$. In Gaussian graphical models, we set to zero some of the elements of a precision matrix. Whereas, a Gaussian factor model has a different sort of structure in which observed quantitative variables make a meaningful pattern with unobserved latent variables called factors, and we set to zero some of the elements of the factor loading matrix. Several sparse factor analysis models detailed in chapter 4 have been proposed to sparsify the factor loading matrix given in Equation (1.1).

1.3 Proposed Decoupled Shrinkage and Selection Method

Let $\boldsymbol{\Sigma_k}$ denote $\boldsymbol{k}$ posterior samples of covariance matrices of future observations, and $\overline{\boldsymbol{\Sigma}}^{\boldsymbol{-1}}$ be the posterior mean of $\boldsymbol{k}$ inverse covariance matrices, then my proposed decoupled shrinkage and selection equation in Gaussian graphical and factor model setting is as follows:

$$fit(\overline{\boldsymbol{\Sigma}}^{\boldsymbol{-1}} | \boldsymbol{\Sigma_k}) = log\left(det(\overline{\boldsymbol{\Sigma}}^{\boldsymbol{-1}})\right) - tr(\boldsymbol{\Sigma_k}\overline{\boldsymbol{\Sigma}}^{\boldsymbol{-1}}) \qquad (1.2)$$

The above Equation (1.2) denotes a sample from distribution of $\boldsymbol{fit}(\overline{\boldsymbol{\Sigma}}^{\boldsymbol{-1}})$.

After getting the posterior samples $\boldsymbol{\Sigma_k}$ and $\boldsymbol{\Sigma_k^{-1}}$, we follow the following steps in one of our proposed DSS approaches called DSS credible interval approach:

1. Calculate the mean of $\boldsymbol{\Sigma_k}$, $\overline{\boldsymbol{\Sigma}}$, and $\overline{\boldsymbol{\Sigma}}^{\boldsymbol{-1}}$. Where, the Bayes estimate of $\overline{\boldsymbol{\Sigma}}^{\boldsymbol{-1}}$ is not sparse.

2. Find a central 90% credible interval for the range of fit of $\bar{\boldsymbol{\Sigma}}^{-1}$.

3. Find the $\boldsymbol{X}\%$ credible intervals for the $\boldsymbol{\omega_{ij}}$ based on the sampled inverse covariance matrices.

4. Observe the credible intervals where the elements are zero representing conditional independence between the variables given the other variables.

5. Find the maximum likelihood estimate with certain values fixed at zero by using the graphical $Lasso$ algorithm (Friedman et al, 2008).

6. Finally, choose the final inverse covariance matrix with a suitable fit based on the log-likelihood.

1.4 Matrix Norms and Performance Measures

The size of the covariance matrix and the inverse covariance matrix, denoted here by $\boldsymbol{\Omega}$, can be estimated by applying matrix norms such as $\boldsymbol{L_1}$-norm, $\boldsymbol{L_2}$-norm and $\boldsymbol{L_\infty}$-norm in a Gaussian graphical models setting, which are used in penalization / regularization strategies such as $Lasso$, the adaptive $Lasso$, the graphical $Lasso$, and the adaptive graphical $Lasso$. The performance of estimated inverse covariance matrix $\boldsymbol{\Omega}$ for graph recovery is measured as sensitivity or recall, specificity, precision or positive predictive value, and $\boldsymbol{MCC}$ (Mathews' correlation coefficient). The details of these matrix norms and performance measures are provided in subsections 1.4.1 and 1.4.2.

1.4.1 Matrix Norms

The details of the matrix norms such as $\boldsymbol{L_1}$ - norm, $\boldsymbol{L_2}$ - norm and $\boldsymbol{L_\infty}$ - norm are given as follows:

$(\boldsymbol{i})$ $\boldsymbol{L_1}$ **– Norm**

It is the sum of the absolute values of a matrix.

$$\|X\|_1 = \sum_i |X_i| \qquad (1.3)$$

where, X_i is a vector.

(ii) L_2 – Norm

It is the square root of the sum of the squared values of a matrix. It is also called the Euclidean norm.

$$\|X\|_2 = \sqrt{\sum_i X_i^2} \tag{1.4}$$

(iii) L_∞ – Norm

It is defined as the maximum absolute value of a matrix.

$$\|X\|_\infty = max(|X_i|) \tag{1.5}$$

1.4.2 Performance Measures

(i) Sensitivity

Sensitivity is the ratio between TP (true positives) and the sum of TP and FN (false negatives). It is also called the true positive rate (TPR), or recall. The proportion of true positives that are correctly identified is measured by 'sensitivity'. It is calculated using the following formula:

$$Sensitivity\ (TPR) = \frac{TP}{TP+FN} \tag{1.6}$$

(ii) Specificity

Specificity is the ratio between TN (true negatives) and the sum of TN and FP (false positives). It is also called the true negative rate (TNR). The proportion of true negatives that are correctly identified is measured by 'specificity'. It is calculated using the following formula:

$$Specificity\ (TNR) = \frac{TN}{TN+FP} \tag{1.7}$$

(iii) Precision

Precision is the ratio between $\boldsymbol{TP}$ (true positives) and all the positives i.e. sum of $\boldsymbol{TP}$ and $\boldsymbol{FP}$ (false positives). It is also called positive predicted value. Precision tells us about correct positive predictions. It's calculated using the following formula:

$$Precision = \frac{TP}{TP+FP} \tag{1.8}$$

(iv) Mathews' Correlation Coefficient

Mathews' correlation coefficient ($\boldsymbol{MCC}$) is the correlation coefficient between true and false negatives and positives which can be used for classes of different sizes. It lies between -1 and $+1$. The -1 value indicates complete disagreement between observed and predicted binary classifications, and $+1$ value indicates a perfect prediction. The formula for $\boldsymbol{MCC}$ is as follows:

$$MCC = \frac{TP*TN-FP*FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{1.9}$$

1.5 Outline of Thesis

1. In chapter 2, I evaluate a new Bayesian method called decoupled shrinkage and selection (DSS) approach in Gaussian graphical models setting for adjusting to zero some of the off-diagonal elements of a precision matrix. The Bayes estimate of the posterior mean inverse covariance matrix is not sparse even for the posterior over sparse models. My proposed DSS method is based on generating posterior samples of covariance matrices, estimating means of the posterior samples of covariance matrices and the inverse covariance matrices, and then shrinking some of the off-diagonal elements of the inverse covariance matrices to exact zeros using different edge selection criterion. I compare the performance of my proposed DSS method with previous strategies on the basis of sensitivity and specificity. I apply my proposed DSS method on a real dataset of frequencies of $p = 174$ metabolites in the fecal sample of 49, 8 year old children born at term, and compare my DSS method with other edge selection strategies such as the adaptive

graphical $Lasso$, thresholding the partial correlations, credible interval, and ratio selection methods.

2. In chapter 3, I extend my proposed DSS method for one population to two populations to find differences in the covariance matrices of two datasets with the same variables. I use simulation studies to compare the performance of my proposed DSS method for two graphs with $\boldsymbol{JGL}$ (Joint graphical $Lasso$) (Danaher et al, 2013) based on sensitivity, specificity, and $\boldsymbol{MCC}$. I apply the DSS method for two graphs to find important $\boldsymbol{\Omega}$ differences between cases and controls metabolites data.

3. In chapter 4, I propose a DSS sparse factor model to shrink some elements of the factor loading matrices to exact zeros. After generating posterior samples of the loading matrices using a certain algorithm, I use the following expression to convert the loading matrices into the covariance matrices, and then apply my DSS sparse factor model to shrink some of the elements of the loading matrices to exact zeros.

$$\Sigma = \Lambda\Lambda^T + \Psi \qquad (1.10)$$

where, Ψ denotes the unique variances obtained using bfa (Bayesian factor analysis) R – package (Murray et. al, 2013).

I use simulation studies to assess the performance of my proposed DSS sparse factor model based on true positive rate, true negative rate, and true discovery rate, with $\boldsymbol{fanc}$ (factor analysis using non-convex penalties) – based methods (Hirose and Yamamoto, 2015). I also compare the number of factors identification with the true factor model using my proposed DSS sparse factor model and $\boldsymbol{fanc}$ – based methods (Hirose et. al, 2015). I apply my proposed DSS sparse factor model on a continuous food questionnaire dataset (Mumme et. al, 2019), and a discrete fishes abundance dataset (Smith, Duffy and Leathwick, 2013), to shrink some of the elements of the factor loading matrices to exact zeros for both continuous and discrete cases.

4. Chapter 5 presents the conclusions and recommendations regarding my proposed decoupled shrinkage and selection methods in chapters 2, 3, and 4 respectively.

Chapter - 02

Sparse Estimates of Precision Matrices Using Gaussian Graphical Models

2.1 Introduction

Sparsifying a precision matrix (graph) by setting some of the off-diagonal elements to exact zeros is desirable because zero elements in a graph reduce the number of parameters for its potential interpretability. Penalized likelihood approaches, such as the graphical $Lasso$ (Friedman et al, 2008), the adaptive graphical $Lasso$, and CLIME (Constrained $\boldsymbol{L_1}$ minimization estimation) (Cai et al, 2011), have been proposed to set to zero some of the off-diagonal elements of a precision matrix. In addition, a variety of Bayesian models have been proposed for model selection to produce sparse graphs in Gaussian graphical models. Generation of a posterior distribution over graphs is computationally intensive and involves exploring a large discrete space, despite recent advances (Muhammadi and Wit, 2015). The Bayes estimate of the posterior mean inverse covariance matrix is not sparse even for the posterior over sparse models (see Table 2.2). Methods such as the conjugate inverse Wishart prior, the factor analysis models (West, 2003), regularized inverse Wishart (Kundu et al., 2018), and the Bayesian adaptive graphical $Lasso$ (Wang, 2012; Peterson et al, 2013) are appropriate methods for shrinking the covariance model where graphs are not required to be sparse. Yet, in many cases, sparse graphs are required and the development of accurate and efficient methods of producing sparse graphs is an active area of research.

Several methods of producing sparse graphs have been proposed. Peterson et al. (2013) consider the edges in the precision matrix to be connected if the absolute partial correlation between the edges is greater than 0.1. This method of selection is considered arbitrary. Wang (2012) suggests that the edges in the precision matrix are connected if and only if the posterior mean of inverse covariance matrices using the graphical $Lasso$ priors and the expected value of the posterior mean of inverse covariance matrices using $\boldsymbol{W(3, Ip)}$ has a ratio greater than 0.5. Here, $\boldsymbol{W(3, Ip)}$ is the standard conjugate Wishart prior. Like that of Peterson et al. (2013), this method of selection is considered arbitrary.

Hahn & Carvalho (2015) proposed the DSS (Decoupled Shrinkage and Selection) method to produce sparse estimates in a regression setting. Hahn & Carvalho's (2015) DSS-based method produces the posterior predictive distribution of the future data, and Hahn & Carvalho (2015) gave preliminary suggestions to extend their DSS method for estimation of precision matrices.

Here, I am extending the DSS-based method for estimation of edges in a precision matrix in Gaussian graphical models. My proposed DSS-based method selects the final precision matrix based on the posterior predictive distribution of the future data and the log-likelihood of the fit of the model to these future data.

The objective of this chapter is to evaluate a new method of model selection for Gaussian graphical models. I propose a DSS method for edge detection in a precision matrix and compare its performance with previous strategies. The proposed method performs well based on sensitivity and specificity for low to moderate dimensional data.

The remainder of this chapter is organised as follows. Section 2.2 presents a review of current approaches, including those based on penalized likelihood (the graphical $Lasso$; the adaptive graphical $Lasso$; Constrained $\boldsymbol{L_1}$ minimization estimation, or CLIME; elastic net; Tuning insensitive graph estimation and regression, or TIGER) and methods for tuning based on log-likelihood (cross-validation; stability selection; and BINCO, a frequentist approach for model selection based on controlling the false discovery rate). Section 2.3 describes Bayesian approaches for the estimation of precision matrix including my proposed DSS method. Section 2.4 describes methodological comparisons based on a simulation study, Section 2.5 describes methodological comparisons based on real data, and section 2.6 presents the conclusion.

2.2 Review of Current Approaches

2.2.1 Penalized Likelihood Approaches

Penalized likelihood estimation approaches include the graphical $Lasso$, the adaptive graphical $Lasso$, CLIME (Constrained $\boldsymbol{L_1}$ minimization estimation), and elastic net. All these methods select edges in a precision matrix based on some penalty with tuning

parameters. The graphical $Lasso$, the adaptive graphical $Lasso$ and CLIME use $\boldsymbol{L_1}$ penalty, whereas, elastic net uses a combination of $\boldsymbol{L_1}$ and $\boldsymbol{L_2}$ penalties.

2.2.1.1 Graphical $Lasso$

In Gaussian graphical models, $\boldsymbol{\Sigma}$ denotes the population covariance matrix, and $\boldsymbol{\Omega}$ denotes the population precision matrix. The data dimension is denoted by $\boldsymbol{p}$, and the number of observations is denoted by $\boldsymbol{n}$. To overcome the problem of estimating $\boldsymbol{\Sigma}$ and $\boldsymbol{\Omega}$ for high-dimensional data ($\boldsymbol{p} > \boldsymbol{n}$), the graphical $Lasso$ was developed by (Friedman et al, 2008). Graphical $Lasso$ penalizes the sum of absolute values of the off-diagonal elements of the inverse covariance matrix. The purpose of the graphical $Lasso$ was the estimation of sparse inverse covariance matrix for undirected graphs. The graphical $Lasso$ is a penalized likelihood estimation method and solves the following problem:

$$maximize_{\boldsymbol{\Omega}}\{log\ det\ \boldsymbol{\Omega} - trace(\boldsymbol{\Sigma}\ \boldsymbol{\Omega}) - \lambda\|\boldsymbol{\Omega}\|_1\}\ , \qquad (2.1)$$

where, λ is a tuning parameter that is non-negative. The sum of absolute off-diagonal entries of the precision matrix are penalized using $\boldsymbol{L_1}$ penalty. The graph becomes sparser as the tuning parameter λ increases, and all the estimates of non-zero entries are biased towards 0.

2.2.1.2 Adaptive Graphical $Lasso$

One problem with the graphical $Lasso$ is that, to achieve more sparsity, one must tolerate downwardly biased estimates of the precision matrix towards zero. To overcome this problem, the adaptive graphical $Lasso$ adjusts the penalty of each ω_{ij} with the following factor:

$$\xi_{ij} = \frac{1}{|\hat{\Omega}|^{\gamma}} \qquad (2.2)$$

Here, $\hat{\Omega}$= $\left(\hat{\omega}_{i,j}\right)_{1\leq i,j\leq p}$ can be any consistent estimate of Ω. Also $\gamma > 0$, (usually taken to a fixed value $\gamma = \frac{1}{2}$) (Zou, 2006).

Thus, the adaptive graphical $Lasso$ algorithm solves the following problem:

$$maximize_{\boldsymbol{\Omega}}\left\{log\ det\ \boldsymbol{\Omega} - trace(\boldsymbol{\Sigma}\,\boldsymbol{\Omega}) - \lambda\left\|\boldsymbol{\xi_{ij}}\,\boldsymbol{\omega_{ij}}\right\|_1\right\} \qquad (2.3)$$

The weighted penalty used in the adaptive graphical $Lasso$ reduces the bias because the weighted penalty imposes less shrinkage on the coefficients with larger magnitude.

2.2.1.3 Constrained L_1 Minimization Estimation (CLIME)

Cai et al. (2011) developed a new approach, CLIME (Constrained $\boldsymbol{L_1}$ minimization estimation), for estimating high-dimensional inverse covariance matrices. Let $\widehat{\Omega}_1$ be the following optimization problem solution set:

$$\min\|\boldsymbol{\Omega}\|_1\ \ subject\ to\colon |\boldsymbol{\Sigma}_n\Omega - \boldsymbol{I}|_\infty\ \leq\ \lambda_n\ ,\ \ \Omega \in R^{p\times p} \qquad (2.4)$$

where, λ_n is the tuning parameter and $\boldsymbol{\Sigma_n}$ is the sample covariance matrix. Here the symmetry condition i.e. $\Omega^T = \Omega$ is not applied on Ω. The symmetrizing of $\widehat{\Omega}_1$ is done to obtain the CLIME estimator of Ω_0. In order to symmetrize $\widehat{\Omega}_1$, we write $\widehat{\Omega}_1$ as below:

$$\widehat{\boldsymbol{\Omega}}_1 = (\widehat{\boldsymbol{\omega}}_{ij}^1) = (\widehat{\boldsymbol{\omega}}_1^1\ ,\ \ldots\ ,\widehat{\boldsymbol{\omega}}_P^1\ ) \qquad (2.5)$$

$\widehat{\Omega}$ that is the CLIME estimator of Ω_0 is defined as below:

$$\widehat{\boldsymbol{\Omega}} = \left(\widehat{\boldsymbol{\omega}}_{ij}\right), \qquad (2.6)$$

where, $$\widehat{\boldsymbol{\omega}}_{ij} = \widehat{\boldsymbol{\omega}}_{ji} = \widehat{\boldsymbol{\omega}}_{ij}^1\boldsymbol{I}\left\{\left|\widehat{\boldsymbol{\omega}}_{ij}^1\right| \leq \left|\widehat{\boldsymbol{\omega}}_{ji}^1\right|\right\} + \widehat{\boldsymbol{\omega}}_{ji}^1\boldsymbol{I}\left\{\left|\widehat{\boldsymbol{\omega}}_{ij}^1\right| > \left|\widehat{\boldsymbol{\omega}}_{ji}^1\right|\right\}$$

This causes the estimated precision matrix $\widehat{\Omega}$ to be symmetric. The graphical $Lasso$ log-likelihood is a smooth curve with respect to λ but the analogous log-likelihood of CLIME forms a polygon. In contrast to the graphical $Lasso$ where the $\boldsymbol{L_1}$ norm is used, element-wise $\boldsymbol{L}_\infty$ norm is used for graphical model selection in CLIME.

Cai et al. (2011) compared the numerical performance of CLIME estimator with the graphical $Lasso$ and SCAD (Smoothly Clipped Absolute Deviation) (Fan and Li, 2001) based on sensitivity, specificity and MCC (Mathews' correlation coefficient) with simulated data. CLIME performed better than the graphical $Lasso$ and SCAD in terms of sensitivity (true positive rate) and MCC, and was comparable with the graphical $Lasso$ and SCAD in

terms of specificity (true negative rate). The tuning parameter λ was selected in the graphical $Lasso$, SCAD and CLIME for comparison using cross-validation. In the graphical $Lasso$, the adaptive graphical $Lasso$ and CLIME, model selection is equivalent to tuning parameter selection.

2.2.1.4 Elastic Net

Ridge regression helps in reducing the multicollinearity (dependence of explanatory variables on each other) by penalizing the sum of squared coefficients using an $\boldsymbol{L_2}$ penalty. The $Lasso$ algorithm is based on an $\boldsymbol{L_1}$ penalty, and the ridge methods are based on an $\boldsymbol{L_2}$ penalty. The $\boldsymbol{L_2}$ penalty penalizes the sum of squared values of the off-diagonal elements of a precision matrix in Gaussian graphical models. Elastic net is a methodology that linearly combines both $\boldsymbol{L_1}$ and $\boldsymbol{L_2}$ penalties of $Lasso$ and ridge methods respectively, as follows:

$$maximize_{\boldsymbol{\Omega}}\{log\ det\ \boldsymbol{\Omega} - trace(\boldsymbol{\Sigma}\ \boldsymbol{\Omega}) - \lambda_2\|\Omega\|_2^2 - \lambda_1\|\Omega\|_1\} \qquad (2.7)$$

where, $\|\Omega\|_2^2$ is the quadratic part of the penalty; and $\|\Omega\|_1$ is the $Lasso$ penalty. Strict convexity is achieved by the quadratic penalty in the loss function 2.7. Hence, it has a unique minimum. Elastic net is equivalent to $Lasso$ when $\lambda_1 = \lambda, and\ \lambda_2 = 0$, and to ridge when $\lambda_1 = 0, and\ \lambda_2 = \lambda$. The elastic net is actually a $Lasso$ on an augmented data set. The ridge method does not enforce the off-diagonal elements of the precision matrix to be exactly zero. However, elastic net does set some elements to zero, thereby producing sparse models (Zou & Hastie; 2005).

2.2.2 Tuning-free Method

Most of the penalized-likelihood approaches require the selection of at least one tuning parameter. Liu and Wang (2012) proposed a new tuning-free method, TIGER (tuning-Insensitive graph estimation and regression), for estimating high-dimensional Gaussian Graphical models. TIGER uses the SQRT-$Lasso$ regression (Belloni et al, 2012) to solve the sparse regression problem. The SQRT-$Lasso$ equation to find the estimates β for the linear regression problem: $\boldsymbol{y} = \boldsymbol{X}\beta + \epsilon$; is as follows:

$$\hat{\beta} = \underset{\beta \epsilon \mathbb{R}^d}{\arg min} \left\{ \frac{1}{\sqrt{n}} \|\boldsymbol{y} - \boldsymbol{X}\beta\|_2 + \lambda \|\beta\|_1 \right\} \quad (2.8)$$

where, y is the response variable, X is the design matrix, β is the unknown coefficients vector, ϵ is the noise vector, and λ is the tuning parameter. Belloni et al (2012) showed that the tuning parameter λ choice for SQRT-$Lasso$ was independent of any unknown parameter. However, heavy reliance on the known standard deviation of the error term is required for methods such as the $Lasso$ and Dantzig selector (Yuan, 2009). The explanation of TIGER in terms of estimation of the precision matrix is given as follows:

Let $\widehat{\boldsymbol{\Gamma}} := diag(\widehat{\boldsymbol{\Sigma}})$ represents a diagonal matrix with dimension "d", and let us assume that it has the same diagonal elements as those in $\hat{\Sigma}$. Now we define:

$$\boldsymbol{Z} \coloneqq (Z_1, \ldots, Z_d)^T = \boldsymbol{X}\widehat{\boldsymbol{\Gamma}}^{-1/2} \quad (2.9)$$

Also we define:

$$\widehat{\beta_j} \coloneqq \widehat{\boldsymbol{\Gamma}}_{\backslash j,\backslash j}^{\frac{1}{2}} \widehat{\boldsymbol{\Gamma}}_{jj}^{-\frac{1}{2}} \alpha_j \quad and \quad \tau_j^2 = \sigma_j^2 \, \widehat{\boldsymbol{\Gamma}}_{jj}^{-1} \quad (2.10)$$

where, $\alpha_j \coloneqq \left(\boldsymbol{\Sigma}_{\backslash j,\backslash j}\right)^{-1} \boldsymbol{\Sigma}_{\backslash j,j} \in \mathbb{R}^{d-1}$ and $\sigma_j^2 \coloneqq \boldsymbol{\Sigma}_{jj} - \boldsymbol{\Sigma}_{\backslash j,j} \left(\boldsymbol{\Sigma}_{\backslash j,\backslash j}\right)^{-1} \boldsymbol{\Sigma}_{\backslash j,j}$

Hence, we have:

$$Z_j = \beta_j^T \boldsymbol{Z}_{\backslash j} + \widehat{\boldsymbol{\Gamma}}_{jj}^{-\frac{1}{2}} \epsilon_j \quad (2.11)$$

Liu and Wang (2012) proposed TIGER for the estimation of precision matrix Θ as follows:

$$\hat{\beta}_j \coloneqq \underset{\beta_j \epsilon \mathbb{R}^{d-1}}{\arg min} \left\{ \sqrt{1 - 2\beta_j^T \, \widehat{\boldsymbol{R}}_{\backslash j,j} + \beta_j^T \, \widehat{\boldsymbol{R}}_{\backslash j,\backslash j} \beta_j} + \lambda \left\| \beta_j \right\|_1 \right\}, \quad (2.12)$$

$$\hat{\tau}_j \coloneqq \sqrt{1 - 2\hat{\beta}_j^T \, \widehat{\boldsymbol{R}}_{\backslash j,j} + \hat{\beta}_j^T \, \widehat{\boldsymbol{R}}_{\backslash j,\backslash j} \hat{\beta}_j} \,,$$

$$\widehat{\boldsymbol{\Theta}}_{jj} = \hat{\tau}_j^{-2} \, \widehat{\boldsymbol{\Gamma}}_{jj}^{-1} \; and \; \widehat{\boldsymbol{\Theta}}_{\backslash j,j} = -\hat{\tau}_j^{-2} \, \widehat{\boldsymbol{\Gamma}}_{jj}^{-\frac{1}{2}} \, \widehat{\boldsymbol{\Gamma}}_{\backslash j,\backslash j}^{\frac{-1}{2}} \hat{\beta}_j$$

The tuning parameter λ for TIGER is as follows:

$$\lambda := \zeta\pi\sqrt{\frac{\log d}{2n}} \tag{2.13}$$

where, ζ is chosen between the interval $\left[\frac{\sqrt{2}}{\pi}, 1\right]$ for finite samples. Since TIGER solves the SQRT-$Lasso$ problem, and SQRT-$Lasso$ does not depend on unknown parameters or quantities, TIGER may be considered a tuning-insensitive method. Therefore, it is sufficient to cross-validate and finalize the best value from one of the following three values: $\zeta \epsilon \left\{\frac{\sqrt{2}}{\pi}, 0.6, 1\right\}$. All these values produce relatively sparse precision matrices solutions, and the TIGER algorithm runs very smoothly and efficiently as well (Liu and Wang, 2012).

2.2.3 Methods for Tuning based on Log-likelihood

The following methods namely, cross-validation, stability selection and BINCO (Bootstrap inference for network construction) (Li et al, 2013), tune the regularization parameter λ based on the log-likelihood.

2.2.3.1 Cross-Validation

In cross-validation, the data are partitioned into a training set and a validation set, with the latter used to evaluate the out-of-sample predictive power of the model, usually based on the log-likelihood. In $\boldsymbol{k}$-fold cross-validation, the original sample is divided into $\boldsymbol{k}$ subsamples. One sample from each of the $\boldsymbol{k}$ subsamples is withheld in turn as a part of the validation sample set, and the remaining $\boldsymbol{k}$ – 1 subsamples are used as training data. This procedure is repeated $\boldsymbol{k}$ times and then the optimum value of the tuning parameter λ is selected based on maximizing the log-likelihood with respect to the withheld data. Cross-validation reduces the chances of over-fitting.

2.2.3.2 Stability Selection

In Gaussian Graphical models, one of the challenges is the selection of the regularization parameter,λ, for high-dimensional data. Methods like cross-validation, AIC (Akaike Information Criterion), and BIC (Bayesian Information Criterion) can be used with Gaussian graphical models to select the values of λ. Cross-validation,AIC and BIC

produce sparse graphs by using the smaller values of regularization parameter λ when applied on a low-dimensional data. However, they do not produce sparse graphs by using the smaller values of regularization parameter λ when applied to a high-dimensional data.

Stability selection approaches are alternative approaches to cross-validation,AIC and BIC for model selection which aim to select the regularization parameter λ to produce sparse graphs. Stability selection approaches work well both for low-dimensional and high-dimensional settings using the least amount of regularization parameter λ to produce sparse graphs. $Flare$ (family of $Lasso$ regression) R-package (Li et al, 2015) implements stability selection. The $Flare$ R-package has two criteria (CLIME and TIGER) for the selection of non-zero elements in a precision matrix. The optimum precision matrix using $Flare$ R-package is chosen using cross-validation or $Stars$ (Stability approach to regularization selection) (Liu et al, 2010). When the regularization parameter λ is 0, the graph is empty (i.e., a null graph). When we increase the regularization parameter λ, the graph variability increases as well and, as a consequence, the stability decreases.

The $Stars$ procedure is as follows:

Let $a = a(n)$ be such that $1 < a(n) < n$. Now we randomly draw *without* replacement N subsamples of size a from $X_1, \ldots, X_n$. The subsamples, denoted as $S_1, \ldots, S_N$, have a total sample size of $\binom{n}{a}$ across all subsamples. Note that $Stars$ differs from bootstrapping in that it uses sampling *without* replacement (Efron, 1982). Graphical $Lasso$ (Friedman et al; 2008) is used to construct a graph for each subsample for each regularization parameter λ, producing N estimated edge matrices.

Let us focus now on one edge (s, t) and one value of regularization parameter λ. Let $\xi_{st}^{a}(\lambda)$ denotes the instability of an edge across the subsamples; and $0 \leq \xi_{st}^{a}(\lambda) \leq 1/2$. The total instability of a graph is obtained by averaging over all the edges:

$$\widehat{D}_a(\lambda) = \frac{\sum_{s<t} \hat{\xi}_{st}^{a}}{\binom{p}{2}} \qquad (2.14)$$

It is evident that $\widehat{D}_a\ (0) = \ 0$ at the boundary $0 \leq \xi_{st}^{a}(\lambda) \leq \frac{1}{2}$. In addition, $\widehat{D}_a(\lambda)$ will increase with increasing values of λ. However, dense graphs are produced for very large values of λ. To obtain stable sparse graphs, $\widehat{D}_a(\lambda)$ is monotonized by defining:

$$\overline{D}_a(\lambda) = \ sup_{0 \leq t \leq \lambda}\, \widehat{D}_a(t) \tag{2.15}$$

The regularization parameter λ is chosen by $Stars$ by defining:

$$\hat{\lambda}_s = sup\{\lambda : \ \overline{D}_a(\lambda) \leq \ \alpha\ \} \tag{2.16}$$

where, α is a specified cut-off value. The value of α is set to an arbitrary value, of 0.05. $Stars$ is based on subsampling; sample size a instead of n is used for the estimation of selected graph. One of the disadvantages of $Stars$ is its efficiency loss for low-dimensional data, (i.e. data with a small number of features p). However, this efficiency loss decreases as the dimension increases.

2.2.3.3 Bootstrap Inference for Network Construction

Traditional methods for model selection, such cross-validation and BIC (Bayesian Information Criterion), work by minimizing the prediction error or maximizing a penalized likelihood function. They do not explicitly estimate and control the FDR (false discovery rate) (Li et al, 2013). Li et al. (2013) proposed a frequentist approach, BINCO (bootstrap inference for network construction) that directly controls the $FDRs$ in the selection of edges. BINCO was proposed for network inference of high-dimensional data. Let $A(\lambda)$ is an edge-selection procedure having λ as a regularization parameter. Let a $p-$ dimensional random vector $\boldsymbol{Y} = \left(Y_1, \dots, Y_p\right)$ that follows a multivariate normal distribution $N(0, \Sigma)$, where Σ is a positive definite matrix with dimension $p \times p$. When $A(\lambda)$ is applied to data $\boldsymbol{Y}$, we obtain the set of selected edges:

$$S^{\lambda}(\boldsymbol{Y}) \equiv S^{\lambda}(A(\lambda), \boldsymbol{Y}) \tag{2.17}$$

Let p_{ij} be the probability of edge selection given as:

$$p_{ij} = E\left(\boldsymbol{I}\left\{(i,j)\epsilon S^{\lambda}(\boldsymbol{Y})\right\}\right) \tag{2.18}$$

where, $I\{.\}$ represents the indicator function.

Let us denote $R(Y)$ as the space of resamples obtained by doing bootstrapping or subsampling on Y. If $\acute{Y}$ represents a random resample from $R(Y)$, then:

$$\tilde{p}_{ij} = E\left(\boldsymbol{I}\left\{(i,j)\epsilon S^{\lambda}\left(\acute{\boldsymbol{Y}}\right)\right\}\right) = E\left(E\left(\boldsymbol{I}\left\{(i,j)\epsilon S^{\lambda}\left(\acute{\boldsymbol{Y}}\right)\right\}\middle|\boldsymbol{Y}\right)\right) \tag{2.19}$$

p_{ij} and $\tilde{p}_{ij}$ are very close in many cases (Li et al, 2013). p_{ij} can be estimated by selection frequency X_{ij} as follows:

$$X_{ij}^{\lambda} \equiv X_{ij}\left(A(\lambda); \boldsymbol{Y}^{1}, \ldots, \boldsymbol{Y}^{B} \epsilon\, R(\boldsymbol{Y})\right) \tag{2.20}$$

$$= \frac{1}{B}\sum_{k=1}^{K} \boldsymbol{I}\left\{(i.j)\epsilon S^{\lambda}(\boldsymbol{Y}^{k})\right\}, \quad 1 \le i < j \le p$$

where, B represents the number of resamples. The B resamples $Y^{1}, \ldots, Y^{B}$ are aggregated in the calculation of selection frequencies expressed in Equation (2.21).

The edges of large selection frequencies are chosen using S_c^{λ} for the aggregation-based procedures as follows:

$$S_c^{\lambda} = \left\{(i,j): X_{ij}^{\lambda} \ge c\right\} \qquad for\ c\ \epsilon\ [0,1] \tag{2.21}$$

If most of the selection frequencies for the true edges are greater than or equal to c, S_c^{λ} is reasonable. The null edges have selection frequencies less than c. The estimation of $FDRs$ is done by fitting a mixture model for edge selection frequency distribution in Equation (2.22). The calculation of selected frequencies is done using model aggregation as explained in Equation (2.21). The mixture density of X_{ij}^{λ} is $f^{\lambda}(x) = (1-\pi)f_0^{\lambda}(x) + \pi f_1^{\lambda}(x), x\epsilon\{0, {}^{1}/_{B}, {}^{2}/_{B}, \ldots, 1\}$, where $f_0^{\lambda}(x)$ and $f_1^{\lambda}(x)$ are the densities of X_{ij}^{λ} if it belongs to null and true categories respectively. The (positive) FDR of S_c^{λ} ; based on the mixture model is as below:

$$FDR\left(S_c^{\lambda}\right) = Pr\left((i,j)\epsilon E^{c}\middle|(i,j)\epsilon S_c^{\lambda}\right) = \frac{\sum_{x\ge c}(1-\pi)f_0^{\lambda}(x)}{\sum_{x\ge c} f^{\lambda}(x)} \tag{2.22}$$

Here E is the edge set, c is the threshold, π is the proportion of true edges, and λ is the regularization parameter.

The estimation of true edges number in S_c^λ is done as follows:

$$\widehat{N}_E(S_c^\lambda) = |S_c^\lambda| \left(1 - \widehat{FDR}(S_c^\lambda)\right) \tag{2.23}$$

where, $\widehat{FDR}(S_c^\lambda)$ is an estimate of $FDR(S_c^\lambda)$. For different values of c and λ, it can be used to compare the achieved power of S_c^λ when the total number of true edges is a constant.

For each λ, the optimal threshold c is calculated using the following formula:

$$c^*(\lambda) = min\{c: \widehat{FDR}(S_c^\lambda) \leq \alpha\} \tag{2.24}$$

where, α is the targeted FDR level.

The calculation of optimal regularization parameter λ is done using the following formula:

$$\lambda^* = \underset{\lambda \epsilon \Lambda}{argmax}\, \widehat{N}_E(S_{c*(\lambda)}^\lambda) \tag{2.25}$$

where, $S_{c^*(\lambda^*)}^{\lambda^*}$ indicates maximal power, while the FDR does not exceed the specified target level of α. BINCO performed well in controlling the FDR and gaining decent power even when the data were generated from non-normal distributions (t-distribution and Uniform distribution).

2.3 Bayesian Approaches

The estimation of precision matrices can also be done using Bayesian approaches. In certain methods, some elements may be inferred to be zero, producing a sparse precision matrix (graph). I describe the Bayesian approaches for model selection in the following subsections.

2.3.1 Model Search in Discrete Space

Bayesian model selection approaches to inferring Σ, or equivalently Ω, usually specify the prior over Σ, conditional on graph G, to be the hyper-inverse Wishart (HIW). This is a conjugate prior, with its support restricted to Σ corresponding to symmetric positive

definite Ω with zero elements as dictated by G. Many authors take this to be a multiple of the identity: $\mathbf{\Sigma} \sim HIW_G(\delta, \tau \boldsymbol{I})$. However, a g-prior approach $(\mathbf{\Sigma}|G) \sim HIW_G(gn, gX^TX)$, is also possible (Scott and Carvalho, 2008).

The G-Wishart over Ω has the following density:

$$p(\mathbf{\Omega}|G, b, D) = I_G(b, D)^{-1}|\mathbf{\Omega}|^{\frac{b-2}{2}} exp\left\{\frac{-1}{2}\ tr\ (\Omega D)\right\}, \quad \Omega \in P_G \tag{2.26}$$

where, the parameter $b > 2$ represents degrees of freedom, the positive definite symmetric matrix D has a dimension of $p\ x\ p$, the normalizing constant is represented by I_G , and the set of all positive definite symmetric matrices having dimension $p\ x\ p$ and ω_{ij} = 0 are represented by P_G. The normalizing constant has closed form only for the subclass of decomposable (triangulated) graphs, making this subclass much more computationally tractable than the non-decomposable graphs.

A uniform prior over graphs places most of the mass on graphs with intermediate numbers of edges, but sparsity can be encouraged with the use of further priors. A common choice is the Bernoulli prior on the number of edges $e: \pi^e(1-\pi)^{t-e}$, where t is the total possible number of edges and π is set to encourage sparsity. Scott and Carvalho (2008) place a uniform prior on π, and show it can be integrated out.

Model selection requires to perform a search in a discrete space that is relatively difficult compared to model search in a continuous space. Stochastic search methods are used to search model spaces which are very high-dimensional. Jones et al. (2005) discussed stochastic computation methods in Gaussian graphical models in detail for their computational efficiency, performance, and scalability with dimension. Their proposed stochastic search method was compared with the MCMC (Markov chain Monte Carlo) for moderate (12-20) to large (150) numbers of variables. A Bernoulli prior over the graph space determined the prior probability of inclusion of an edge in a graph to produce sparser graphs. MH (Metropolis-Hastings), SSS (Shotgun stochastic search) algorithms were considered both for decomposable models and unrestricted models for the comparison of run time and quality of best graph (based on the computation of an entire likelihood and perfect ordering of each proposed graph) between them. The SSS

(decomposable) had the lowest run time (in hours) and the highest log posterior (log of the posterior) as compared to MH (decomposable) and SSS (Unrestricted) for moderate dimensional data. For unrestricted models, the SSS algorithm performed better as compared to the MCMC algorithm.

FINCS (Future-inclusion stochastic search) method (Scott and Carvalho, 2008) is a search algorithm shown to achieve better model selection in terms of prediction than $Lasso$-based approaches, especially in case of higher dimensions. FINCS is used for an edge addition or deletion at a time t. The addition or deletion of an edge is done by posterior probability of inclusion of an edge at step t using the following formula:

$$\hat{q}_{ij}(t) = \frac{\sum_{k=1}^{k=t} 1_{(i,j)\epsilon\mathcal{G}_k} P(X|\mathcal{G}_k)\pi(\mathcal{G}_k))}{\sum_{k=1}^{k=t} P(X|\mathcal{G}_k)\pi(\mathcal{G}_k))} \tag{2.27}$$

where, $\pi(\mathcal{G}_k)$ is the graph's prior probability. FINCS retains the models list based on higher posterior probability of inclusion of an edge at time t. $P(X|\mathcal{G}_k)$ is calculated using the following formula:

$$p(X|\mathcal{G}) = (2\pi)^{\frac{-np}{2}} \frac{h(\mathcal{G},gn,gX^TX)}{h(\mathcal{G},n,X^TX)} \tag{2.28}$$

where, the sum of squares of data matrix X are represented by X^TX and; g (the g-prior) is set to $1/n$; and h represents the hyper-inverse Wishart distribution normalizing constant.

Fitch et al. (2014) studied the behaviour of Bayesian methods of model selection for decomposable Gaussian graphical models for which the true model was non-decomposable. A superset graph is a graph that includes all the edges of an undirected graph $\mathcal{G} = (V, E)$ plus one other edge at least that is not in the edge set E. It was shown that the Bayesian procedures converged to minimal supersets (i.e., minimum number of extra edges required in a non-decomposable graph to achieve decomposability) of the true graph. FINCS-based methods were compared by Fitch et al. (2014) with $Lasso$-based methods on the basis of precision (positive predictive value) and recall (sensitivity, or true positive rate), Kullback-Leibler divergence criterion, and sum of squared prediction errors. FINCS - based methods had consistently higher precision than the precision of

$Lasso$ - based methods. However, for small sample sizes, the recall of the $Lasso$ - based methods were almost the same as the recall of the FINCS – based methods. FINCS-based methods produced lower Kullback-Leibler divergence as compared to the Kullback-Leibler divergence of the $Lasso$ -based methods. The sum of squared prediction errors was low for FINCS-based methods as compared to the sum of squared prediction errors for the $Lasso$ - based methods. Thus, overall, FINCS-based methods produced superior results as compared to the $Lasso$ - based methods, especially for higher dimensional data.

Generation of a posterior distribution of graphs is computationally intensive and involves exploring a large discrete space. Mohammadi and Wit (2015) proposed a unique Bayesian framework for the determination of a Gaussian graphical model based on a continuous time birth-death process. In the continuous time birth-death process, birth is regarded as an appearance of an individual and death is regarded as the removal of an individual. They designed a Bayesian algorithm, BDMCMC (birth-death MCMC), to perform parameter estimation and graph structure learning as a birth-death process. FINCS (Scott and Carvalho, 2008) is used to add and delete edges based on posterior probability of the inclusion of that edge at each iteration t in Equation (2.27). However, BDMCMC adds an edge in the event of a birth (appearance of an individual), and removes an edge in the event of a death (removal of an individual). They modelled the birth and death rates using independent Poisson processes, and the time between the two successive events with an exponential distribution. The probability of the next birth or death event is calculated as follows:

$$P(birth\ for\ edge\ e) = \frac{\beta_e(K)}{\beta(K)+\delta(K)}, \qquad for\ each\ e \in \bar{E} \qquad (2.29)$$

$$P(death\ for\ edge\ e) = \frac{\delta_e(K)}{\beta(K)+\delta(K)}, \qquad for\ each\ e \in E \qquad (2.30)$$

where, K is the precision matrix, $\beta_e(K)$ is the birth rate in a precision matrix, $\delta_e(K)$ is the death rate in a precision matrix, $\beta(K)$ is the overall birth rate, and $\delta(K)$ is the overall death rate.

Let us consider the birth rates $\beta_e(K)$ and death rates $\delta_e(K)$ as follows:

$$\beta_e(K) = \frac{P(G^{+e}, K^{+e} \setminus (k_{ij}, k_{jj})/X)}{P(G, K \setminus k_{jj}/X)}, \qquad for\ each\ e \in \bar{E} \tag{2.31}$$

$$\delta_e(K) = \frac{P(G^{-e}, K^{-e} \setminus k_{jj}/X)}{P(G, K \setminus (k_{ij}, k_{jj})/X)}, \qquad for\ each\ e \in E \tag{2.32}$$

The process jumps to a new state; G^{-e}, K^{-e} according to Equation (2.32) in the event of a death; and, G^{+e}, K^{+e} according to Equation (2.31) in the event of a birth.

The sum of the birth rates $\beta_e(K)$ and the death rates $\delta_e(K)$ are expressed as follows:

$$\beta(K) = \sum_{e \in \bar{E}} \beta_e(K) \tag{2.33}$$

$$\delta(K) = \sum_{e \in E} \delta_e(K) \tag{2.34}$$

Therefore, the proposed BDMCMC algorithm (Mohammadi and Wit, 2015) calculates the birth rates according to Equation (2.31), the death rates according to Equation (2.32), aggregate the birth rates according to Equation (2.33) and the death rates according to Equation (2.34) in the first step, calculates the waiting times $w(K) = {1}/{\left(\beta(K) + \delta_e(K)\right)}$ in the second step, simulates the jump type (birth or death) by Equation (2.29) and Equation (2.30) in the third step; and then samples from the new inverse covariance matrix on the basis of the type of jump. In this way, the BDMCMC algorithm samples from the posterior distribution of the inverse covariance matrix (K).

As the dimension grows, the BDMCMC efficiently eliminates the problems of convergence, computes the prior normalizing constant, and generates samples from the posterior distribution of the inverse covariance matrix. In contrast, the Bayesian graph structure learning in Gaussian graphical models has problems with these processes as the dimension is increased. The BDMCMC algorithm is extremely efficient and computationally fast because it always accepts moves between the models, contrary to the reverse-jump MCMC algorithms of (Wang and Li, 2012; Lenkoski, 2013; Cheng et al, 2012).

2.3.2 Bayesian Graphical $Lasso$ and Bayesian Adaptive Graphical $Lasso$

Laplace prior distributions may be used in a Bayesian setting in the same way as the $Lasso$-type penalties are used in a penalized-likelihood setting. The estimates of $Lasso$ match the Bayesian MAP (*maximum a posteriori*) estimates when we place DE (double exponential) priors over regression coefficients (with fixed λ) (Tibshirani, 1996). This is also true for graphs.

Wang (2012) used the graphical $Lasso$ priors with a double exponential density as follows:

$$p\left(\boldsymbol{y}_i|\boldsymbol{\Omega}\right) = N\left(\boldsymbol{y}_i|0, \boldsymbol{\Omega}^{-1}\right), \quad (i = 1, \ldots, n), \tag{2.35}$$

$$p\left(\boldsymbol{\Omega}|\lambda\right) = C^{-1} \prod_{i<j}\left\{DE\left(\omega_{ij}|\lambda\right)\right\} \prod_{i=1}^{p}\left\{EXP\left(\omega_{ii}\middle|\frac{\lambda}{2}\right)\right\} 1_{\boldsymbol{\Omega}\,\epsilon\,\boldsymbol{M}^{+}}$$

The off-diagonal elements of Ω are shrunk towards zero by double exponential (DE) density function in the Bayesian graphical $Lasso$ Equation (2.35). The term $EXP\left(\omega_{ii}\middle|\frac{\lambda}{2}\right)$ represents the exponential density function. Here C is the normalizing constant and λ is the tuning parameter. The adaptive version of the Bayesian graphical $Lasso$ they proposed is:

$$p\left(\boldsymbol{y}_i|\boldsymbol{\Omega}\right) = N\left(\boldsymbol{y}_i|0, \boldsymbol{\Omega}^{-1}\right), \quad (i = 1, \ldots, n), \tag{2.36}$$

$$p\left(\boldsymbol{\Omega}\middle|\left\{\lambda_{ij}\right\}_{i\leq j}\right) = C^{-1}_{\left\{\lambda_{ij}\right\}_{i\leq j}} \prod_{i<j}\left\{DE\left(\omega_{ij}|\lambda_{ij}\right)\right\} \prod_{i=1}^{p}\left\{EXP\left(\omega_{ii}\middle|\frac{\lambda_{ii}}{2}\right)\right\} 1_{\boldsymbol{\Omega}\epsilon M^{+}},$$

$$p\left(\left\{\lambda_{ij}\right\}_{i<j}\middle|\left\{\lambda_{ii}\right\}_{i=1}^{p}\right) \propto C_{\left\{\lambda_{ij}\right\}_{i\leq j}} \prod_{i<j} GA(r,s)$$

where, $GA\left(r, s\right)$ represents the Gamma hyper-prior on the tuning parameters λ_{ij}; where the hyper-parameters r and s are fixed. The hyper-parameters for the diagonal elements are denoted by $\{\lambda_{ii}\}_{i=1}^{p}$. Over-penalization of large effects is a drawback of the Bayesian graphical $Lasso$, which is avoided in the Bayesian adaptive graphical $Lasso$. Applying a prior over λ does not produce sparse posterior mean $\bar{\Sigma}$ in the Bayesian graphical $Lasso$.

2.3.3 Some Bayesian Methods for Edges Selection

Some other Bayesian methods for selection of edges in a precision matrix have been proposed in the literature. Peterson et al. (2013) consider that the edges in the precision matrix are connected if the absolute partial correlation between the edges is greater than 0.1. This method of selection is considered arbitrary.

Wang (2012) suggested that the edges in the precision matrix were connected if and only if the estimate of posterior mean of inverse covariance matrices using graphical $Lasso$ priors and the expected value of the posterior mean of inverse covariance matrices using $\boldsymbol{W(3, Ip)}$ had a ratio greater than 0.5 (an arbitrary value). Here $\boldsymbol{W(3, Ip)}$ is the standard conjugate Wishart prior. Like that of (Peterson et al, 2013), this method of selection is considered arbitrary as well. One more approach was considered where the edges were connected if the 95% credible interval of the posterior sample of the precision matrix did not include 0. Wang (2012) proposed an algorithm named block Gibbs sampler for covariance matrices simulations. Wang (2012) also compared the performance of their proposed Bayesian adaptive graphical $Lasso$ method with the Bayesian graphical $Lasso$ and the frequentist approaches such as the graphical $Lasso$, the adaptive graphical $Lasso$ and SCAD (Smoothly clipped absolute deviations) (Fan et. al, 2009). Wang (2012) considered $n = 50, p = 30$ and $n = 200, p = 100$ scenarios. 10,000 simulations with 5,000 burn-ins were considered. The performance of the five methods was judged based on sensitivity, specificity and MCC. The proposed Bayesian adaptive $Lasso$ method performed well based on higher specificity and MCC as compared to the Bayesian graphical $Lasso$, the graphical $Lasso$, the adaptive graphical $Lasso$ and SCAD .

Peterson et al. (2013) proposed the Bayesian adaptive graphical $Lasso$ with informative priors for the estimation of precision matrix.

The hyper-prior on the shrinkage parameters is as follows:

$$p\left(\left(\{\lambda_{ij}\}_{i<j}\middle|\{\lambda_{ii}\}_{i=1}^{P}\right)\right) \propto C \prod_{i<j} \underbrace{\frac{s_{ij}^{r}\,\lambda_{ij}^{r-1}\exp\{-\lambda_{ij}s_{ij}\}}{\Gamma r}} \tag{2.37}$$

where, the term $\underbrace{\frac{s_{ij}^{r}\,\lambda_{ij}^{r-1}\exp\{-\lambda_{ij}s_{ij}\}}{\Gamma r}}$ is Gamma (r, s_{ij}) prior on λ_{ij} ; and s_{ij} are the hyper-parameters allowing the incorporation of edge specific information.

2.3.4 Model Selection through Shrinkage and Selection

The most commonly used priors for model shrinkage and selection in a Bayesian setting are the Laplace prior and the Bernoulli prior respectively. The Laplace prior shrinks some of the off-diagonal partial correlations of a precision matrix, but does not set them to exact zeros, so it is not considered a method of model selection. Rajesh Talluri et al. (2014) combined these strategies (Laplace and Bernoulli priors) and proposed various Bayesian graphical methods (adaptive GGMs and mixtures of GGMs) for model selection and parameter estimation simultaneously. The parameterization was divided into shrinkage and selection.

Rajesh Talluri et al. (2014) used the Laplace prior on the shrinkage matrix (R) elements defined as:

$$f\left({R_{ij}}/{\tau_{ij}}\right) \propto \frac{1}{2\tau_{ij}} exp\left(-\frac{|R_{ij}|}{\tau_{ij}}\right) \tag{2.38}$$

where, the level of sparsity was controlled by τ_{ij}.

The shrinkage matrix (R) elements variable selection was performed by the selection matrix (A). The off-diagonal elements of the selection matrix (A) were the binary variables (either 0 or 1). The exchangeable Bernoulli prior was used on the off-diagonal elements of selection matrix (A) defined as:

$${A_{ij}}/{q_{ij}} \sim Bernoulli(q_{ij}), i < j \tag{2.39}$$

where, q_{ij} represented the selection probability of 1 for the ij^{th} element**.**

The joint prior for A and R was expressed as follows:

$${R_{ij}\,, A_{ij}}/{\tau_{ij}\,, q_{ij}} \sim Laplace\,(0\,, \tau_{ij}) Bernoulli(q_{ij}) I(\boldsymbol{C} \in \mathbb{C}_p) \tag{2.40}$$

where, $-1 \leq R_{ij} \leq 1,\ 0 \leq q_{ij} \leq 1\ and\ I\left(\boldsymbol{C} \in \mathbb{C}_{p_p}\right) = 1$ in case C was a correlation matrix and 0 elsewhere.

The Hadamard product between the selection matrix (A) and the shrinkage matrix (R) should be equal to the correlation matrix $(C = A \odot R)$ to fulfil positive definite constraint for joint prior specification. The implementation of posterior inference in the proposed models was done through posterior simulation schemes like Gibbs sampling and MCMC sampling using the proposed adaptive Bayesian model and the non-adaptive fit (Fraley & Raftery, 2007). The validation of the proposed Bayesian models was done by comparing these with the graphical $Lasso$ and MB (Meinshausen & Buhlmann, 2006) methods on genomics dataset, with the proposed Bayesian methods outperforming the graphical $Lasso$ and MB in terms of Kullback-Leibler loss.

2.3.5 Decoupled Shrinkage and Selection for Regression

Variable selection in regression is a long studied problem; the graphical $Lasso$, the adaptive graphical $Lasso$, and Bayesian variable selection methods were all preceded by analogous regression methods. Hahn & Carvalho (2015) proposed a DSS (decoupled shrinkage and selection) loss function, which is a posterior variable selection summary approach, as follows:

$$\mathcal{L}(\gamma) = \lambda\|\gamma\|_0 + n^{-1}\left\|\boldsymbol{X}\bar{\beta} - \boldsymbol{X}\gamma\right\|_2^2 \qquad (2.41)$$

Here $||\gamma||_0$, is the parsimony - encouraging penalty (counting penalty in which non-zero elements in a vector are counted which is called the L_0– norm) that distinguishes DSS from $Lasso$, γ are the model selection vectors**;** and $\bar{\beta}$ is the posterior mean of β**.** The optimal solution β_λ to the DSS loss function is as follows:

$$\beta_\lambda \equiv arg\ min_\gamma\ \ \lambda\|\gamma\|_0 + n^{-1}\left\|\boldsymbol{X}\bar{\beta} - \boldsymbol{X}\gamma\right\|_2^2 \qquad (2.42)$$

Hahn & Carvalho (2015) approximated this with the adaptive $Lasso$ solution. DSS distils the full posterior distribution into a sequence of sparse linear predictors. DSS summary plots were given for different datasets using different priors like the horseshoe prior, a g-

prior (Zellner, 1986), where $g = n$, and the robust prior (Bayarri et. al, 2012). Selection of the number of variables was done by plotting the expected value and 90% credible intervals for posterior samples of ρ_λ^2 (variation-explained in X) against the model size and to compare the fit of the sparsified vector (posterior mean) to "typical" posterior samples. Models selected on the basis of DSS outperformed the median probability model in both fit and sparsity.

2.3.6 Proposed Method: Decoupled Shrinkage and Selection in a Graph

Hahn & Carvalho (2015) proposed the DSS method for regression. Here, I extend the DSS method for estimation of edges in a precision matrix in Gaussian graphical models.

Let $\Omega = \Sigma^{-1}$ and Γ be the estimate of $\Omega = \Sigma^{-1}$. Let the future observations are denoted by $\tilde{X}$ and the sample size is denoted by n^*. The log-likelihood is an efficient method to find out the fit (predictive accuracy) of n^* future observations $\tilde{X}$. The expectation of $\frac{\tilde{X}^T \tilde{X}}{n^*}$ is equivalent to the covariance matrix posterior mean denoted by $\bar{\Sigma}$. The expected fit of Γ is as follows:

$$E[fit(\boldsymbol{\Gamma})] = E\left[logdet(\boldsymbol{\Gamma}) - tr\left(\frac{\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{X}}\boldsymbol{\Gamma}}{n^*}\right)\right] \qquad (2.43)$$
$$= logdet(\boldsymbol{\Gamma}) - tr(\bar{\boldsymbol{\Sigma}}\,\boldsymbol{\Gamma})$$

The expected fit is maximized at $\Gamma = \bar{\Sigma}^{-1}$ (Figure 2.5).

The $\mathrm{fit}(\boldsymbol{\Gamma}) = \mathrm{logdet}(\boldsymbol{\Gamma}) - \mathrm{tr}\left(\frac{\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{X}}\boldsymbol{\Gamma}}{n^*}\right)$ is a random variable that is controlled by the posterior predictive distribution of $\tilde{X}$. Let Σ_k denote the posterior samples of covariance matrices of future observations. Then:

$$fit(\bar{\boldsymbol{\Sigma}}^{-1}|\boldsymbol{\Sigma_k}) = log\left(det(\bar{\boldsymbol{\Sigma}}^{-1})\right) - tr(\boldsymbol{\Sigma_k}\bar{\boldsymbol{\Sigma}}^{-1}) \qquad (2.44)$$

The above Equation (2.44) denotes a sample from distribution of $fit(\bar{\Sigma}^{-1})$.

where, $k \in 1,2, \ldots, K$. The next thing is to judge the acceptable fit. If the estimate with the best expected fit will have actual fit below F, say, 5% of the time, a sparsified choice for

$\boldsymbol{\Gamma}$ with expected fit F should be considered adequate. The choice of F= 5% is arbitrary, and can be set by the user.

Once we have the posterior samples Σ_k and $\Omega_k = \Sigma_k^{-1}$, then the procedure for the proposed DSS credible interval approach is as follows:

1. Calculate the mean of Σ_k , $\overline{\Sigma}$, and $\overline{\Sigma}^{-1}$. The Bayes estimate of Ω is denoted by $\overline{\Sigma}^{-1}$ which is not sparse.

2. Let the future data be represented by the sampled Σ_k. And then calculate the $\overline{\Sigma}^{-1}$ fit for each future dataset. Compute 5% quantile of these fits.

3. Calculate the P credible intervals for ω_{ij} which are based on sampled Ω_k.

4. Find the elements where the credible intervals include 0. For this location, constrain Γ_P to be 0.

5. Find Γ_P that maximizes $logdet\ (\Gamma_P) - tr(\overline{\Sigma}\Gamma_P)$ (see below)

6. If $fit\left(\frac{\Gamma}{\overline{\Sigma}}\right)$ = $logdet\ (\Gamma_P) - tr(\overline{\Sigma}\Gamma_P)$ is greater than 5% quantile of $fit\left(\frac{\overline{\Sigma}^{-1}}{\Sigma_k}\right)$ and; $P < 100\%$, increase the value of P and come back to step 3.

The steps 1 and 2 are the same for the other proposed DSS approaches. The different steps for the proposed DSS adaptive graphical $Lasso$ approach are as follows:

3. Compute ω_{ij} for different values of the tuning parameter λ which are based on sampled Ω_k.

4. Find Γ_A that maximizes $logdet\ (\Gamma_A) - tr(\overline{\Sigma}\Gamma_A)$ (see below)

5. If $fit\left(\frac{\Gamma}{\overline{\Sigma}}\right)$ = $logdet\ (\Gamma_A) - tr(\overline{\Sigma}\Gamma_A)$ is greater than 5% quantile of $fit\left(\frac{\overline{\Sigma}^{-1}}{\Sigma_k}\right)$, increase the value of λ and come back to step 3.

The different steps for the proposed DSS ρ threshold approach are as follows:

3. Compute the absolute partial correlations $|\rho|$ of ω_{ij} which are based on sampled Ω_k.

4. Find Γ_{ABS} that maximizes $logdet\ (\Gamma_{ABS}) - tr(\bar{\Sigma}\Gamma_{ABS})$ (see below)

5. If $fit\left(\frac{\Gamma}{\bar{\Sigma}}\right) = logdet\ (\Gamma_{ABS}) - tr(\bar{\Sigma}\Gamma_{ABS})$ is greater than 5% quantile of $fit\left(\frac{\bar{\Sigma}^{-1}}{\Sigma_k}\right)$, increase the value of $|\rho|$ and come back to step 3.

After the above steps, we choose the sparsest estimate of Γ with a suitable fit based on the log-likelihood (Figure 2.5).

Step (5) is carried out using the graphical $Lasso$ algorithm (Friedman et al, 2008).

$$maximize_{\mathbf{\Omega}}\{log\ det\ \mathbf{\Omega} - trace(\mathbf{\Sigma}\ \mathbf{\Omega}) - \lambda\|\mathbf{\Omega}\|_1\}$$

When Σ is positive definite, we can set $\lambda = 0$ in my proposed algorithm in order to find the best fitted Γ which follows a specified 0 pattern. $\bar{\Sigma}$ will always be positive definite because I consider the posterior distributions which are over positive definite matrices. I use cross-validation to select the tuning parameter in the graphical $Lasso$ algorithm, and to generate the starting point of inverse covariance matrix in the Bayesian adaptive graphical $Lasso$ MCMC algorithm.

The proposed DSS methodology differs from Hahn & Carvalho's (2015) methodology in the following ways:

1. Hahn & Carvalho (2015) obtained posterior samples of regression coefficients matrices and then calculated the posterior mean denoted by $\bar{\beta}_i$. However, we obtained the posterior samples of Σ_k and $\Omega_k = \Sigma_k^{-1}$, and then calculated their means $\bar{\Sigma}$, and $\bar{\Sigma}^{-1}$.

2. Hahn & Carvalho (2015) plotted 90% credible intervals for posterior samples of ρ_λ^2 (variation-explained in X) sparsified vector (posterior mean) to "typical" posterior samples. However, we plotted the sparsified versions of precision matrices for different strategies against the range of fit of posterior mean inverse $\bar{\Sigma}^{-1}$.

2.4 Methodological Comparison

In this section, I compare the proposed method with alternatives proposed in the literature. Comparisons were made using simulated data and real data.

2.4.1 Example: Simulated Data

In this section, I assess the performance of the proposed decoupled shrinkage and selection method (DSS) and compare it to alternative methods—specifically, thresholding the absolute partial correlations (Peterson et al, 2013), credible interval method (edges are connected if the 95% credible interval of the posterior sample of precision matrix does not include 0), and ratio selection method (Wang, 2012)—using simulated data with known properties. Methods were compared on the basis of sensitivity (true positive rate) and specificity (false positive rate).

2.4.1.1 Data Generation and Simulation

The data generation and simulation were done as follows:

1. Generate data matrices y of size $n = 50$ with dimension $p = 30, n = 100$ with dimension $p = 100, n = 200$ with dimension $p = 100$, and $n = 300$ with dimension $p = 100$. This was done using the MatLab code proposed by Wang (2012).
2. Fit the Bayesian adaptive graphical $Lasso$, with $r = 10^{-2}$ and $s = 10^{-6}$ as the prior distributions of λ_{ij} with $\lambda_{ii} = 1$ for $i = 1, \ldots, p$, to obtain the posterior covariance matrices, which are not sparse.
3. Draw 10,000 samples from the dense posterior covariance matrices, thinned to 1,000, with 5,000 burn-ins. Two simulation scenarios were considered for this purpose, which are as follows:

    a) Simulation 1: An $AR\,(2)$ model having $\omega_{ii} = 1$, $\omega_{i,i-1} = \omega_{i-1,1} = 0.5$, and $\omega_{i,i-2} = \omega_{i-2,1} = 0.25$. Just to give an idea of the structure of true inverse covariance matrix (Ω), we present it for $p = 10$ as follows:

$$\begin{pmatrix} 1 & 0.5 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 1 & 0.5 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.25 & 0.5 & 1 & 0.5 & 0.25 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0.5 & 1 & 0.5 & 0.25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.25 & 0.5 & 1 & 0.5 & 0.25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.25 & 0.5 & 1 & 0.5 & 0.25 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.25 & 0.5 & 1 & 0.5 & 0.25 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.25 & 0.5 & 1 & 0.5 & 0.25 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0.5 & 1 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0.5 & 1 \end{pmatrix} \quad (2.45)$$

b) Simulation 2: A $Star$ model in which only the first node is connected to every node. And $\omega_{ii} = 1$, $\omega_{1,i} = \omega_{i,1} = 0.1$ and $\omega_{ij} = 0$ elsewhere.

$$\begin{pmatrix} 1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0.1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0.1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.46)$$

My main motivation was to sparsify the posterior mean in a way that did not degrade the fit.

2.4.1.2 Results

Stability selection using TIGER had better sensitivity both for $AR(2)$ and $Star$ cases (Table 2.1); when $n = 50$ and $p = 30$. However, the performance measures for the DSS adaptive graphical $Lasso$, DSS credible interval, and DSS ρ threshold methods were more or less the same for $Star$ case when $n = 50$ and $p = 30$.. For $n = 100$ and $p = 100$, the DSS ρ threshold method performed well for $AR(2)$ case; and DSS adaptive graphical $Lasso$ method performed well for the $Star$ case. For $n = 200$ and $p = 100$, the DSS ρ threshold method performed well for $AR(2)$ case; and DSS adaptive graphical

$Lasso$ method performed well for $Star$ case. For $n = 300$ and $p = 100$, the DSS ρ threshold method performed well for $AR(2)$ case; and DSS adaptive graphical $Lasso$ method performs well for $Star$ case.

Table 2.1: Multivariate normal data was simulated from two different model structures, $AR(2)$ and $Star$, for each of the p; n combinations given. BAGL posterior samples were generated, and three different sparsification strategies applied, to select the sparsest model with fit above the 5% fit quantile. Fifty replicates are performed; average sensitivity and specificity for each scenario are given as percentages.

	$AR(2)$		$Star$	
	Sensitivity (%)	Specificity (%)	Sensitivity (%)	Specificity (%)
		$n = 50, p = 30$		
DSS adaptive Graphical $Lasso$	25.6	99.7	-	100.0
DSS credible Interval	29.0	100.0	-	100.0
DSs ρ Threshold	28.0	100.0	0.1	100.0
TIGER	38.3	97.9	0.55	99.8
		$n = 100, p = 100$		
DSS adaptive Graphical $Lasso$	77.0	98.0	99.9	97.5
DSS credible Interval	84.4	98.4	73.9	94.4
DSs ρ Threshold	87.1	98.5	47.9	92.8
TIGER	52.0	99.3	35.4	100.0
		$n = 200, p = 100$		
DSS adaptive Graphical $Lasso$	95.1	99.3	100.0	98.9
DSS credible Interval	97.9	99.5	90.0	97.0
DSs ρ Threshold	98.3	99.6	78.0	94.0
TIGER	79.6	99.3	69.4	100.0
		$n = 300, p = 100$		
DSS adaptive Graphical $Lasso$	98.4	99.7	100.0	99.8
DSS credible Interval	99.4	99.9	96.7	98.5
DSs ρ Threshold	99.5	99.9	92.0	96.5
TIGER	94.8	99.2	82.5	99.9

In summary, for most of the sample sizes and dimensions examined here, the DSS ρ threshold method generally performed best for $AR(2)$ case, whereas, the DSS adaptive graphical $Lasso$ methods performed well for the $Star$ case.

For comparisons of predictive accuracy for future data across methods, we present box plots of the log-likelihood (Equation 2.43), for $n = 200, p = 100$ for $AR(2)$ and $Star$ cases, evaluated for new data, simulated using the true covariance matrix (Figures 2.1, 2.2). Stability selection approach $(flare)$ was applied using TIGER method. Then optimum precision matrix was selected using cross-validation. DSS credible interval fits the future data well on the basis of log-likelihood as compared to the other methods (Figure 2.1).

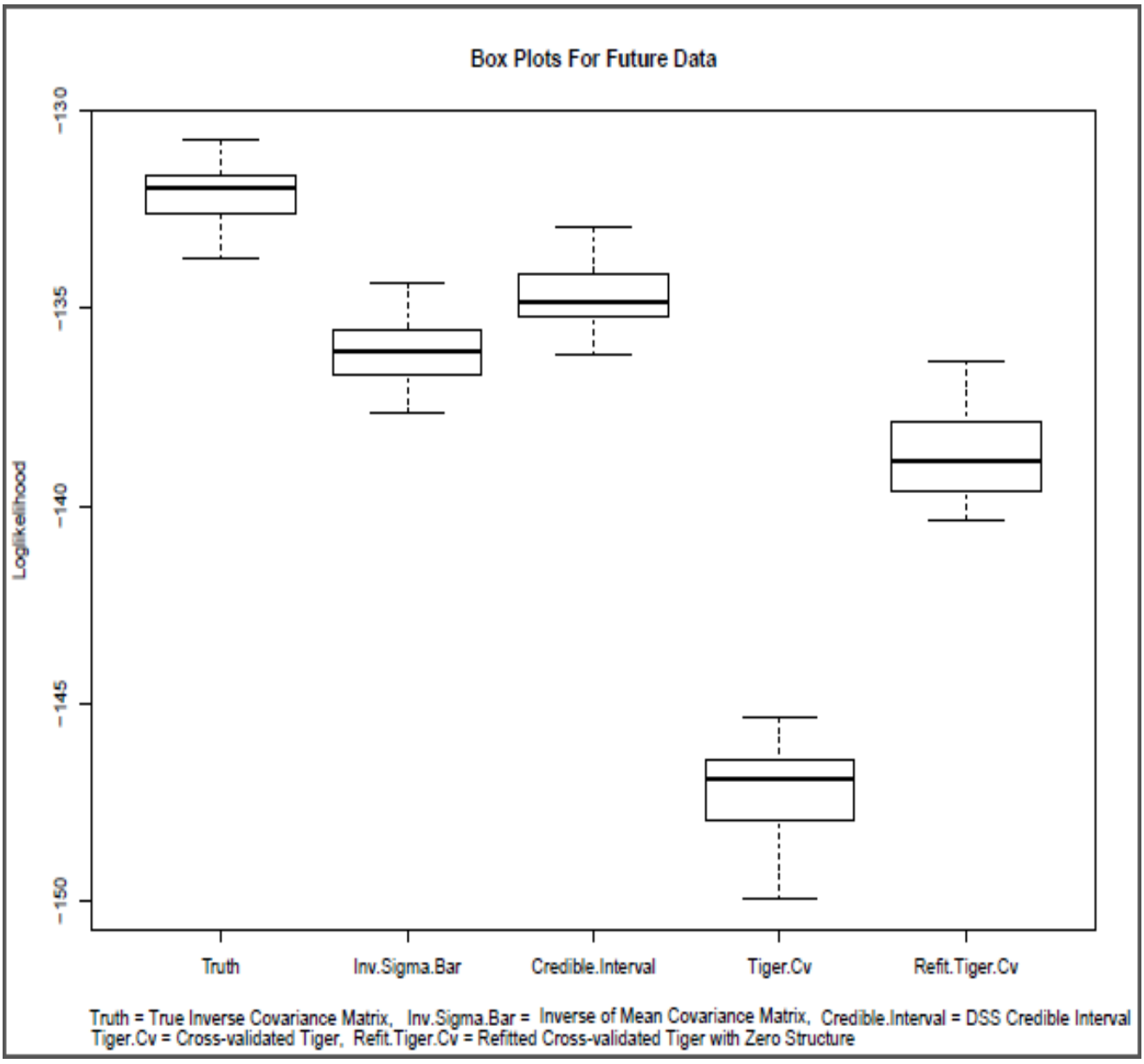


Figure 2.1: Box plots for future data for $AR(2)$ case, $n = 200, p = 100$

For populations with $AR(2)$ covariance structure, our proposed DSS credible interval method performs well on the basis of log-likelihood and fits the future data well as compared to the other methods (Figure 2.1).

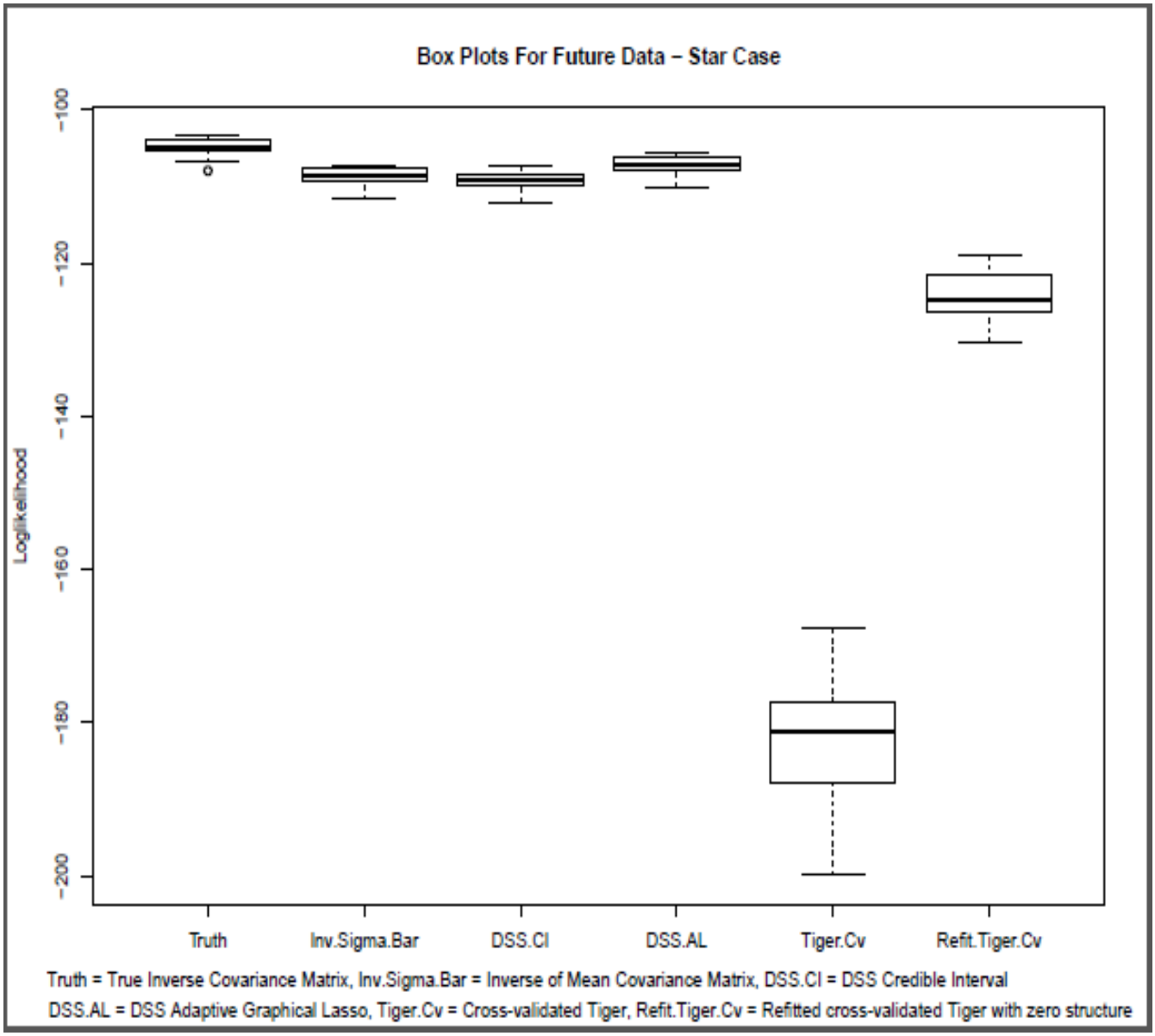


Figure 2.2: Box plots for future data for $Star$ case, $n = 200, p = 100$

For populations with $Star$ covariance structure, our proposed DSS adaptive graphical $Lasso$ method performs well on the basis of log-likelihood and fits the future data well as compared to the other methods (Figure 2.2).Stability selection using CLIME was also used for precision matrix estimation, but it had poor log-likelihood estimates and did not fit the future data well.

2.4.2 Example: Metabolomics Network

The methods used in the previous section were also compared here to estimate the number of connected edges in a precision matrix.

2.4.2.1 Data and Methods

Here, I used a published dataset of frequencies of $p = 174$ metabolites in the fecal sample of 49, 8 year old children born at term. It was the control data in a sense that 8 year old children born at term had no disease. The purpose of the study was to find out the conditionally dependent metabolites. The same Matlab code Wang (2012) that I used for generating posterior samples of covariance matrices in the simulation study was used to obtain the posterior samples of covariance matrices generated from the control data using Bayesian adaptive graphical $Lasso$ method.

Since $\boldsymbol{p > n}$, I calculated the initial inverse covariance matrix for metabolites data using graphical $Lasso$ (Friedman et al, 2008). The inverse covariance matrix obtained using graphical $Lasso$ was then used as an initial value for 10,000 samples simulations on control data with 5,000 burn-ins to obtain the posterior samples of covariance matrices using Bayesian adaptive graphical $Lasso$ method. The 10,000 posterior covariance matrices samples were then thinned to 1,000.

The validity of the simulated covariance matrices using the Bayesian adaptive graphical $Lasso$ was checked by plotting 1,000 simulated λ (tuning parameter) values shown in Figure 2.3, and by plotting the maximum eigenvalues of a sample of 1,000 simulated covariance matrices shown in Figure 2.4, on the basis of smooth plot pattern. Figure 2.3 shows that the simulated tuning parameters are mixing well. However, Figure 2.4 shows that some but not all of the simulated covariance matrices are auto-correlated.

I considered a sample of 1,000 covariance matrices and then calculated the posterior mean from those matrices. Then I used the posterior sample to characterize the range of fit for the posterior mean inverse. The purpose was to understand how the lack of fit

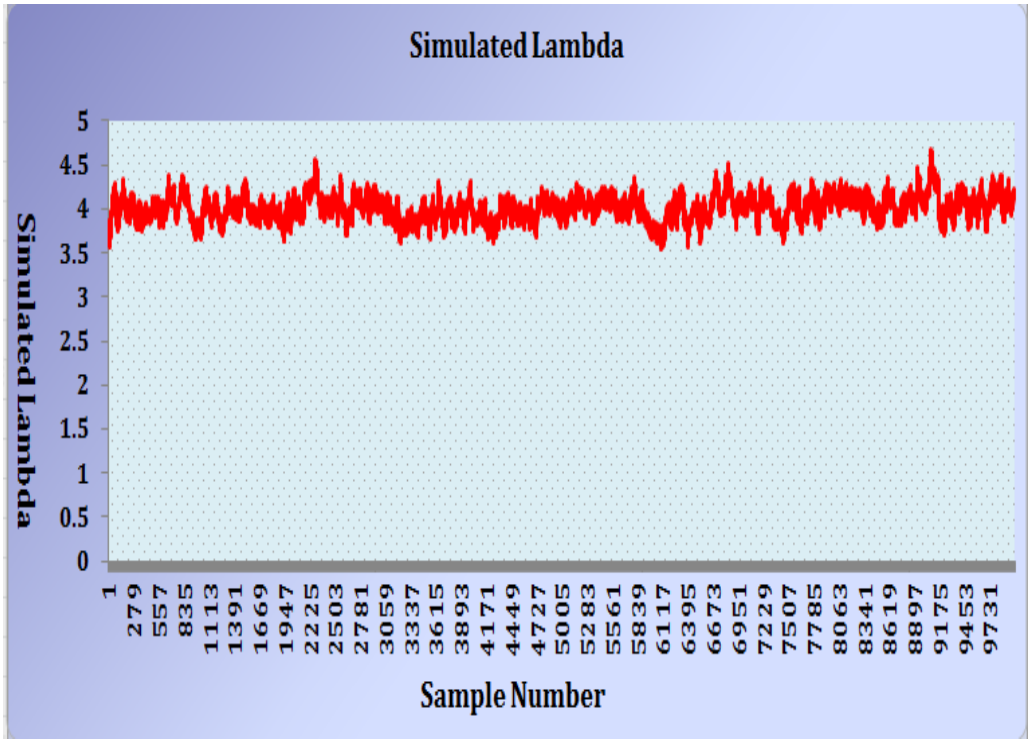


Figure 2.3: 1000 Simulated tuning parameter values using Bayesian adaptive graphical $Lasso$

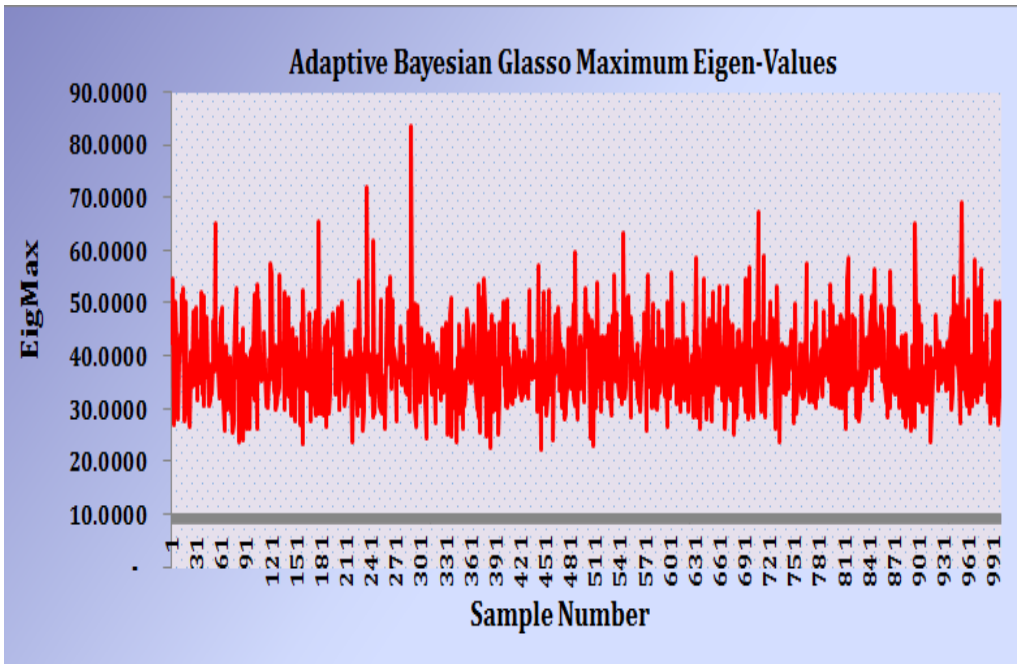


Figure 2.4: Maximum Eigenvalues for 1,000 simulated covariance matrices using Bayesian adaptive graphical $Lasso$

Induced by shrinking the posterior mean compared to my uncertainty about the parameter. A 90% credible interval was used for the range of fit of posterior mean of inverse covariance matrices.

Sparsification methods, such as rho threshold based on thresholding the absolute partial correlations, credible interval based on thresholding $X\%$ credible interval, ratio selection (Wang, 2012), and adaptive graphical $Lasso$, can be used to detect the connected edges in a graph. All these strategies can be tuned up to produce a sparse graph.

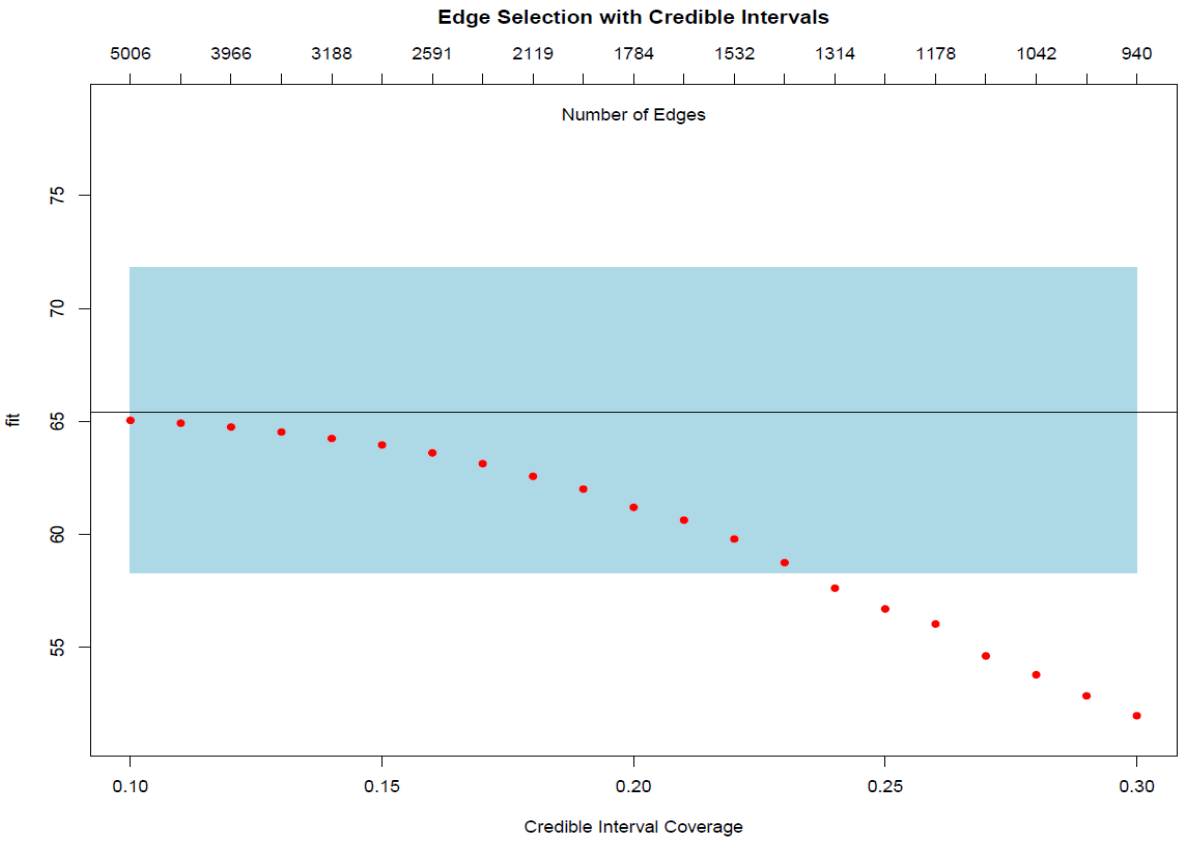


Figure 2.5: Expected fit for sparse summaries of the precision matrix for 174 volatile compounds measured for 49 individuals. The $x\ axis$ shows the coverage of the credible interval used to select zero elements for Γ; the top axis shows the resulting number of edges in the graph. The 90% credible interval for the fit of $\overline{\Sigma}^{-1}$ (blue region), and its expected fit (central line) are shown for comparison.

2.4.2.2 Comparison of Selection Strategies

The expected fit for sparse summaries of the precision matrix for 174 volatile compounds measured for 49 children born at term using the DSS credible interval method, where we

set those elements to zero where $X\%$ credible interval contains zero, is portrayed in Figure 2.5. The connected edges are those edges where $X\%$ credible intervals do not contain zero. I obtained different numbers of connected edges in a precision matrix for different credible intervals ranging from 10% to 99% (I have shown only 10% to 30% credible intervals in Figure 2.5). Each red dot is an estimated inverse covariance matrix for $X\%$ credible interval. The last red dot within the blue region denotes the final estimated precision matrix that I am going to pick. It can be seen that the last red dot inside the blue shaded area is against the 23% credible interval $(x - axis)$, and the corresponding connected edges against the 23% credible interval are 1407 $(y - axis)$. Therefore, my final estimated inverse covariance matrix against a 23% credible interval has 1407 non-zero elements. My approach can also be used for DSS adaptive graphical $Lasso$, DSS ρ threshold and DSS ratio selection approaches.

Table 2.2: Comparison of sparsification strategies for the volatilome data. For each strategy, the criteria corresponding to the sparsest model inside the top 95% of fits is given, as well as the number of edges of that model. By design, the expected fit of each selected model should be approximately the same, and the E(fit) column confirms this. The expected fit of the Bayes estimate is also given for comparison.

Approach	Criteria to Retain	Number of Edges	E(fit)
Bayes estimate ($\bar{\boldsymbol{\Sigma}}^{-1}$)	-	15051	65.4
DSS ratio selection	Ratio > 0.45	2212	59.4
DSS adaptive graphical $Lasso$	$\lambda > 6.8 \times 10^{-7}$	1760	58.4
DSS ρ threshold	$\lvert\rho\rvert > 0.015$	1551	58.7
DSS credible Interval	23% credible interval	1407	58.8

The number of connected edges based on Bayes estimate, DSS ratio selection, DSS adaptive graphical $Lasso$, DSS ρ threshold and DSS credible interval are given in Table (2.2). The inverse of the posterior mean is dense, having all the possible edges (15051). Therefore, the posterior mean inverse matrix is not sparse at all. DSS ratio selection has 2212 connected edges, DSS adaptive graphical $Lasso$ has 1760 connected edges, DSS ρ threshold has 1551 connected edges, and DSS credible interval has 1407 connected edges. All the sparsification strategies produce approximately the same level of fit (Table 2.2). The sparsification strategy that produces optimal fit with the sparsest graph is

desirable. Here, the DSS credible interval method produces the sparest graph (1407 connected edges) for approximately the same level of fit. I can conclude that my proposed DSS credible interval method can be used to produce sparsest graphs for moderate to high-dimensional data. Figure 2.6 is the graphical display of the comparison of sparsification strategies for the volatilome data (Table 2.2). The number of edges retained decreases since the precision matrix becomes sparser as the threshold increases (Figure 2.6).

2.5 Comparison of DSS Methods with other Methods on Metabolomics Data

I used my proposed DSS methods to estimate the number of connected edges in the final inverse covariance matrix for metabolites data. I compared my DSS methods with

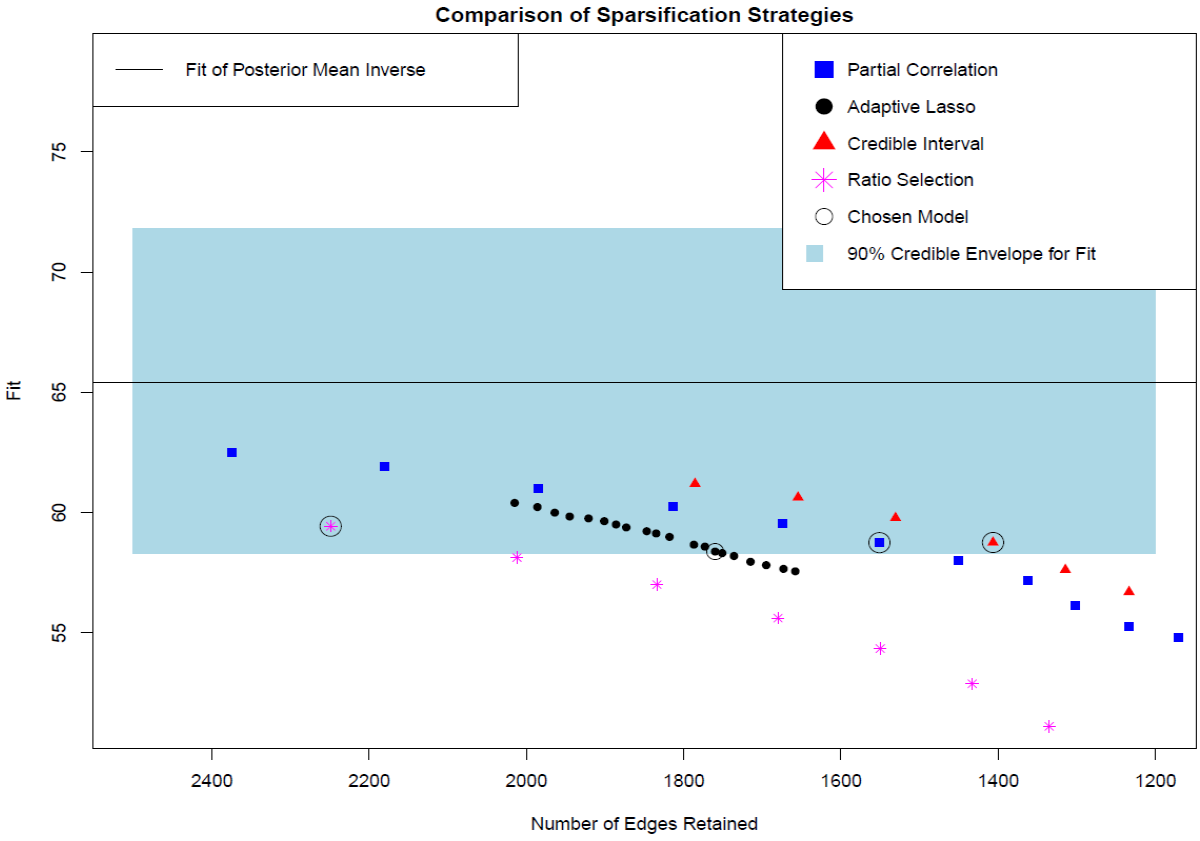


Figure 2.6: Expected fit for sparse summaries of the precision matrix for 174 volatile compounds measured for 49 individuals, using different selection criteria. The 90% credible interval for the fit of $\bar{\Sigma}^{-1}$ (blue region), and its expected fit (central line) are shown for comparison.

CLIME, stability selection and BINCO for the estimation of connected edges in the inverse covariance matrix in the following subsections.

2.5.1 CLIME Using Cross-Validation for Estimation of Connected Edges

I used R-package CLIME (Cai et al, 2011) to produce the inverse covariance estimates for a grid of constrained λ values on metabolites data having dimension $p = 174$ with sample of size $n = 49$. CV.CLIME (Cross-validated CLIME) is a function in R-package CLIME that selects the optimal value of tuning parameter λ based on the log-likelihood loss to produce the optimal graph, and, this is an advantage since we do not need to worry about choosing the tuning parameter. The estimated precision matrix corresponding to the optimal lambda value (λ = 0.1507) was selected to identify the connected edges. There were 680 off-diagonal edges selected, based on the criterion of having an absolute value greater than 10^{-03}. CLIME produced a sparser graph (680 connected edges) as compared to our proposed DSS methods, but it yielded a poorer fit according to the log-likelihood (Figure 2.7).

2.5.2 Stability Selection Using CLIME for Estimation of Connected Edges

$Flare$ (A new family of $Lasso$ regression) R-package (Li et al, 2015) was used for stability selection. The $Flare$ function $sugm$ was used for the estimation of high-dimensional ($p = 174$ metabolites) sparse precision matrices with CLIME method. We obtained different precision matrices estimates for a grid of λ values. The next step was to apply the stability approach in order to find out the optimal graph using cross-validation selection that gives us the cross-validated optimal λ value and its corresponding selected final model.

The resampling was done using cross-validation. The log-likelihood loss was used in cross-validation. The $sugm.select$ (Li et al, 2015) function applied cross-validation to select the optimal graph at the optimal lambda value of λ = 0.218. Those absolute off-diagonal edges were selected which were greater than 10^{-3}. There were 392 connected edges on the basis of this criterion.

Stability selection using CLIME produced a sparser model (392 edges) as compared to CLIME using cross-validation (680 edges). Although, stability selection using CLIME

produced the sparest graph (392 connected edges) as compared to our proposed DSS methods, yet, for the simulated data, it did not fit the future data well based on log-likelihood (Figure 2.7).

2.5.3 BINCO for Estimation of Connected Edges

The stability selection package $Flare$ uses cross-validated CLIME for the final model selection without estimating and controlling the FDR (False discovery rate). Whereas, BINCO uses $SPACE$ (Sparse partial correlation estimation) algorithm (Peng et al, 2009) to estimate the partial correlations using the neighbourhood selection approach (Meinshausen and Buhlmann,2006). BINCO directly controls the $FDRs$ (false discovery rates) for selecting the edges. The resampling for stability selection using cross-validated CLIME is done using cross-validation using the log-likelihood loss function. Whereas, resampling in BINCO is done using bootstrapping. I obtain the connected edges in the final model using BINCO for different $FDRs$.

BINCO requires the generation of selection frequencies for a specified number of resamples, generally 100 resamples. This is done using the $SPACE$ algorithm. There are two ways to apply $SPACE$ algorithm, one using $Lasso$, and the other using the elastic net. BINCO is applied using $Lasso$, and elastic net in $SPACE$. The $SPACE$ algorithm has two penalty terms λ_1 and λ_2. If λ_2 = 0, then $Lasso$ method is applied in $SPACE$ algorithm. If λ_2 is not equal to zero, then elastic net method is applied in $SPACE$ algorithm. Both L_1 and L_2 penalties of $Lasso$ and ridge methods respectively are linearly combined in elastic net. I obtained the inverse covariance matrices using $SPACE$ algorithm using $Lasso$ in $SPACE$ and elastic net in $SPACE$, and then applied the BINCO R-package (Li et al, 2013) to estimate the connected number of edges for different $FDRs$.

BINCO results are shown in Table 2.3. The number of connected edges for BINCO using $Lasso$ in $SPACE$ for FDRs 0.05, 0.10 and 0.20 are 2349, 2610, and 3219 respectively. However, the number of connected edges for BINCO using elastic net in $SPACE$ for the same FDRs are 2784, 3219, and 4002 respectively. Therefore, BINCO using $Lasso$ in $SPACE$ algorithm produces sparser graphs as compared to BINCO using elastic net in $SPACE$ algorithm.

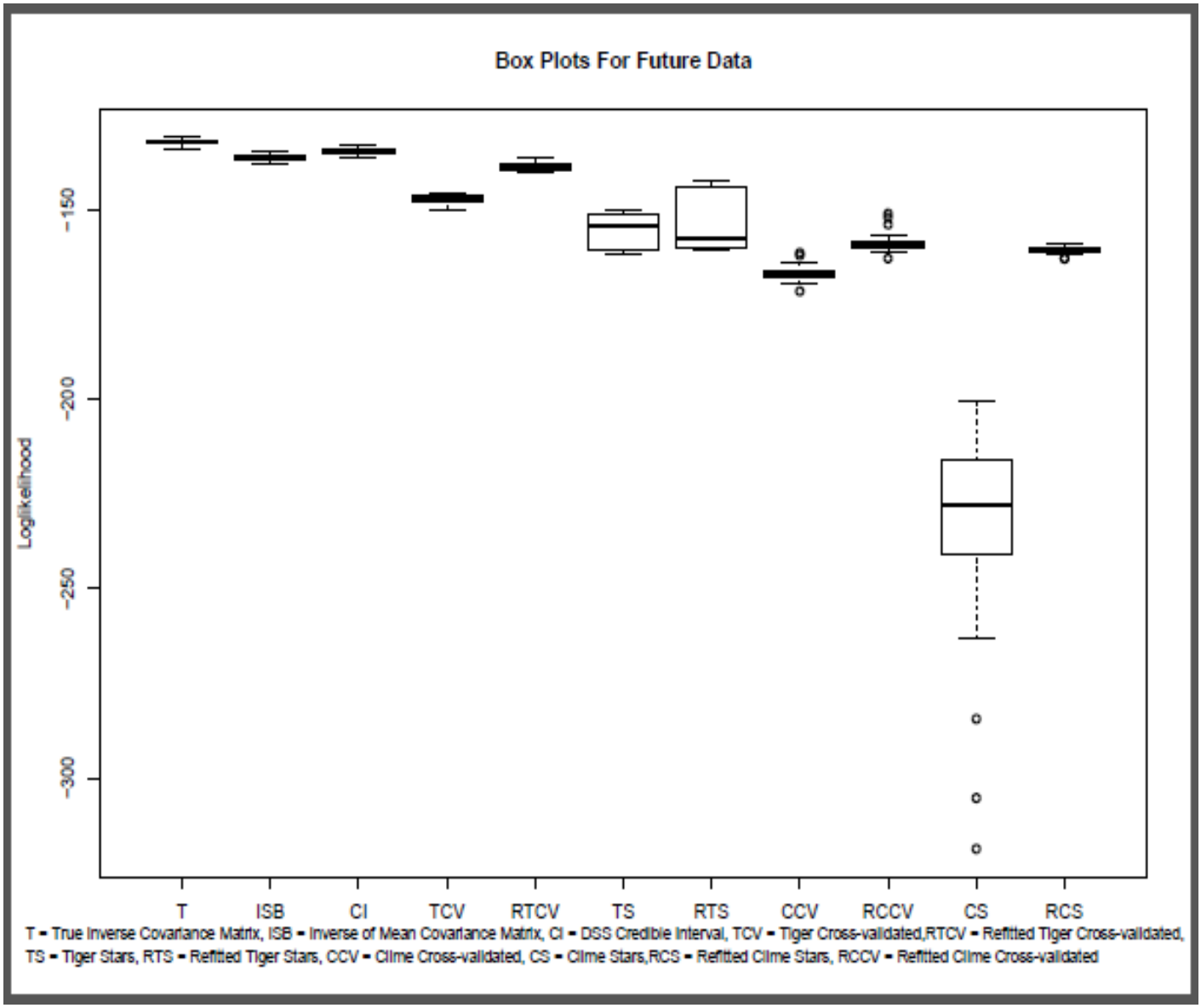


Figure2.7: Box plots for future data for $AR(2)$ case, $n = 200, p = 100$

If we compare the stability selection approach with BINCO for detection of connected edges in the final model, we can clearly see that the stability selection using cross-validated CLIME produces the sparsest model (392 connected edges). Whereas, the connected edges selected by BINCO given in Table 2.3 indicate that BINCO does not produce a sparser model as compared to stability selection approach using cross-validated CLIME. My proposed DSS methods still produce sparser graphs as compared to the graphs produced by the BINCO algorithm.

Table 2.3: BINCO Results

BINCO Using	FDR	Connected Edges
$Lasso$ in $SPACE$	0.05	2349
	0.10	2610
	0.20	3219
Elastic Net in $SPACE$	0.05	2784
	0.10	3219
	0.20	4002

2.6 Conclusions

In this chapter, I proposed flexible DSS-based methods to produce sparse estimates from the posterior distribution over the inverse covariance matrices. The posterior mean serves as a positive definite covariance matrix that is used to satisfy the convergence property of my proposed algorithm. In simulation studies, my proposed DSS-based methods generally produced sparser graphs than the other strategies, such as CLIME, BINCO, and stability selection, by providing a good fit to future data (based on the log-likelihood). For a real metabolites dataset, my DSS-based method for credible intervals produced the sparsest graph as compared to the other DSS-based methods. The DSS credible interval method also fitted the future data well in the simulation study for $AR(2)$ case based on the log-likelihood of future data. Therefore, the DSS credible interval is recommended when sparse inverse covariance matrices are required. DSS based on thresholding the partial correlations performed better based on higher sensitivity and specificity in $AR(2)$ case in the simulation study. However, DSS adaptive graphical $Lasso$ outperformed the other methods in $Star$ case based on highest sensitivity in the simulation study. To summarize the findings, I can conclude that my proposed DSS-based sparsification strategies have a great advantage over other sparsification strategies when producing sparse graphs.

The main advantage of the proposed DSS-based methods is that they produce a sparse, interpretable summary of a complex posterior distribution. I apply various shrinkage techniques to the relevant parameterization of the posterior mean covariance, and the uncertainty represented by the posterior distribution provides the relevant scale to

understand when I have shrunk enough versus too much. The sensitivity and specificity keep increasing for my proposed methods as the dimension and sample size grow. Having low sensitivity for a low-dimensional data for the $Star$ case in the simulation study is a disadvantage of my DSS-based methods. Therefore, I can conclude that my proposed DSS-based methods perform well based on sensitivity and specificity for moderate-dimensional data, and do not perform well for low-dimensional data.

Chapter - 03

Estimating Multiple Graphs with Gaussian Graphical Models

3.1 Introduction

The off-diagonal elements of a precision matrix represent the partial correlations between pairs of variables, given the relationships among the other variables. The partial correlations are calculated by the formula $\rho_{ij} = -\frac{\omega_{ij}}{\sqrt{\omega_{ii}.\omega_{jj}}}$. If we have two or more precision matrices, we may wish to identify which elements are the same or different between the two. Substantial attention has been paid to identify changes in the precision matrices under different conditions (Tian et al, 2016; Zhao et at, 2014; Guo et al, 2011; Cai et al, 2016; Danaher et al, 2013), including the Bayesian approaches (Peterson et al, 2015; Mitra et al, 2016). Danaher et al. (2013) proposed a new joint graphical $Lasso$ (JGL) convex-optimization methodology for the estimation of precision matrices in a high dimensional data setting for multiple classes, which has a faster computation time than the method proposed by (Guo et al, 2011). The joint graphical $Lasso$ $(JGL)$is an extension of the graphical $Lasso$ (Equation 2.1) for multiple classes. The joint graphical $Lasso$ (JGL) proposed by (Danaher et al, 2013) has an arbitrary criteria for the selection of tuning parameters, and it uses ADMM (alternating directions method of multipliers) algorithm. Moreover, for high-dimensional data, JGL does not detect important sparse set of differences between the precision matrices (Table 2, Danaher et al, 2013).

In this chapter, I further extend my DSS-based method proposed for one graph (Chapter 2) to DSS- based method for two graphs to find the sparse sets of differences between the two precision matrices. I modify the ADMM algorithm (Danaher et al, 2013) by replacing the graphical $Lasso$ penalty with the adaptive graphical $Lasso$ penalty to produce the sparse set of differences between the inverse covariance matrices. I also modify the original fused joint graphical $Lasso$ loss function (Danaher et al, 2013) such that my modified loss function picks the tuning parameters automatically. My proposed DSS-based method has an advantage over JGL (Danaher et al, 2013) in that the selection of tuning parameters is not arbitrary, and it can identify a sparse set of differences between the precision matrices even for high-dimensional data. A review of current

methodologies for multiple graphs is given in section 3.2. My proposed DSS-based method to find sparse sets of differences between the two precision matrices is explained in section 3.3.

3.2 Review of Current Methodologies for Multiple Graphs

Comparison between two or more populations is usually done by the researchers using the principal method for selecting the penalty for performance improvement fitting of multiple graphs. This could include searching for similarities (zero patterns of precision matrices elements) and differences among the precision matrices in two or more sets of observations from different populations on the same variables. The estimation of precision matrices for multiple graphs can be done using penalized likelihood approaches or Bayesian approaches given in the subsections 3.2.1 and 3.2.2 respectively.

3.2.1 Penalized Likelihood Approaches

The graphical methods such as the graphical $Lasso$, the adaptive graphical $Lasso$ and CLIME focus on the estimation of a single precision matrix only, but can be extended to find similarities (connected edges) and differences (differential edges) between the two inverse covariance matrices.

The likelihood based method proposed by (Guo et al, 2011) was used for joint estimation of multiple graphical models. They used the following penalized log-likelihood function:

$$maximize_{\{\boldsymbol{\theta}\}} \left\{ \sum_{k=1}^{K} n_k \left(logdet\boldsymbol{\theta}^{(k)} - trace\left(\boldsymbol{S}^{(k)}\boldsymbol{\theta}^{(k)}\right) \right) - P(\{\boldsymbol{\theta}\}) \right\} \qquad (3.1)$$

$$\text{Where,} \qquad P(\{\boldsymbol{\theta}\}) = \lambda \sum_{i \neq j} \sqrt{\sum_k \left|\boldsymbol{\theta}_{ij}^{(k)}\right|} \qquad (3.2)$$

where, $\boldsymbol{S}^{(k)}$ is the sample covariance matrix for the kth class, $\boldsymbol{\theta}^{(k)}$ is the estimated inverse covariance matrix for the kth class, and $P(\{\boldsymbol{\theta}\})$ is the penalty term.

Here, λ is the single penalty term that was selected using BIC (Bayesian Information Criterion). This method encourages only similar patterns (zero patterns i.e. common structure of precision matrices elements) of sparsity across multiple classes. This method has a limitation of ignoring the non-zero edges values and signs. However, for multiple

graphical structures, cooperative $Lasso$ (Chiquet et al, 2011), that is built on Group- $Lasso$ (Yuan and Lin, 2006), enforces the same signs among the non-zero edges.

The method proposed by Guo et al. (2011) has a limitation that it has a single penalty term in Equation (3.2) that is not convex and makes computation time very slow, and therefore, the method is avoided for high-dimensional data. Danaher et al. (2013) proposed a new joint graphical $Lasso$ (JGL) convex-optimization methodology for the estimation of precision matrices in a high dimensional data setting for multiple classes, which has a faster computation time as compared to the computation time of method proposed by Guo et al. (2011). Danaher et al. (2013) proposed the joint graphical $Lasso$ as an extension of graphical $Lasso$ (Equation 2.1) for multiple classes. Danaher et al. (2013) approach was to maximize the penalized log-likelihood function in Equation (3.3). Two variations of the JGL approach may be used—fused graphical $Lasso$ (FGL) and Group graphical $Lasso$ (GGL)—which differ with respect to the term $P(\{\theta\})$, which is the penalized log-likelihood function.

Both FGL and GGL have two penalty terms λ_1 and λ_2.Penalty term λ_1 encourages the sparsity between the two precision matrices by penalizing the off-diagonal elements, and the penalty term λ_2 encourages the similarity between the precision matrices by penalizing the sum of absolute differences between the corresponding elements of each pair of precision matrices.

The expression for P $(\{\Theta\})$ for FGL and GGL, respectively, are given by:

$$P(\{\theta\}) = \lambda_1 \sum_{k=1}^{K} \sum_{i \neq j} \left|\theta_{ij}^{(k)}\right| + \lambda_2 \sum_{k<\acute{k}} \sum_{i,j} \left|\theta_{ij}^{(k)} - \theta_{ij}^{(\acute{k})}\right| \tag{3.3}$$

$$P(\{\boldsymbol{\theta}\}) = \lambda_1 \sum_{k=1}^{K} \sum_{i \neq j} \left|\boldsymbol{\theta}_{ij}^{(k)}\right| + \lambda_2 \sum_{i \neq j} \sqrt{\sum_{k=1}^{K} \boldsymbol{\theta}_{ij}^{(k)^2}} \tag{3.4}$$

Danaher et al. (2013) selected the penalty terms using AIC (Akaike Information Criterion). Danaher et al. (2013) found FGL to be better as compared to GGL, since FGL encouraged similarity (identical inverse covariance matrix elements) between precision matrices, whereas, GGL only encouraged shared patterns of sparsity. An ADMM (Danaher et al, 2013) algorithm was introduced for the estimation of joint graphical $Lasso$. Danaher et al.

(2013) presented a comparison among the graphical $Lasso.$, $FGL, GGL,$ and the method proposed by (Guo et al, 2011) using both simulated and real data. For the simulation study, true positive differential edges were plotted against false positive differential edges and it was seen that FGL performed better as compared to the other methods because of having the highest area under the ROC (receiver operating characteristic) curve (AUC). Sum of the squared errors were plotted against total edges selected and it was seen that $FGL, GGL,$ and the method proposed by (Guo et al, 2011) had smaller sum of squared errors as compared to separate estimation with the graphical $Lasso$. The graphical $Lasso$, which estimated networks separately, had the worst Kullback-Leibler divergence. The run time in seconds was plotted against total non-zero edges selected and it was seen that the graphical $Lasso$ was fastest followed by $FGL,$ then by GGL. Guo et al.'s (2011) method had the longest running time. Both Kullback-Leibler divergence and sensitivity of detection of non-zero edges improved for FGL and GGL as the sample size increased. FGL and GGL are better than the penalty (Equation 3.2) proposed by (Guo et al, 2011) in a way that penalty (Equation 3.2) is not convex that makes the run time very slow for the algorithm proposed by (Guo et al, 2011), whereas, FGL (Equation 3.3) and GGL (Equation 3.4) have convex penalty terms that result in faster computation time.

Guo et al.'s (2011) method uses a penalized-likelihood approach for joint estimation of multiple graphical models that encourages similar patterns of sparsity across multiple classes. In contrast, Cai et al. (2016) proposed $\frac{L_\infty}{L_1}$ weighted constrained minimization method for joint estimation of K sparse matrices (MPE) when the precision matrices were expected to be similar. MPE is an extension of CLIME (Equation 2.4) (Cai et al, 2011) for multiple precision matrices. For K precision matrices, sparsity is encouraged by the $\frac{L_\infty}{L_1}$ objective function. The precision matrices Ω^k for K groups were estimated, where $\Omega^k = (\omega_{ij}^k)$, using the following constrained optimization (Cai et al, 2016):

$$\min_{\Omega_1^{(k)} \epsilon R^{PXP}, 1 \le k \le K} \left(\max_{1 \le k \le K} \left| \Omega_1^{(k)} \right|_1 \right), \tag{3.5}$$

$$subject\ to \quad \max_{i,j} \left\{ \sum_{k=1}^{K} \omega_k \left| \left(\hat{\Sigma}^{(k)} \Omega_1^{(k)} - I \right)_{ij} \right|^2 \right\}^{\frac{1}{2}} \leq \lambda_n$$

where $\omega_k = \frac{n_k}{n}$ is the kth group weight and λ_n is the penalty term. If the solution to Equation (3.5) is denoted by $\widehat{\Omega}_1{}^{(k)} (1 < k \leq K)$, then in general $\widehat{\Omega}_1{}^{(k)}$ are not necessarily symmetrical and the model would be misspecified. The estimator $\widehat{\Omega}^{(k)} = \left(\widehat{\omega}_{ij}{}^{(k)}\right)$ of Ω^k is obtained by symmetrizing $\widehat{\Omega}_1{}^{(k)}$ as follows:

$$\widehat{\omega}_{ij}^{(k)} = \widehat{\omega}_{ji}^{(k)} \coloneqq \widehat{\omega}_{1ij}^{(k)} I\left(\left| \widehat{\omega}_{1ij}^{(k)} \right| \leq \left| \widehat{\omega}_{1ji}^{(k)} \right| \right) + \widehat{\omega}_{1ji}^{(k)} I\left(\left| \widehat{\omega}_{1ij}^{(k)} \right| > \left| \widehat{\omega}_{1ji}^{(k)} \right| \right) \tag{3.6}$$

The tuning parameters selection was done using Bayesian information criterion. The Equation (3.5) is similar to the sum of constrained optimization function for estimation of a single precision matrix with CLIME (Equation 2.4); the main difference is that CLIME does not use the weight factor $\omega_k = \frac{n_k}{n}$. Cai et al.'s (2016) method was proposed for partial homogeneity in the graphical structures of K precision matrices.

Cai et al. (2016) did simulation studies and considered the dimension $p = 200$, three number of groups $K = 3$, sample sizes of each group $n_k = 80,120,150$, for the three groups $k = 1,2,3$ respectively, with 100 replications. The Cai et al.'s (2016) method had the best sensitivity, specificity and Mathews' correlation coefficient , as compared to CLIME for single precision matrix (Cai et al, 2011), the graphical $Lasso$ (Friedman et al, 2008), (Guo et al's, 2011) method, FGL and GGL (Danaher et al, 2013).

3.2.2 Bayesian Approaches

Peterson et al. (2015) proposed a Bayesian approach for the joint inference of multiple Gaussian graphical models. The proposed Bayesian approach was used for inferring multiple undirected networks that may be unrelated or may share some common features (same edges). The undirected graphical models are also referred to as Markov random fields (MRFs). The estimated graphical structures were linked to MRF prior, which encouraged common structures (same edges) in the related graphs. If we have $1, \ldots, K$ graphs, then the binary vector $g_{ij} = \left(g_{1,ij}, g_{2,ij}, \ldots, g_{K,ij}\right)^T$ represents the inclusion of edge

(i,j) in these graphs. If the probability of an edge inclusion for binary vectors g_{ij} is represented by $P(g_{ij}|v_{ij},\Theta)$, then the graphs $(G_1,G_2,\ldots,G_K)$ have a product of each edge densities as a joint prior as follows:

$$p(G_1,\ldots,G_K|v,\Theta)=\prod_{i<j}p(g_{ij}|v_{ij},\Theta) \qquad (3.7)$$

$$Where, \qquad v=\{(v_{ij}|1\le i<j\le P)\}$$

Here v_{ij} is an edge-specific parameter related to g_{ij} (each set of edges), Θ is a symmetric matrix having K rows and K columns that represent the graphs pairwise relatedness (same edges) for each sample group. We set the diagonal entries of Θ to zero, and the network relatedness is represented by the non-zero entries in the off-diagonal elements of Θ. The edges prior probabilities of inclusion are influenced by the parameters v and Θ.

Peterson et al. (2015) performed the simulation studies for the assessment of parameter inference, and to compare the performance of the proposed method (Peterson et al, 2015) with fused graphical $Lasso$, and group graphical $Lasso$ (Danaher et al, 2013) using tuning parameters chosen with AIC, and separate estimation of the two precision matrices with G-Wishart prior:

$$p(\boldsymbol{\Omega}|G,b,D)=I_G(b,D)^{-1}|\boldsymbol{\Omega}|^{\frac{b-2}{2}}exp\left\{\frac{-1}{2}\,tr\,(\Omega D)\right\}, \quad \Omega\in P_G \qquad (3.8)$$

where, the parameter $b>2$ represents degrees of freedom, the positive definite symmetric matrix D has a dimension of $p\ x\ p$, the normalizing constant is represented by I_G , and the set of all positive definite symmetric matrices having a dimension $p\ x\ p$ and ω_{ij} = 0 are represented by P_G.

In simulation studies, the dimension was set to $p=20$. The precision matrices obtained were made positive-definite by using the sum of off-diagonal elements of each row of the precision matrix as a divisor for each off-diagonal element of the precision matrix, and then taking average of the precision matrix with its transpose (Danaher et al, 2013). For the detection of differential edges, the true positive rate (TPR), and area under an ROC curve (AUC) were maximum both for $n=50$ and $n=100$ for Peterson's method (Peterson et al, 2015) as compared to FGL and GGL (Danaher et al, 2013). For graph structure

learning, FGL and GGL (Danaher et al, 2013) had the highest TPR (true positive rate) and FPR (false positive rate) both for $n = 50$ and $n = 100$, and, the proposed method (Peterson et al, 2015) had the highest AUC.

The similarities among the groups were judged on the basis of PPI (posterior probabilities of inclusion) of the parameters. Those set of edges were selected which had $PPI > 0.50$. An MCMC sampler was constructed for updating graph, precision matrix, parameters of network relatedness and edge-specific parameters. The small values of standard errors of 25 simulated datasets showed the stability of the results for the proposed method (Peterson et al, 2015) when the sample sizes of the datasets were moderate $(n = 100)$ and the dimension p was equal to 20.

Peterson et al. (2015) proposed a Bayesian approach for the joint inference of multiple Gaussian graphical models for several related graphs using MRF over graphs. However, (Mitra et al, 2016) proposed a Hierarchical Bayesian graphical model to address heterogeneity (differences) and joint inference on dependence structure across two related subgroups. Let G_1 and G_2 be the two graphs, y be the observed data, π be the hyper-parameter which is the joint probability of common edges between G_1 and G_2 and Θ and β be the parameters for y. Let the prior graph is denoted by G_0. Then the uniform prior for G_1 is, $G_1 \sim U(G_0)$. The graph G_1 comprises of each edge of the graph G_0 independently selected with probability 0.5. The difference between two graphs is denoted by $\delta_{ij,}$ and $\delta_{ij} \sim Ber\,(\pi), i < j$. The hyper-prior $\sim Beta\,(a, b)$ **,** where a and b are the parameters of Beta distribution chosen arbitrarily. The non-zero elements have independent priors which are as follows:

$$\beta_{ij}^{k} \sim N\left(0, \sigma_{\beta}^{2}\right),\ i, j \epsilon E^{k}\ , k = 1,2 \qquad (3.9)$$

The process for comparing graphs depended on a threshold for the posterior probability of an edge (or lack of an edge) being the same in the two graphs. However, Mitra et al. (2016) gave the relationship between the latent binary indicators v_{kti}, where v_{kti} are interpreted as protein activation or histone modification presence, respectively. The data model is as follows:

$$p(y_{kti}|\nu_{kti}, \theta^k) \propto \begin{cases} N(\mu_{1ik}, \sigma^2_{1ik}) & if\ \nu_{kti} = 0, \\ N(\mu_{2ik}, \sigma^2_{2ik}) & if\ \nu_{kti} = 1 \end{cases} \quad (3.10)$$

where, θ^k represent the parameters used for indexing the sampling model.

Simulation studies were conducted by Mitra et al. (2016) to compare the performance of their proposed differential graph method with the methods proposed (Danaher et al, 2013; Guo et al, 2011) , based on AUC. The proposed differential graph method by Mitra et al. (2016) outperformed methods designed for multivariate normal data (Danaher et al, 2013; Guo et al, 2011) by having the highest area under the ROC curve (AUC). Mitra et al's. (2016) method was not compared to Peterson et al's (2015) method. The limitations of the method proposed by Mitra et al. (2016) are that it can only compare two graphs that are low or moderate in size, and it is computationally intensive.

3.3 Extension of DSS-based Method to Two Graphs to find Important Ω Differences Across Conditions in Precision Matrices

My main emphasis is to extend my proposed DSS-based method for a single precision matrix to finding a sparse set of differences between two precision matrices. The process involves first generating independent posterior samples of precision matrices for each group, and then combining the posterior samples to form a joint posterior distribution. Then, some elements are made identical subject to modification in the combined posterior mean inverse covariance matrix $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{c}}^{-\mathbf{1}}$.

The modification in $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{c}}^{-\mathbf{1}}$ is done using a variation of the FGL – fused Joint Graphical $Lasso$ (Danaher et al, 2013). The penalized likelihood form of this algorithm is as follows:

$$\max_{\Gamma_c} \left[\sum_{c=1}^{C} n_c \left(logdet\ \Gamma_c - tr(S_c\Gamma_c)\right) + \lambda_1 \sum_{c=1}^{C} \sum_{i \neq j} |\gamma_{cij}| + \lambda_2 \sum_{c<c'} \sum_{i,j} |\gamma_{cij} - \gamma_{c'ij}|\right] \quad (3.11)$$

The sparsity of the elements is governed by λ_1 , and the similarity is governed by λ_2 . To produce exact equality between the off-diagonal elements of C precision matrices, the L_1 penalties could be used, but very strong penalization is required for producing exact equality between the off-diagonal elements of C precision matrices. The matrices that already had identical patterns of non-zero elements at many positions, and penalty

parameters that were moderate, the number of identical elements were reduced by optimization. Danaher et al. (2013) suggested that the selection of penalty parameter be guided by "practical considerations" since overly dense models are produced by the conventional criterion for penalty parameters selection such as AIC, BIC and cross-validation.

3.3.1 Proposed Method

I used the posterior covariance mean $\bar{\boldsymbol{\Sigma}}$ in my proposed DSS algorithm for one population to set to zero some of the off-diagonal elements of a precision matrix. Further, I extended my proposed DSS approach for one population to two populations by using the combined posterior mean covariance matrix $\bar{\boldsymbol{\Sigma}}_{\boldsymbol{c}}$ instead of $\boldsymbol{S}$ in Equation (3.11) to detect sparse sets of differences between two precision matrices. I also changed the penalization strategy.

To obtain sparse matrices, an adaptive graphical $Lasso$ penalty of the following form may be used:

$$\Gamma_{\lambda} = \max_{\Gamma} \left[logdet\,(\Gamma) - tr(\Gamma S) - \lambda \sum_{i \neq j} \frac{|\gamma_{ij}|}{\sqrt{\gamma^{*}_{ij}}} \right] \qquad (3.12)$$

Suppose that a set $\boldsymbol{\mathcal{H}}$ holds the selected elements (identical elements across conditions), then the objective function may be modified:

$$\max_{\Gamma c} \left[\sum_{c=1}^{C} n_c \left(logdet\,\Gamma_c - tr(\bar{\Sigma}_c \Gamma_c) \right) + \lambda_1 \sum_{c=1}^{C} \sum_{i \neq j} \frac{|\gamma_{cij}|}{\sqrt{\gamma^{*}cij}} + \lambda_2 \sum_{c<c'} \sum_{i,j \in \mathcal{H}} \left| \gamma_{cij-} \gamma_{c'ij} \right| \right] \quad (3.13)$$

In order to penalize the differences in elements across conditions, I choose to penalize only those elements that are identical across conditions. The sparsity of the elements is governed by λ_1 **,** and the similarity is governed by λ_2.

The term $\left| \gamma_{cij-} \gamma_{c'ij} \right|$ is forced to 0 for $(i,j) \epsilon\, \mathcal{H}$ by taking large values of λ_2. Optimization is done using the ADMM algorithm. The elements of $\boldsymbol{\mathcal{H}}$ are selected using the posteriors of $\boldsymbol{\Omega_c}$; specifically, if zero is included in a $P\%$ credible interval for $\gamma_{cij-} \gamma_{c'ij}$, $(i,j)_{c,c'}$ is included in $\mathcal{H}_p$ (set of zero elements based on $P\%$ credible interval). Progressively sparse

sets of differences are generated by increasing the credible interval P. The tuning parameter n_c is set to be proportional to each group's sample size.

3.4 Simulation Study

A simulation study was done to evaluate the performance of my proposed method by detecting the differences in pairs between the inverse covariance matrices. 50 replicates were considered for the two scenarios in the simulation study. The two scenarios were the sparse case and the dense case.

For the simulations, the sets of differences between the true inverse covariance matrices were sparse. In both sparse and dense cases, true inverse covariance matrices were modified by making 50 random changes along the off-diagonal elements. Thus, both sparse and dense cases had a true inverse covariance matrix and a modified true inverse covariance matrix. I obtained 10,000 samples of posterior covariance matrices using Bayesian adaptive graphical $Lasso$ (Wang, 2012), thinned to 1,000, with 5,000 burn-ins both for sparse and dense cases. I considered $|\gamma_{cij-}\gamma_{c'ij}|$ = 0.1 in order to select edges which differ between the pair in the simulation study.

3.4.1 Sparse Case

In the sparse case, one of the two matrices is an $AR\ (2)$ inverse covariance matrix with $\omega_{ii} = 1$, $\omega_{i,i-1} = \omega_{i-1,1} = 0.5$, and $\omega_{i,i-2} = \omega_{i-2,1} = 0.25$. Just to give an idea of the structure of true inverse covariance matrix (Ω), I present it for $p = 10$ as follows:

$$\begin{pmatrix} 1 & 0.5 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 1 & 0.5 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.25 & 0.5 & 1 & 0.5 & 0.25 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0.5 & 1 & 0.5 & 0.25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.25 & 0.5 & 1 & 0.5 & 0.25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.25 & 0.5 & 1 & 0.5 & 0.25 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.25 & 0.5 & 1 & 0.5 & 0.25 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.25 & 0.5 & 1 & 0.5 & 0.25 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0.5 & 1 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0.5 & 1 \end{pmatrix}$$

The other matrix is the inverse covariance matrix differing at 50 random off-diagonal elements of the $AR\ (2)$ inverse covariance matrix.

3.4.2 Dense Case

In the dense case, one of the two matrices is an inverse covariance matrix having ones as the diagonal element and 0.05 as all the off-diagonal elements. Just to give an idea of the structure of true inverse covariance matrix (Ω), I present it for $p = 10$ as follows:

$$\begin{pmatrix} 1 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 \\ 0.05 & 1 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 \\ 0.05 & 0.05 & 1 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.05 & 1 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.05 & 0.05 & 1 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 1 & 0.05 & 0.05 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 1 & 0.05 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 1 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 1 & 0.05 \\ 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 1 \end{pmatrix}$$

The other matrix is the inverse covariance matrix differing at 50 random off-diagonal elements of the dense inverse covariance matrix.

I ran simulations with $n = 200$, and $p = 100$ for both sparse and dense cases. Previous work has shown that the sample size $n = 200$ was challenging for detecting the differential edges using JGL (Table 2, Danaher et al, 2013). I varied λ_1 and λ_2 for the sparse case. For the dense case, I set λ_1 equal to zero and varied the value of λ_2.

Results from both sparse and dense cases were compared with the results from JGL. JGL was fine-tuned to produce the same number of differences between the precision matrices as were obtained for sparse and dense cases for fair comparison.

3.4.3 Simulation Results

The individual inverse covariance matrices pairs obtained through JGL and my posterior summary had a difference of not more than two, producing an identical number of average differences (Table 3.2). Improved sensitivity and MCC as compared to the sensitivity and

MCC using JGL were obtained by our posterior summary method for the sparse case, throughout the 50 replicates (Table 3.1). There was an improvement of 16% in the sensitivity with a standard deviation of 6% (Table 3.3) using our proposed posterior summary method. The specificity of my proposed posterior summary method was almost the same as the specificity using JGL (Table 3.1). The dense case had a comparable sensitivity to JGL with my posterior summary method (Table 3.1). The MCC was better for the posterior summary methods as compared to MCC for JGL both for sparse and dense cases (Table 3.1).

Table 3.1: Sensitivity, specificity and MCC for my procedure for detecting precision matrix differences, averaged over 50 replicates of simulated data with $p = 100, n = 200$. In each case there are 50 off-diagonal differences between the matrices being compared, with magnitude 0.1. The matrix before changes is either an $AR(2)$ structure (sparse case), or one with diagonal 1 and all off-diagonals 0.05. I compare the JGL inferred differences, with λ selected to match the number of edges detected.

	Sparse Case		**Dense Case**	
	Posterior Summary	JGL	**Posterior Summary**	JGL
Sensitivity	58.6	42.6	8.9	8.1
Specificity	97.9	97.8	98.9	98.9
MCC	35.5	25.4	7.1	6.4

Figures 3.1 portrays the differential edges between the inverse covariance matrices for sparse case at $X\%$ (where $X < 100\%$) credible interval.

The central line represents the fit of the differences between the posterior mean inverse covariance matrices, gray area represents central 90% credible interval for the range of fit of posterior mean inverse covariance differences, and the red dots represent the differential edges matrices between the two inverse covariance matrices at $X\%$ credible interval. The last red dot above the gray area denotes the final estimated matrix of precision matrix differences that I am going to pick. The last dot above the gray area is a matrix of precision matrices differences having 170 differential edges (Figure 3.1).Figure 3.2 portrays the differential edges between the inverse covariance matrices for dense

case at $X\%$ credible interval. The last red dot above the gray area is a matrix of precision matrices differences having 80 differential edges.

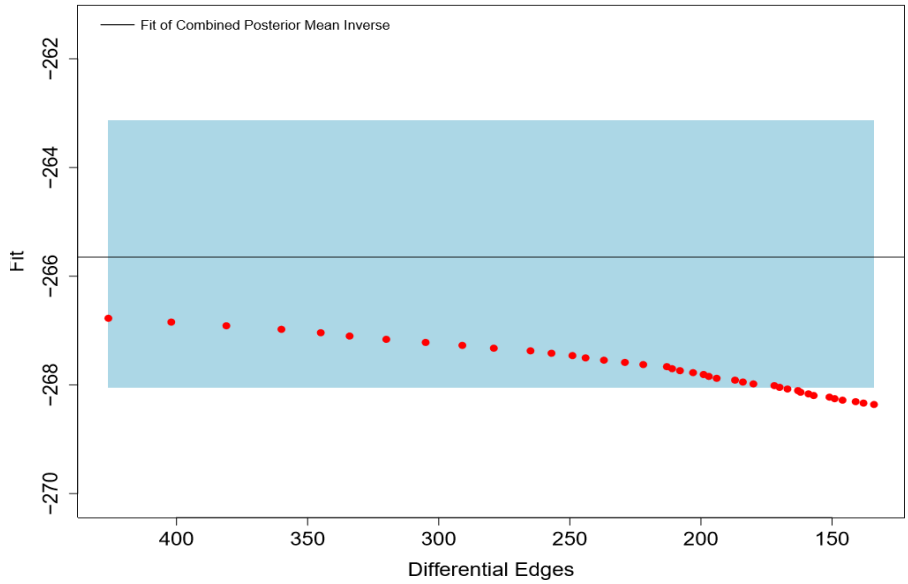


Figure 3.1: Graph with edges corresponding to my inferred differential precision matrices elements for sparse case simulation study considering $n = 200, p = 100$. $Y - axis$ represents the range of fit of posterior means differences.

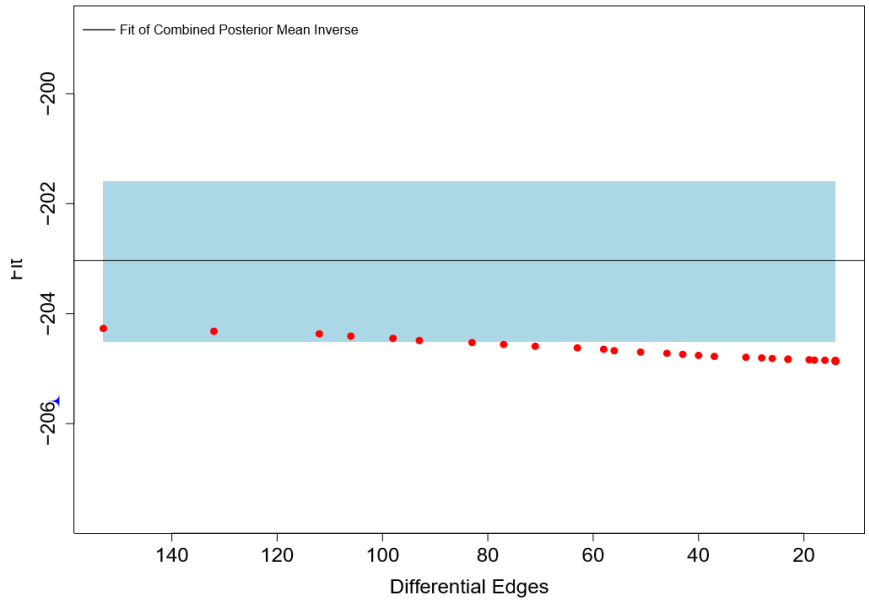


Figure 3.2: Graph with edges corresponding to my inferred differential precision matrices elements for dense case simulation study considering $n = 200, p = 100$

Table 3.2: Comparison between JGL differential edges and my posterior summary approach differential edges for sparse case over 50 replicates of simulated data with $p = 100, n = 200$

Replication	JGL Differential Edges	Posterior Summary Differential Edges	Difference	Replication	JGL Differential Edges	Posterior Summary Differential Edges	Difference
1	170	170	0	**26**	161	161	0
2	107	106	1	**27**	130	128	2
3	185	186	-1	**28**	130	131	-1
4	86	84	2	**29**	97	96	1
5	131	130	1	**30**	123	125	-2
6	114	113	1	**31**	127	128	-1
7	109	110	-1	**32**	139	137	2
8	145	145	0	**33**	127	126	1
9	130	131	-1	**34**	126	127	-1
10	151	152	-1	**35**	110	111	-1
11	149	150	-1	**36**	75	75	0
12	138	138	0	**37**	112	111	1
13	127	126	1	**38**	147	148	-1
14	148	148	0	**39**	167	166	1
15	131	131	0	**40**	109	110	-1
16	103	103	0	**41**	111	112	-1
17	161	160	1	**42**	124	125	-1
18	139	137	2	**43**	115	116	-1
19	111	110	1	**44**	157	156	1
20	121	119	2	**45**	152	151	1
21	137	136	1	**46**	125	125	0
22	112	112	0	**47**	159	160	-1
23	142	141	1	**48**	105	106	-1
24	161	162	-1	**49**	129	130	-1
25	131	131	0	**50**	116	117	-1
					Average of Differential Edges		**0.06 ≅ 0**

Table 3.3: Comparison between JGL inferred differences in sensitivity and my posterior summary approach for sparse case over 50 replicates of simulated data with $p = 100, n = 200$

Replication	Sensitivity Posterior Summary	Sensitivity JGL	Difference	Replication	Sensitivity Posterior Summary	Sensitivity JGL	Difference
1	62%	46%	16%	**26**	62%	36%	26%
2	64%	42%	22%	**27**	58%	46%	12%
3	74%	48%	26%	**28**	52%	32%	20%
4	38%	20%	18%	**29**	56%	36%	20%
5	52%	46%	6%	**30**	56%	36%	20%
6	52%	34%	18%	**31**	62%	46%	16%
7	44%	30%	14%	**32**	64%	56%	8%
8	56%	50%	6%	**33**	62%	42%	20%
9	56%	44%	12%	**34**	54%	36%	18%
10	60%	54%	6%	**35**	50%	40%	10%
11	64%	56%	8%	**36**	48%	38%	10%
12	68%	42%	26%	**37**	50%	36%	14%
13	64%	40%	24%	**38**	76%	62%	14%
14	60%	46%	14%	**39**	62%	32%	30%
15	64%	48%	16%	**40**	60%	42%	18%
16	52%	28%	24%	**41**	56%	48%	8%
17	72%	44%	28%	**42**	74%	52%	22%
18	60%	54%	6%	**43**	58%	46%	12%
19	58%	50%	8%	**44**	62%	44%	18%
20	64%	48%	16%	**45**	60%	46%	14%
21	58%	38%	20%	**46**	52%	40%	12%
22	52%	36%	16%	**47**	66%	48%	18%
23	54%	42%	12%	**48**	54%	32%	22%
24	52%	38%	14%	**49**	58%	46%	12%
25	58%	48%	10%	**50**	58%	42%	16%
					Average of Sensitivity Differences		**16%**
					Standard Deviation of Sensitivity Differences		**6%**

3.5 Metabolomics Example – Important Ω Differences Between Case and Control

Here, I used a published dataset comprising of the dimension $p = 174$ metabolites in the fecal sample of 49 eight year old children born at term (a child born between 37 and 42 weeks of pregnancy), and 42, eight year old children born preterm (a child born before 37 weeks of pregnancy). The 49 eight year old children's data was considered as controls since the children were not born premature, and the 42, eight year old children's data was considered as cases since the children were born premature. I compared the differences (differential edges) between the precision matrices of 49 controls with the precision matrices of 42 cases.

My inferences were based on posterior distributions generated independently from the Bayesian adaptive graphical $Lasso$ (Wang, 2012). The cases and controls precision matrices differential edges graph is shown in Figure 3.3, representing 499 differences. Most of the vertices have differences of 1 to 10 of their incident edges (sharing common vertices). There are 14 vertices shaded in black, which have differences in excess of 10 of their incident edges. Two vertices are not involved in any altered edges.

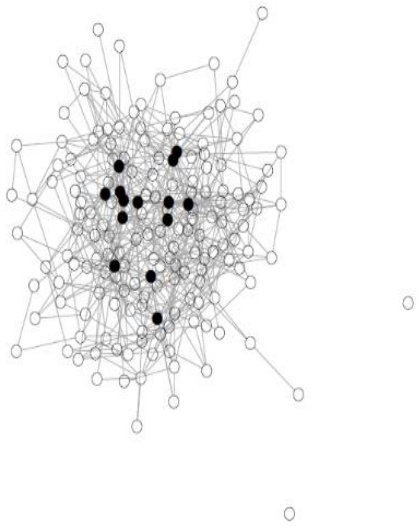


Figure 3.3: Graph with edges corresponding to my inferred differential precision matrix elements for 174 volatile compounds measured for 42 cases and 49 controls. Vertices with more than 10 altered matrix elements are shown in black.

3.6 Conclusions

I demonstrate that my extended DSS-based approach is recommended to select sparse sets of differences in the inverse covariance matrices across conditions. Inference about which elements are similar in a set of the inverse covariance matrices is not addressed by any existing Bayesian methods. Consequently, when we have individual dense matrices, my proposed method is adequate to infer sparse sets of differences between the inverse covariance matrices, though for moderate sample sizes, my proposed method appeared to be a challenging problem in my simulation study (Table 3.1). Clear criterion to select sparse sets of differences between the inverse covariance matrices is not provided by the frequentist approaches, though increasingly sparse sets of differences are produced by the frequentist approaches. A further advantage of my proposed DSS-based method for one graph is that it is easily extendable to the multiple graphs case without requiring any separate implementation. If the shared elements in the precision matrices are modelled in detail (Peterson et al, 2015), even then my proposed method will be very worthwhile preliminary to modelling choices investigation.

Chapter - 04

Decoupled Shrinkage and Selection Sparse Factor Models

4.1 Introduction

In Chapter 2, I proposed decoupled shrinkage and selection methodology to sparsify some elements of a precision matrix to exact zeros in a Gaussian graphical models setting. In Chapter 3, I extended the decoupled shrinkage and selection approach to two populations to detect sparse sets of differences between the precision matrices. In this chapter, I apply similar decoupled shrinkage and selection methodology to sparsify factor analysis models.

Factor analysis is a statistical technique used to analyse the covariance/correlation structure among the observed quantitative variables in terms of unobserved variables called factors. The factor loading matrix $\boldsymbol{\Lambda}$ represents the relationships between the factors and the variables. Sparsification of $\boldsymbol{\Lambda}$ may be desirable for simplifying the interpretation of the results of factor analysis by setting some parameters to zero. A factor loading of zero indicates no relationship between a variable and a factor. One of the drawbacks of standard factor analysis is that it does not shrink any elements of the factor loading matrix to exactly zero, so sparse factor models may be used when model interpretation is important.

Various Gaussian penalized factor analysis models have been proposed to induce sparsity in the factor loading matrix using penalized maximum likelihood estimation (Choi et al, 2010; Hirose and Yamamoto 2014, 2015; Ning and Georgiou, 2011). Bayesian sparse factor models were proposed by West (2003) and Carvalho et. al, (2008). Hui et al. (2018) applied $\boldsymbol{OFAL}$ – Ordinary factor analysis $Lasso$ – using non-convex penalties on normal as well as non-normal (negative Binomial) responses without pre-specifying the number of factors k, and Kim et. al (2015) applied $\boldsymbol{SFA}$ – sparse factor analysis methodology on non-normal (combination of binary and count) data. Bayesian approaches for non-normal data i.e. multivariate probit data have also been proposed (Chib and Greenberg, 1998; Hahn, Carvallho and Scott, 2012).

Let $\boldsymbol{p}$ be the data dimension, and $\boldsymbol{k}$ be the number of factors. A Gaussian factor model is expressed as follows:

$$y_i = \boldsymbol{\Lambda}\eta_i + \epsilon_i, \tag{4.1}$$

where, the observed variables vector is represented by y_i with dimension $p \times 1$, the factor loading matrix of dimension $p \times k$ is represented by $\boldsymbol{\Lambda}(k < p)$**,** factor scores or latent variables vector of dimension $k \times 1$ is represented by $\eta_i \sim N(\mathbf{0}, \boldsymbol{I})$, and the idiosyncratic noise is represented by $\epsilon_i \sim N(\mathbf{0}, \boldsymbol{\Psi})$ with $\boldsymbol{\Psi} = diag(\sigma_1^2, \ldots, \sigma_p^2)$.

The covariance matrix Σ can be estimated as follows:

$$\Sigma = \Lambda\Lambda^T + \Psi \tag{4.2}$$

Some Bayesian factor models have been proposed which generate the posterior samples of factor loading matrices (Hahn, Carvalho and Scott, 2012; Murray et. al. 2013; Bhattacharya and Dunson 2011). Hahn, Carvalho and Scott (2012) proposed a Bayesian probit factor model to sparsify the posterior samples of the factor loading matrices, Murray et. al (2013) worked with the dense matrix, and Bhattacharya and Dunson (2011) generated a sparse posterior. A penalization method called $\boldsymbol{fanc}$ (factor analysis using non-convex penalties) was proposed by Hirose and Yamamoto (2015) to shrink the factor loading matrix with shrinkage parameter selection by using different selection criterion such as AIC, BIC, $CAIC$, and $EBIC$.

In chapter 2, I proposed a decoupled shrinkage and selection (DSS) method for one graph in a GGM setting that was based on generating posterior samples of inverse covariance matrices and then shrink some of the off-diagonal elements of the inverse covariance matrices to exact zeros. Here, I apply the DSS method in the factor analysis setting to produce sparse factor models. My proposed DSS sparse factor model includes generating posterior samples of factor loading matrices, and then shrinking the factor loading matrices by selecting tuning parameters. The difference between my proposed DSS method for one graph and the DSS sparse factor model is that I am sparsifying the factor loading matrices instead of sparsifying the inverse covariance matrices. My proposed DSS sparse factor model picks the final factor loading matrix that is within the 90% credible interval of range of fit of posterior mean inverse against the sparsity-inducing parameters (ρ, γ). The main difference with $\boldsymbol{fanc}$ is that I operate on a posterior covariance mean instead of a sample covariance matrix

when I use $\boldsymbol{fanc}$ to sparsify the factor loading matrix. In addition, the final sparsity-inducing parameters against which the final factors loading matrix is selected using my proposed DSS factor analysis model is different from the sparsity-inducing parameters selected using $\boldsymbol{fanc}$. I show that the DSS sparse factor model has a higher true negative rate and picking correct number of factors as compared to the $\boldsymbol{fanc}$-based methods in the simulation studies.

The remainder of this chapter is structured as follows. Section 4.2 details existing sparse factor models; section 4.3 details my proposed decoupled shrinkage and selection sparse factor model; section 4.4 details the simulation studies for continuous and discrete cases; section 4.5 details two applications of my proposed DSS sparse factor model on real life data both for continuous (multivariate normal) and discrete (multivariate binary) cases; and section 4.6 presents conclusions and recommendations.

4.2 Sparse Factor Models

4.2.1 Bayesian Approaches (Gaussian Case)

Bayesian specifications (Arminger & Muth´en, 1998; Song & Lee,2001) of factor analysis models (Equation 4.1) have used inverse-gamma prior distributions on the error variance, and normal prior distributions on the diagonal and off-diagonal elements of the loading matrix. These choices led to a Gibbs sampler for posterior computation. However, when there were highly correlated variables, the Gibbs sampler behaved very poorly. More efficient Gibbs samplers have been proposed to address this issue (Ghosh & Dunson, 2009; Liu & Wu, 1999; Gelman, 2006).

A Bayesian sparse factor model that specifies the normal priors on the elements Λ_{ij} of the factor loading matrix $\boldsymbol{\Lambda}$, to induce zeros in the factor loading matrix with high probability, considering many explanatory variables was proposed by West (2003). The proposed Bayesian specification of a sparse factor model is as follows:

$$\pi_j \delta_0(\Lambda_{ij}) + (1-\pi_j) N(\Lambda_{ij}|0,1) \qquad (4.3)$$

where π_j has a prior that has heavy concentration close to 1; $\delta_0(\cdot)$ denotes the unit point mass at 0; and Λ_{ij} denotes the i^{th} variable and j^{th} factor elements of the factor loading matrix $\boldsymbol{\Lambda}$. The sparse latent-factor model proposed by West (2003) worked well

in sparsifying the factor loading matrix $\boldsymbol{\Lambda}$ for small numbers of factors. Fitting a large number of factors in the sparse latent factor model proposed by West (2003) was computationally challenging.

Carvalho et. al (2008) extended the Bayesian specification of sparse factor model 4.3 proposed by West (2003) by adopting the sparsity prior of Lucas et. al (2006) as follows:

$$\Lambda_{ij} \sim (1 - \pi_{ij})\delta_0(\Lambda_{ij}) + \pi_{ij}\, N\left(\Lambda_{ij} | 0, \tau_j\right) \qquad (4.4)$$

where, Λ_{ij} denote the elements of the factor loading matrix $\boldsymbol{\Lambda}$; i represents variables; j represents the factors; and π_{ij} (the individual association probabilities or variable inclusion probabilities with any factor) represents prior probability that Λ_{ij} is exactly zero. Therefore, π_{ij} proposed in Equation (4.4) is an extension to π_j used in Equation (4.3). The selection of non-zero factor loading were based on the following posterior probabilities:

$$\hat{\pi}_{ij} = \Pr(\Lambda_{ij} \neq 0 | \boldsymbol{X}_{1:n}) \qquad (4.5)$$

Variable-factor significant relationships were defined based on higher $\hat{\pi}_{ij}$ values.

The extended Bayesian sparse factor model in Equation (4.4) improved the sparsity structure of the factor loading matrix as compared to the sparsity structure of factor loading matrix obtained using Equation (4.3).

Dependence in the multivariate data is typically characterized using Gaussian factor models. A method based on generation of posterior samples of loading matrices and covariance matrices was proposed by (Murray et. Al, 2013). Murray et. al (2013) developed bfa (Bayesian factor analysis) R - package to implement Gaussian factor models (bfa_{gauss} in the bfa package), Gaussian copula factor models (bfa_{copula}), and mixed-scale Gaussian factor models (bfa_{mixed}) for discrete margins probit specifications (the latter two are discussed in subsection 4.2.2). bfa has a drawback that the user must specify the number of factors. The bfa_{gauss} initializes and fits a Gaussian factor model by pre-specifying the number of MCMC iterations, and the number of factors to produce posterior samples of loading matrices and covariance matrices. Murray et. al (2013) proposed an efficient Gibbs sampling algorithm to produce the posterior samples of factor loading matrices, and covariance matrices.

The generalized form of a Gaussian factor model to the Gaussian copula factor model proposed by Murray et. al (2013) is as follows:

$$\eta_i \sim N(0, \boldsymbol{I}), \quad z_i|\boldsymbol{\eta_i} \sim N(\boldsymbol{\Lambda}\,\boldsymbol{\eta_i}, \boldsymbol{I}) \tag{4.6}$$

where z_i are the latent Gaussian variables, and different priors such as normal prior, GDP prior, and Pointmass prior can be used one at a time on $\boldsymbol{\Lambda}$. The GDP (generalized double Pareto) prior (Armagan and Dunson, 2011) is a default choice in the bfa R package, with density:

$$\pi(\lambda_{jh}) = \frac{\alpha}{2\beta}\left(1 + \frac{|\lambda_{jh}|}{\beta}\right)^{-(\alpha+1)} \tag{4.7}$$

The scaled loading are referred to as $\lambda_{jh} \sim GDP(\alpha, \beta)$, where α and β are scale parameters.

Inferring the number of factors in factor analysis has always been challenging (Lucas et. al, 2006; Carvalho et. al 2008) due to the sensitivity in choosing the priors subjectively. Bhattacharya and Dunson (2011) addressed this problem by proposing a multiplicative Gamma process shrinkage prior that allowed infinitely many factors. The factor loading matrix was shrunk to zero as the column index increased. They proposed Bayesian latent factors models to sparsify the factor loading matrices in a high-dimensional (large p) setting. Bhattacharya and Dunson (2011) developed an efficient Gibbs sampler for posterior computation that scaled well as the data dimension increased and picked the number of factors without pre-specifying the number of factors.

4.2.2 Bayesian Approaches (Non-Gaussian Case)

The latent variables influence both the form and dependence structure of the marginal distributions, complicating the interpretations of the Gaussian factor models when the Gaussian factor models are generalized to the non-normal measured variables. To decouple the marginal distributions from the latent factors, Murray et. al (2013) proposed a class of Gaussian copula factor models. Let the copula be denoted by $\mathbb{C}$ having a dimension p, then the joint distribution of copulas is as follows:

$$F(y_1, \ldots, y_p) = \mathbb{C}\left(F_1(y_1), \ldots, F_p(y_p)\right) \tag{4.8}$$

The Gaussian copula is as follows:

$$\mathbb{C}\left(u_1,\ldots,u_p\right) = \Phi_p(\Phi^{-1}(u_1),\ldots,\Phi^{-1}(u_p)/\boldsymbol{C}),(u_1,\ldots,u_p)\epsilon[0,1]^p, \qquad (4.9)$$

The Gaussian CDF with dimension p, and the correlation matrix $\boldsymbol{C}$ is represented by $\Phi_p(.|\boldsymbol{C})$, and the univariate standard normal cumulative distribution function is denoted by Φ. Therefore, the joint distribution of F can be written as follows:

$$F\left(y_1,\ldots,y_p\right) = \Phi_p(\Phi^{-1}\left(F_1(y_1)\right),\ldots,\Phi^{-1}\left(F_p(y_p)\right)|\boldsymbol{C}) \qquad (4.10)$$

The latent Gaussian variables are denoted by z, the pseudo-inverse of the marginal distributions F_j of F is expressed as follows:

$$F_j^{-1}(t) = \inf\{t:\ F_j(y) \geq t,\ y\,\epsilon\,\mathbb{R}\} \qquad (4.11)$$

Let the covariance matrix be denoted by $\boldsymbol{\Sigma}$, and $\boldsymbol{C}$ be its correlation matrix. If the latent Gaussian variables $z \sim N\,(0,\boldsymbol{\Sigma})$, and,

$$y_j = F_j^{-1}\left(\Phi\left({}^{z_j}\!/_{\sqrt{\omega_{jj}}}\right)\right) for\ 1 \leq j \leq p, \qquad (4.12)$$

Then the Gaussian copula is represented by $F\,(y)$ has the correlation matrix $\boldsymbol{C}$, and the univariate marginal distributions F_j.

The normal prior $\lambda_{jh}{\sim}N\left(0,\frac{1}{b}\right)$ is a common prior on unrestricted loading matrix for probit, Gaussian, or mixed factor models. However, the normal priors have some properties which are inappropriate when the model is not a Gaussian factor model i.e. mixed Gaussian/probit or probit factor models. The unique variances u_j in the Gaussian/probit, copula (Murray et. al, 2013), or probit factor models have implied prior when $\sigma_j \equiv 1$ as follows:

$$\pi(u_j) = \frac{({}^{b}\!/_{2})^{-k/2}}{\Gamma k/2}\left(\frac{1}{{u_j}^2}\right)\left(\frac{1-u_j}{u_j}\right)^{(k/2-1)} \times exp\left[-\frac{b}{2}\left(\frac{1-u_j}{u_j}\right)\right] \qquad (4.13)$$

The normal priors are quite informative on the scaled loadings for smaller k values, but they do not shrink $\check{\lambda}_{jh}$ toward zero values. Increasing the variance of the prior worsens this effect. The problem arises due to the normal prior putting insufficient mass near 0, which deflates u_j, produces spurious correlations, and assigns higher probability to the scaled loadings values close to ± 1. Therefore, the normal prior is a poor choice for this type of Gaussian copula factor models.

The copula factor model considers the continuous cases to interpret the conditional independence between two variables given the other variables. However, when the cases are discrete, or a mix of continuous and discrete, then care should be taken to interpret the coefficients using a copula factor model. The special cases of the Gaussian copula factor models are the probit factor models and the Gaussian factor model (Equation 4.1). The probit factor models are for ordered categorical or binary data for discrete margins, where the margins are parameterized by the "cut-points". These probit factor models extend to mixed-scale Gaussian factor models for discrete margins probit specifications. The ordered categorical or binary data can be modeled using the probit factor models. Let $\gamma_{j0},\ldots,\gamma_{jcj}$ be the collection of "cut-points" considering $\gamma_{j0} = -\infty$ and $\gamma_{jcj} = \infty$ so that $F_j(c) = \Phi\left(\gamma_{jc}\left(1+\sum_{h=1}^{k}\lambda_{jh}{}^2\right)^{-1/2}\right)$. Then the pseudoinverse of F_j is as follows:

$$F^{-1}(u_{ij}) = \sum_{c=1}^{c_j} c\mathbf{1}\left(\Phi\left(\frac{\gamma_{jc-1}}{\sqrt{1+\sum_{h=1}^{k}\lambda_{jh}{}^2}}\right) < u_{ij} \leq \Phi\left(\frac{\gamma_{jc}}{\sqrt{1+\sum_{h=1}^{k}\lambda_{jh}{}^2}}\right)\right) \qquad (4.14)$$

After plugging Equation (4.14) in Equation (4.12) and simplifying, we obtain the following expression for an ordinal probit factor model:

$$y_{ij} = \sum_{c=1}^{c_j} c\mathbf{1}\left(\gamma_{jc-1} < z_{ij} \leq \gamma_{jc}\right) \qquad (4.15)$$

where, $z_i \sim N(\mathbf{0}, \boldsymbol{\Lambda\Lambda^T} + \boldsymbol{I})$.

The bfa_mixed option in the bfa R package initializes and fits mixed-scale Gaussian factor model for discrete margins probit specifications. The bfa_mixed option pre-specifies the number of MCMC iterations, and the number of factors to produce posterior samples of loading matrices and covariance matrices (Murray et al, 2013). Chib and Greenberg (1998) also proposed a Bayesian multivariate probit model for the analysis of correlated binary data, with the drawback of not including the latent factor scores in their model. Hahn, Carvallho and Scott (2012) extended the existing Bayesian multivariate probit model proposed by (Chib and Greenberg, 1998) that included the latent factor scores in their model. The sparse factor probit model proposed by Hahn, Carvalho and Scott (2012) was based on drawing correlated posterior samples using a Gibbs sampler.

Alternative methods in this setting for covariance estimation, such as the L_1 regularization or banding, lack the interpretational advantages of Hahn, Carvalho and Scott's (2012) proposed sparse factor probit model, and cannot accommodate some useful modelling structures, such as spatial models and time series models. Hahn, Carvalho and Scott's (2012) sparse factor probit model could be considered as an exploratory tool for high-dimensional categorical data that are correlated.

Most research into factor analysis has been focused on count, continuous, and categorical data. The observed continuous data were converted to the z-scale by Quinn (2004) to combine the underlying categorical and ordered data with the latent z-scale. Quinn's method was extended by Murray et al (2013), where empirical inverse cumulative distribution function (CDF) was used on the z-scale for the placement of any variable class. Kim et al (2015) proposed sparse factor analysis ($\boldsymbol{SFA}$) for the estimation of underlying dimensionality of binary or count data, and then finding the transformed scale correlational structure. The $\boldsymbol{SFA}$ method differs from Murray et al's (2013) method in the way that the estimation of cut points is done. $\boldsymbol{SFA}$ was designed to combine textual data and vote data when estimating word effect, underlying dimensionality, and ideal points. The $\boldsymbol{SFA}$ method, Quinn's (2004) method, and Murray et al's (2013) method are all examples of Gaussian copula models.

4.2.3 Penalized Likelihood Approaches (Gaussian Case)

In factor analysis, different factor rotation techniques can be used to produce a factor loading matrix. Factor rotation has an alternative in the form of penalization for sparse estimation of factor loading matrix. Penalized factor analysis models using the L_1 penalty were independently proposed by Ning and Georgiou (2011) and Choi et al. (2010). Ordinary $Lasso$ based on L_1 penalization is biased towards producing overly dense models (Zhang, 2010; Zou, 2006). Choi et al. (2010) addressed this issue and proposed the adaptive $Lasso$ to obtain sparser solutions than ordinary $Lasso$. The penalized adaptive $Lasso$ estimator proposed by Choi et al (2010) is as follows:

$$\max_{\Lambda,\,\Psi} l(\boldsymbol{\Lambda}, \boldsymbol{\Psi}) - \rho \sum_{i=1}^{p} \sum_{j=1}^{q} \widehat{w}_{ij} \left|\lambda_{ij}\right| \qquad (4.16)$$

Where, $\widehat{w}_{ij} = \frac{1}{|\hat{\lambda}_{ij}|}$ .The adaptive $Lasso$ proposed by Choi et al. (2010) is different from an ordinary adaptive $Lasso$. The construction of $\widehat{w}_{ij}$ is dependent on an unpenalized

maximum likelihood estimator in ordinary adaptive $Lasso$. However, rotational indeterminacy causes an uncertain maximum likelihood estimator in factor analysis in adaptive $Lasso$ proposed by (Choi et al, 2010), and we are limited as to which maximum likelihood estimator should be used to construct $\widehat{w}_{ij}$. An adaptive $Lasso$ estimator proposed by Choi et al (2010) is more sophisticated in estimating $\widehat{w}_{ij}$ as compared to an ordinary adaptive $Lasso$, but is highly dependent on ρ to construct the weight $\widehat{w}_{ij}$. Hirose and Yamamoto (2015) addressed this issue and proposed a penalized factor analysis algorithm, $\boldsymbol{fanc}$ (factor analysis using non-convex penalties) for sparse estimation of factor analysis model, and to interpret changes in the factor loading matrix for a range of tuning parameters (ρ, γ).

$$\max_{\Lambda, \Psi} l(\boldsymbol{\Lambda}, \boldsymbol{\Psi}) - n \sum_{i=1}^{p} \sum_{j=1}^{q} \rho P\left(|\lambda_{ij}|;\ \rho, \gamma\right) \tag{4.17}$$

The non-convex penalty called the minimax concave penalty (MCP) (Zhang, 2010) is denoted by $P(\theta; \rho, \gamma)$.

$$\rho P(|\theta|;\ \rho, \gamma) = \int_0^{|\theta|} \left(1 - \frac{x}{\rho\gamma}\right)_+ dx \tag{4.18}$$

$$= \rho \left(|\theta| - \frac{\theta^2}{2\rho\gamma}\right) I(|\theta| < \rho\gamma) + \frac{\rho^2\gamma}{2} I(|\theta| \geq \rho\gamma)$$

A $Lasso$ penalty is yielded for each value of $\rho > 0$, and $\gamma \to \infty$. The shrinkage increases with an increase in the ρ values and decrease in the γ values.

The $\boldsymbol{fanc}$ algorithm is a two-step approach in which it computes and then shrinks some elements of the loading matrix based on the combinations of two sparsity-inducing penalties (ρ, γ) as follows:

$$fit < -fanc\ (X, k, \rho, \gamma) \tag{4.19}$$

where, X denotes the data matrix or covariance (correlation) matrix and k denotes the number of factors to be considered. The tuning parameter ρ varies and γ value is kept fixed. The $\boldsymbol{fanc}$ algorithm uses model selection criterion such as AIC,BIC, $EBIC$, and $CAIC$ as follows:

$$AIC = -2\ell\left(\widehat{\Lambda}, \widehat{\Psi}\right) + 2d_k, \tag{4.20}$$

$$BIC = -2\ell\left(\widehat{\Lambda}, \widehat{\Psi}\right) + (logN)d_k, \tag{4.21}$$

$$EBIC = BIC + 2d_k\delta\, log(pk), \tag{4.22}$$

$$CAIC = -2\ell\left(\widehat{\Lambda}, \widehat{\Psi}\right) + (logN + 1)d_k \tag{4.23}$$

where $\widehat{\Lambda}$ is the estimated factor loading matrix, $\widehat{\Psi}$ is the estimated diagonal matrix, p is the data dimension, k is the number of factors, and d_k denotes the non-zero parameters numbers. One of the advantages and flexibility of the $\boldsymbol{fanc}$ algorithm is that the final model can be selected based on any model selection criterion, such as AIC, BIC, $EBIC$, or $CAIC$. Incorrect number of factors can be suggested to $\boldsymbol{fanc}$ to see if it picks the true number of factors in the model, by way of having all loadings for some factors equal to zero. For example, suppose we generate data from a multivariate normal distribution with 2 true factors. We then estimate the observed covariance matrix of generated data, and use this observed covariance matrix as an argument in the $\boldsymbol{fanc}$ algorithm using an incorrect number of factors, say $k = 5$, for a range of sparsity-inducing penalties (ρ, γ). The $\boldsymbol{fanc}$ algorithm will produce sparse factor loading matrices with correct i.e. $k = 2$, or incorrect i.e. $2 < k \leq 5$ number of factors in the factor loading matrices.

Hui et al. (2018) proposed $\boldsymbol{OFAL}$ – Ordered factor $Lasso$ for factor order selection, and to achieve sparsity in $\boldsymbol{GLLVMs}$ (generalized linear latent variables models). The $\boldsymbol{OFAL}$ penalty was the first penalty that shrunk entire columns of the factor loading matrix to zero for latent variable models. $\boldsymbol{OFAL}$ achieved sparsity in the individual elements using adaptive $Lasso$. The $\boldsymbol{OFAL}$ estimator proposed by (Hui et al, 2018) is as follows:

$$\hat{\theta} = \underset{\theta}{^{argmax}}\, l(\theta) - ns\sum_{i=1}^{d}\omega_{1l}\left(\sum_{k=l}^{d}\sum_{j=1}^{p}\lambda_{jk}^2\right)^{\frac{1}{2}} - ns\sum_{j=1}^{p}\sum_{k=1}^{d}\omega_{2jk}\left|\lambda_{jk}\right| \tag{4.24}$$

where, the marginal likelihood is denoted by $l(\theta)$, and the positive adaptive weights are denoted by $\{\omega_{1l}; l = 1, \ldots, d\}$ and $\{\omega_{2jk}; j = 1, \ldots, p; k = 1, \ldots, d\}$. The $\boldsymbol{OFAL}$ penalty's first component in Equation (4.24) performs to achieve the factor order selection in the factor loading matrix by shrinking all the column entries of the factor loading matrix to zero. My proposed DSS method (see section 4.3) for non-normal responses could be extended to specify a parametric marginal distribution such as negative Binomial using the $\boldsymbol{OFAL}$ penalty. The $\boldsymbol{OFAL}$ penalty does not remove the first factor while retaining the second factor in the model. The tuning parameter s for

$\boldsymbol{OFAL}$ were selected using $\boldsymbol{ERIC}$ – Extended Regularized Information Criterion as follows:

$$ERIC(s) = -2l(\hat{\theta}) - log(s)\sum_{j=1}^{p}\sum_{k=1}^{d}\mathbb{1}(\hat{\lambda}_{jk} \neq 0) \qquad (4.25)$$

where, the marginal log-likelihood is denoted by $l(\hat{\theta})$, and $\mathbb{1}(\hat{\lambda}_{jk} \neq 0)$ is equal to 1 in case the estimated loading $\hat{\lambda}_{jk}$ is not shrunk to 0, and 0 elsewhere. Sparser estimates of the factor loading matrix were obtained using $ERIC$ for tuning parameter selection as compared to other tuning parameter selection criterion i.e. $(fanc: AIC, fanc: BIC, fanc: EBIC)$ (Hirose and Yamamoto, 2015).

4.2.4 Penalized Likelihood Approaches (Non-Gaussian Case)

Hui et al. (2018) also applied $\boldsymbol{OFAL}$ penalty on non-normal responses i.e. negative Binomial $\boldsymbol{GLLVM}$ on a lower dimension $p = 27$ and $n = 54$ species data (Hostie et al, 2003). Hui et al. (2018) considered only moderate dimensional data i.e. ($p = 30$) in the simulation studies they conducted.

4.3 Proposed Decoupled Shrinkage and Selection Sparse Factor Model

I propose a decoupled shrinkage and selection sparse factor model. I introduce a factor model for a particular k, and then obtain the simulated factor loading using the R package $\boldsymbol{bfa}$ (Murray et. al, 2013). I consider 10,000 MCMC iterations with 1,000 burn-ins to obtain the posterior samples of factor loading matrices of dimension $p \times k$. Then I thin the 10,000 posterior samples of factor loading matrices to 1000 by keeping every 10th posterior sample. We can explore the performance of my proposed decoupled shrinkage and selection sparse factor model for low to moderate-dimensional data as well for a range of number of factors. My main purpose is to shrink some elements of the factor loading matrix to exactly zero. After thinning the factor loading to 1,000 posterior samples of factor loading matrices, I use the following formula to convert the factor loading matrices to the implied covariance matrices:

$$\hat{\Sigma} = \hat{\Lambda}^T\hat{\Lambda} + \hat{\Psi} \qquad (4.26)$$

where, $\hat{\Sigma}$ is the estimated implied covariance matrix, $\hat{\Lambda}$ is the estimated factor loading matrix, and $\hat{\Psi}$ represents the unique variances, $\hat{\Psi} = diag(\sigma_1^2, \ldots, \sigma_p^2)$. After converting the 1,000 posterior samples of factor loading matrices to 1,000 posterior samples of covariance matrices using Equation (4.26), I average the 1,000 posterior samples of

implied covariance matrices and find the implied covariance mean. I then use the $\boldsymbol{fanc}$ algorithm (Hirose and Yamamoto, 2015) to process this posterior mean covariance matrix and infer the factor loading matrices for different combinations of ρ and γ, where ρ is varied and γ is a fixed value. We can use different dimensions, sample sizes, and number of factors in $\boldsymbol{fanc}$.

In Figure: 4.1, the x-axis shows the coverage of ρ used to select zero elements for $\widehat{\Lambda}$; the top axis shows the resulting number of factor loading in the factor loading matrices. The 90% credible interval for the fit of $\bar{\boldsymbol{\Sigma}}^{-1}$ (blue region), and its expected fit (central line) are shown for comparison. Let $\boldsymbol{\Sigma_k}$ denote the posterior samples of covariance matrices of future observations. Then:

$$fit(\bar{\boldsymbol{\Sigma}}^{-1}|\boldsymbol{\Sigma_k}) = log\left(det(\bar{\boldsymbol{\Sigma}}^{-1})\right) - tr(\boldsymbol{\Sigma_k}\bar{\boldsymbol{\Sigma}}^{-1}) \qquad (4.27)$$

The above Equation (4.27) is same as Equation (2.44) that appeared in chapter 2, denotes a sample from distribution of $fit(\bar{\Sigma}^{-1})$.

I vary the parameter ρ, and keep γ equal to a specific grid value in $\boldsymbol{fanc}$ algorithm. Progressively sparser estimates of the factor loading matrices are obtained for a sequence of ρ values, and a fixed γ value. It can be observed in Figure: 4.1 that there are flat regions where the fit is insensitive to small changes in the ρ values for a fixed $\gamma = 3.75$ value. This indicates that $\boldsymbol{fanc}$ algorithm is not altering the elements of Λ unless one can be shrunk all the way to zero. The last red dot within the 90% credible envelope represents the final sparse factor model. The factor loading matrix represents associations between the variables and the factors. A factor loading equal to zero means no relationship between that variable and that factor.

4.4 Simulation Studies

4.4.1 Continuous Case

I simulated the true factor model having the true factor loading matrices (Appendix A) using the multivariate Normal distribution for 18 different "scenarios"; i.e., combinations of dimensions $p = 30, 100$, sample sizes $n = 100, 500, 1000$, and number of factors $k = 2, 5, 10$.

The simulated data was then used as an input for bfa_{gauss} to obtain posterior samples of factor loading matrices for all 18 scenarios (Table: 4.1). I initially specified the correct number of factors, but then to assess robustness, incorrect number of factors k were allocated in the bfa_{gauss} package as follows:

- k = 5, when k = 2 in the true factor model.
- k = 10, when k = 5 in the true factor model.
- k = 15. When k = 10 in the true factor model.

I used the decoupled shrinkage and selection procedure explained in section 4.3. The final posterior factor loading matrices were selected using my proposed DSS sparse factor model, $\boldsymbol{fanc}$-based methods using the selection criterion such as AIC, BIC, and $CAIC$. The performance of my proposed DSS sparse factor model was compared with $\boldsymbol{fanc}$-based methods on the basis of TPR (true positive rate), TNR (true negative rate), and TDR (true discovery rate):

$$TPR = \frac{TP}{TP+FN} \tag{4.28}$$

$$TNR = \frac{TN}{TN+FP} \tag{4.29}$$

$$FDR = \frac{FP}{FP+TP} \tag{4.30}$$

$$TDR = 1 - FDR \tag{4.31}$$

4.4.1.1 Results

Initially I fixed the number of factors k in $\boldsymbol{fanc}$ to the exact number of factors as in the true factor model. The main purpose of fixing the number of factors in $\boldsymbol{fanc}$ exactly equal to the number of factors in the true factor model was to compare the performance of my proposed DSS sparse factor model with $\boldsymbol{fanc}$-based methods for the same number of factors in $\boldsymbol{fanc}$ and my proposed DSS sparse factor model. I vary the parameter ρ, and keep γ always equal to $\gamma = 3.75$ in $\boldsymbol{fanc}$ algorithm. The true factor loading matrices for all the 6 combinations of p, and k are given in the Appendix A.

I have compared the performance measures of my proposed DSS sparse factor model with $fanc{:}\,AIC$, $fanc{:}\,BIC$, and $fanc{:}\,CAIC$, on the basis of averages over 50

replications of TPR (true positive rate), TNR (true negative rate), and TDR (true discovery rate) in Table: 4.1.

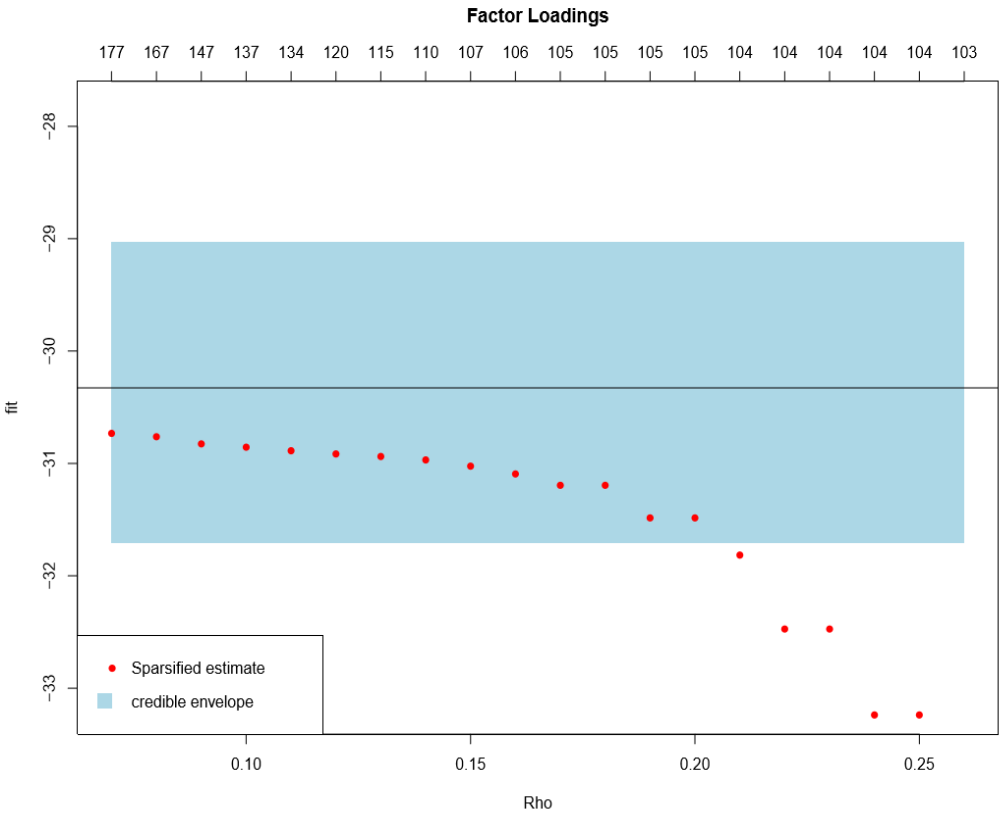


Figure 4.1: Expected fit for sparse summaries of factor loading for continuous case considering $p = 100, n = 500, k = 5$. Rho is the tuning parameter. Range of fit of the posterior mean inverse is along the $y - axis$. Blue area represents 90% credible interval for the range of fit of posterior mean inverse.

For a lower dimension i.e. $p = 30$, my proposed DSS sparse factor model outperforms all the other $\boldsymbol{fanc}$ – based methods by having the highest TNR and TDR for most of the cases. My proposed DSS sparse factor model has the best TNR and TDR for $p = 30, n = 100, k = 10$. However, $fanc{:}\,AIC$ has the highest TPR for almost all the cases.

However, for a higher dimension i.e. $p = 100,$ my proposed DSS sparse factor model has the highest TNR for all the cases. My proposed DSS sparse factor model is more specific than $\boldsymbol{fanc}$ – based methods, and $fanc{:}\,AIC$ is more sensitive for sparsifying the factor loading matrices for almost all the possible p, n, k combinations detailed in Table: 4.1.

Table: 4.1 (Performance Measures Comparisons – Continuous Case)

$p = 30$													
n	Performance Measures	$k = 2$				$k = 5$				$k = 10$			
		DSS FA	$fanc$ AIC	$fanc$ BIC	$fanc$ CAIC	DSS FA	$fanc$ AIC	$fanc$ BIC	$fanc$ CAIC	DSS FA	$fanc$ AIC	$fanc$ BIC	$fanc$ CAIC
100	TPR (%)	50	**90**	88	86	22	**67**	57	38	12	**50**	22	20
	TNR (%)	**57**	11	13	18	**95**	32	43	69	**99**	50	85	91
	TDR (%)	**52**	50	50	51	**58**	20	21	25	**66**	10	16	22
500	TPR (%)	60	**89**	84	81	18	**37**	36	34	9	**18**	17	15
	TNR (%)	**60**	9	13	17	**87**	69	71	74	**97**	89	90	92
	TDR (%)	**62**	49	49	49	**26**	23	23	23	**22**	16	16	18
1000	TPR (%)	48	**89**	85	84	20	**31**	**31**	**31**	11	**19**	18	14
	TNR (%)	**48**	13	17	20	**85**	75	76	76	**97**	91	92	92
	TDR (%)	46	**51**	50	50	24	**25**	**25**	**25**	**29**	18	19	15
$p = 100$													
n	Performance Measures	$k = 2$				$k = 5$				$k = 10$			
		DSS FA	$fanc$ AIC	$fanc$ BIC	$fanc$ CAIC	DSS FA	$fanc$ AIC	$fanc$ BIC	$fanc$ CAIC	DSS FA	$fanc$ AIC	$fanc$ BIC	$fanc$ CAIC
100	TPR (%)	53	**98**	98	98	25	**93**	92	91	12	**87**	85	84
	TNR (%)	**20**	3	3	3	**72**	6	8	10	**88**	13	14	16
	TDR (%)	39	**50**	**50**	**50**	19	**20**	**20**	**20**	**10**	**10**	**10**	**10**
500	TPR (%)	49	**97**	96	96	18	**91**	71	69	10	**79**	64	64
	TNR (%)	**43**	3	3	3	**79**	9	29	30	**90**	21	36	36
	TDR (%)	46	**50**	**50**	**50**	17	**20**	**20**	**20**	**10**	**10**	**10**	**10**
1000	TPR (%)	60	**97**	**97**	**97**	23	**86**	63	63	11	**70**	56	56
	TNR (%)	**59**	3	3	3	**81**	14	34	34	**90**	31	45	45
	TDR (%)	**60**	50	50	50	**22**	20	19	19	**11**	10	10	10

**The best cases are bold.*

It can be concluded that my DSS sparse factor model is to be preferred in situations where specificity is important.

Sparser estimates of factor loading matrices are obtained by varying ρ, and keeping γ to a fixed value. I conducted initial experiments with different γ values, and used $\gamma = 3.75$ in the simulation studies since it produced appropriately sparse factor loading matrices (smaller γ values produced sparser factor loading).

4.4.1.2 Identification of Correct Number of Factors for Continuous Case

The exact number of factors in the true factor models were $k = 2, 5, 10$ respectively both for $p = 30$, and $p = 100$. However, in order to compare my proposed DSS sparse factor model with $\boldsymbol{fanc}$ – based methods for identification of correct number of factors as in the true factor models, I allocated incorrect number of factors both in $\boldsymbol{fanc}$ and bfa_gauss as follows:

- k = 5, when k = 2 in the true factor model.
- k = 10, when k = 5 in the true factor model.
- k = 15. When k = 10 in the true factor model.

Table: 4.2 compares my proposed DSS sparse factor model with $fanc: AIC$, $fanc: BIC$, and $fanc: CAIC$ for the identification of the correct number of factors.

For a lower dimension i.e. $p = 30, n = 100, 500, 1000, k = 2,$ my proposed DSS sparse factor model outperforms the $\boldsymbol{fanc}$-based methods. However, for the higher k values such as $k = 5, 10,$ $fanc: AIC$ and $fanc: BIC$ outperform my DSS sparse factor model by discovering more factors. This behaviour is consistent with the higher specificity for DSS observed in subsection 4.4.1.1. For a higher dimension i.e. $p = 100$, my proposed DSS sparse factor model outperforms the $\boldsymbol{fanc}$-based methods in identifying the correct number of factors as in the true factor model.

4.4.2 Discrete Case

The main purpose of the simulations studies for the discrete case was to choose between the two priors namely the GDP prior (Equation 4.7) and the Pointmass prior (Equation 4.3) to give as an argument in the bfa_mixed package. I simulated the true factor model using multivariate Normal distribution considering 18 different

Table: 4.2 (Correct Number of Factors Comparisons)

$p = 30$													
	True Factors	***k = 2***				***k = 5***				***k = 10***			
n	***Factors given in $fanc$ & bfa***	***k = 5***				***k = 10***				***k = 15***			
	Descriptive Statistics	***DSS FA***	***$fanc$ AIC***	***$fanc$ BIC***	***$fanc$ CAIC***	***DSS FA***	***$fanc$ AIC***	***$fanc$ BIC***	***$fanc$ CAIC***	***DSS FA***	***$fanc$ AIC***	***$fanc$ BIC***	***$fanc$ CAIC***
100	***Mean***	**2.0**	5.0	5.0	4.5	2.0	10.0	**5.6**	3.4	1.9	15.0	**5.5**	4.4
	Minimum	**2.0**	5.0	5.0	**2.0**	2.0	10.0	2.0	2.0	1.0	15.0	3.0	3.0
	Maximum	**2.0**	5.0	5.0	5.0	2.0	10.0	10.0	**6.0**	3.0	15.0	**9.0**	7.0
500	***Mean***	**2.0**	3.3	2.4	2.2	3.6	**4.2**	4.0	3.8	4.0	**6.1**	5.9	5.3
	Minimum	**2.0**	**2.0**	**2.0**	**2.0**	2.0	**4.0**	**4.0**	2.0	3.0	**5.0**	4.0	4.0
	Maximum	**2.0**	5.0	4.0	4.0	4.0	6.0	**5.0**	4.0	5.0	**7.0**	6.0	6.0
1000	***Mean***	**2.0**	2.7	2.3	2.2	**4.0**	**4.0**	**4.0**	**4.0**	4.2	**6.1**	6.0	5.8
	Minimum	**2.0**	**2.0**	**2.0**	**2.0**	3.0	**4.0**	**4.0**	**4.0**	4.0	**6.0**	5.0	4.0
	Maximum	**2.0**	5.0	4.0	4.0	4.0	**5.0**	4.0	4.0	5.0	**7.0**	6.0	6.0
$p = 100$													
100	***Mean***	**2.0**	5.0	5.0	5.0	**5.0**	10.0	10.0	10.0	**10.0**	15.0	15.0	15.0
	Minimum	**2.0**	5.0	5.0	5.0	**5.0**	10.0	10.0	10.0	**10.0**	15.0	15.0	15.0
	Maximum	**2.0**	5.0	5.0	5.0	**5.0**	10.0	10.0	10.0	**10.0**	15.0	15.0	15.0
500	***Mean***	**2.0**	5.0	5.0	5.0	**5.0**	10.0	10.0	10.0	**10.0**	15.0	15.0	13.4
	Minimum	**2.0**	5.0	5.0	5.0	**5.0**	10.0	10.0	10.0	**10.0**	15.0	15.0	**10.0**
	Maximum	**2.0**	5.0	5.0	5.0	**5.0**	10.0	10.0	10.0	**10.0**	15.0	15.0	15.0
1000	***Mean***	**2.0**	5.0	5.0	5.0	**5.0**	10.0	9.5	8.3	**10.0**	14.9	11.5	10.9
	Minimum	**2.0**	5.0	5.0	5.0	**5.0**	10.0	**5.0**	**5.0**	**10.0**	13.0	**10.0**	**10.0**
	Maximum	**2.0**	5.0	5.0	5.0	**5.0**	10.0	10.0	10.0	**10.0**	15.0	15.0	13.0

**The best cases are bold.*

combinations of dimensions $p = 30, 100$, sample sizes $n = 100, 500, 1000$, and number of factors $k = 2, 5, 10$. After that, I set a cut-off value of 0.5 on the simulated data to convert it into multivariate binary data. The values less than 0.5 were converted to 0, and the values greater than or equal to 0.5 were converted to 1.

The simulated data was then used as an input in the R package – bfa_mixed to obtain posterior samples of factor loading matrices for all 18 scenario (Table: 4.3). Incorrect number of factors k were allocated in the bfa_mixed package as follows for a lower dimension i.e. $p = 30$ case:

- k = 5, when k = 2 in the true factor model.
- k = 10, when k = 5 in the true factor model.
- k = 15. When k = 10 in the true factor model.

However, correct number of factors i.e. k = 2, k = 5, $and\ k$ = 10 were given in bfa_mixed package for a higher dimension i.e. $p = 100$ case.

I repeated the decoupled shrinkage and selection procedure explained in section 4.3. I varied the parameter ρ, and kept γ always equal to the 7th grid value i.e. $\gamma = 3.75$ in $\boldsymbol{fanc}$ algorithm. The final posterior factor loading matrices were selected using my proposed DSS sparse factor model using the "GDP prior" and the "Pointmass prior" separately, and the performance was compared as well on the basis of TPR, TNR, and TDR. The expected fit for sparse summaries of factor loading for discrete case both for the "GDP prior" and the "Pointmass prior" displayed in the Figures: 4.2 and 4.3 showed that the fit was sparser using the "Pointmass prior" than using the "GDP prior".

The expected fit using the "GDP prior" selected the factor loading matrix that was only 13% sparse (Figure: 4.2). Whereas, the expected fit using the "Pointmass prior" selected the factor loading matrix that was 46% sparse (Figure: 4.3). However, the true sparsity level was 50%.

4.4.2.1 Results

Initially I fixed the number of factors k in $\boldsymbol{fanc}$ to the exact number of factors in the true factor model to assess the performance of my proposed DSS sparse factor model based on "GDP prior" and "Pointmass prior" for the same number of factors in $\boldsymbol{fanc}$ and the number of factors given in bfa. For a lower dimension i.e. $p = 30$, the factor

loading matrices using “Pointmass prior” had the highest $TPR, TNR, and\ TDR$ for most of the cases (Table: 4.3). For a higher dimension i.e. $p = 100$, the factor loading matrices using “GDP prior” had the highest TPR for most of the cases. However, factor loading matrices using “Pointmass prior” still had better $TNR, and\ TDR$ for most of the

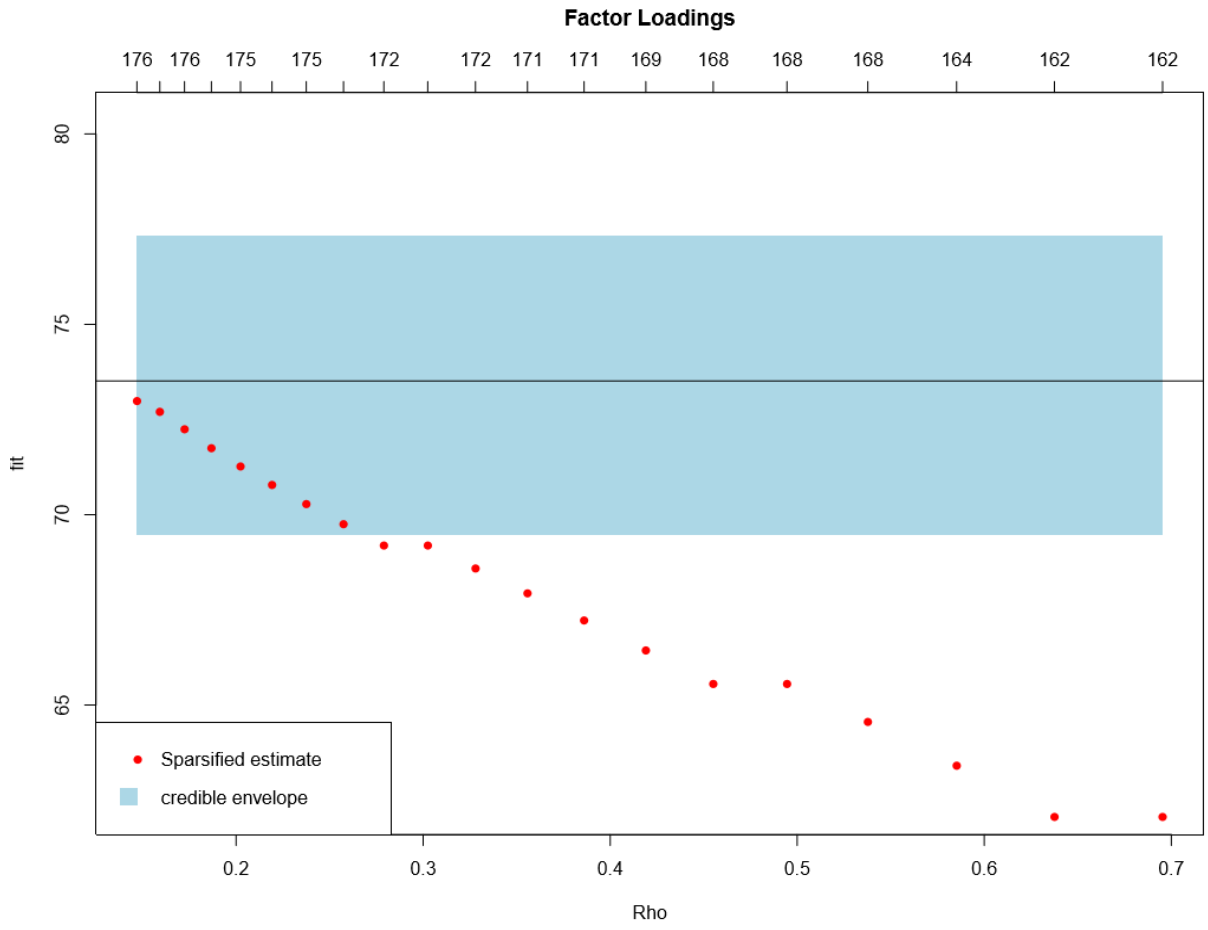


Figure 4.2: Expected fit for sparse summaries of factor loading for discrete case considering $p = 100, n = 100, k = 2$. and GDP prior.

cases. For the rest of the cases, “GDP prior” and “Pointmass prior” were competitive. It is evident from Figures: 4.2, 4.3, and Table:4.3 that “Pointmass prior” produced sparser factor loading matrices as compared to the factor loading matrices produced using “GDP prior” since factor loading matrices using “Pointmass prior” had the highest TNR for most of the cases confirming the presence of more zeros in the factor loading matrices (Table: 4.3).

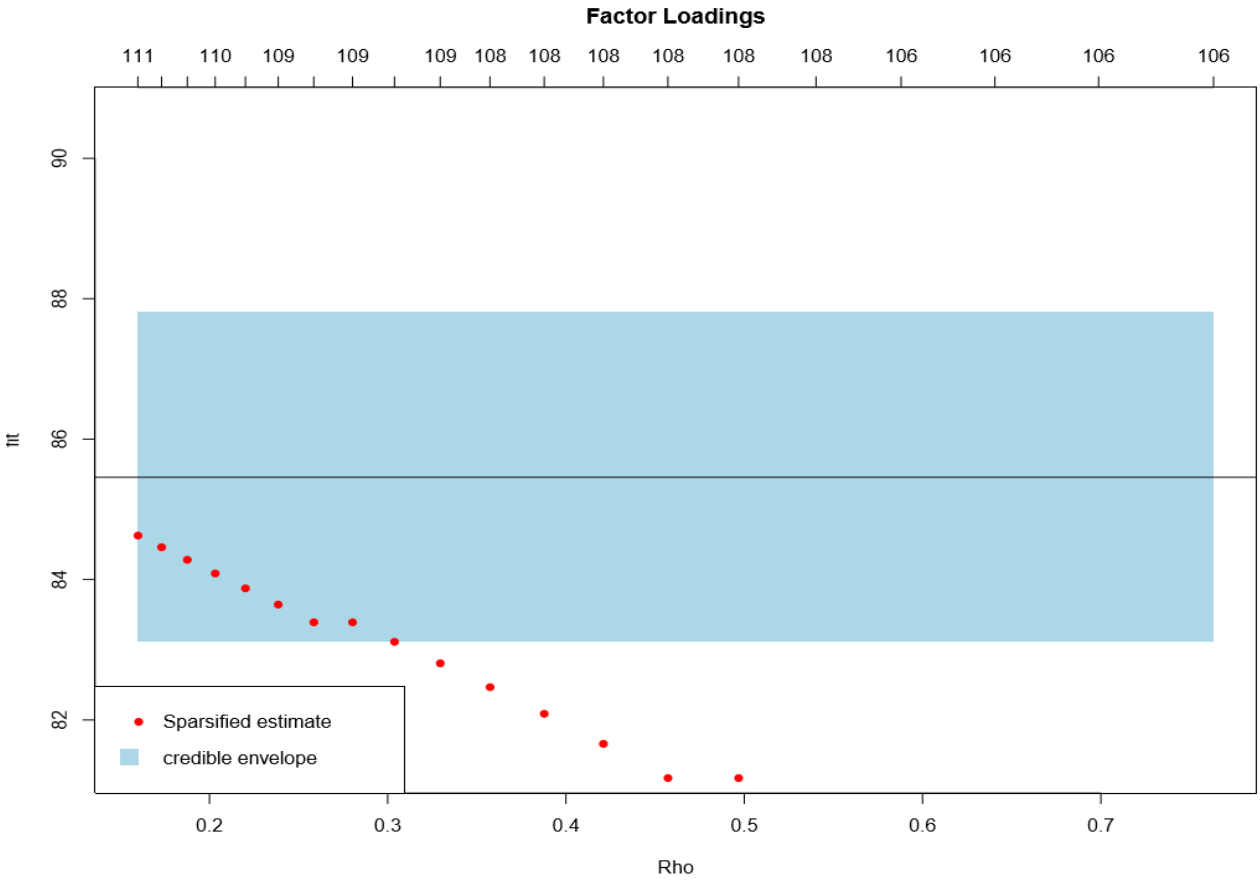


Figure 4.3: Expected fit for sparse summaries of factor loading for discrete case considering $p = 100, n = 100, k = 2$. ***and Pointmass prior.***

4.5 Examples

I considered data from a food frequency questionnaire that had been converted to amounts in grams for 56 different foods (Mumme et. al, 2019). A further example is provided for binary data, using the presence and absence of different fish species (Smith, Duffy and Leathwick, 2013).

4.5.1 Food Questionnaire Data (Continuous Case)

Data on $p = 56$ food items comprising of fruits, vegetables, meat, drinks, processed food, and poultry were collected from $n = 367$ respondents using a food frequency questionnaire. I used the food questionnaire standardized data as an input in

Table: 4.3 (Performance Measures Comparisons – Discrete Case)

$p = 30$							
n	Performance Measures	$k = 2$		$k = 5$		$k = 10$	
		GDP	Pointmass	GDP	Pointmass	GDP	Pointmass
100	TPR (%)	**62**	40	**28**	20	**14**	2
	TNR (%)	51	**61**	81	**96**	90	**100**
	TDR (%)	**56**	48	31	**55**	13	**74**
500	TPR (%)	46	**53**	16	**19**	**10**	9
	TNR (%)	43	**55**	83	**94**	96	**97**
	TDR (%)	45	**54**	17	**42**	22	**24**
1000	TPR (%)	**55**	54	**25**	16	**11**	8
	TNR (%)	**54**	**54**	86	**88**	**97**	96
	TDR (%)	**52**	**52**	**29**	22	**28**	23
$p = 100$							
n	Performance Measures	$k = 2$		$k = 5$		$k = 10$	
		GDP	Pointmass	GDP	Pointmass	GDP	Pointmass
100	TPR (%)	**87**	66	**73**	21	**50**	10
	TNR (%)	12	**55**	28	**77**	48	**89**
	TDR (%)	50	**55**	**20**	**20**	**10**	9
500	TPR (%)	**66**	51	**41**	22	**40**	11
	TNR (%)	28	**36**	56	**78**	60	**89**
	TDR (%)	**51**	41	19	**20**	**10**	9
1000	TPR (%)	**63**	43	**32**	22	**28**	10
	TNR (%)	**40**	29	68	**78**	73	**90**
	TDR (%)	**53**	37	20	**21**	**10**	**11**

**The best cases are bold.*

bfa_copula and used the "GDP prior" to obtain the posterior samples of factor loading matrices. I used copula because it was not necessarily Gaussian as shown in the histograms for common foods such as apples/pears, and bananas. Both the histograms for apples/pears, and bananas depict skewed distributions shown in Figures: 4.4 and 4.5.

The number of factors given to the bfa_copula package was $k = 2$ that is concordant with the 2 dietary patterns derived by principal component analysis with rotation for the same dataset (Mumme et. al, 2019).

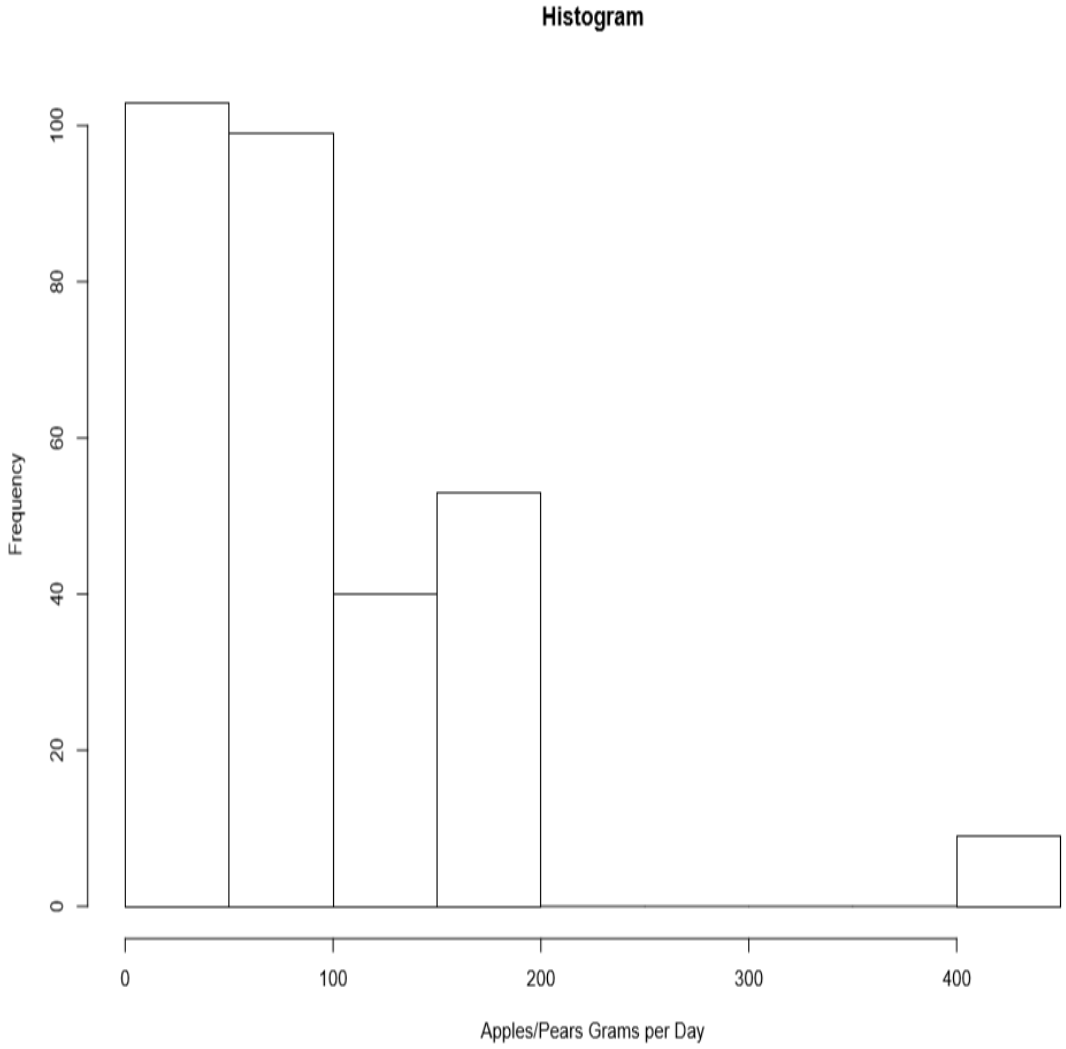


Figure 4.4: Histogram of apples/pears intake in grams per day.

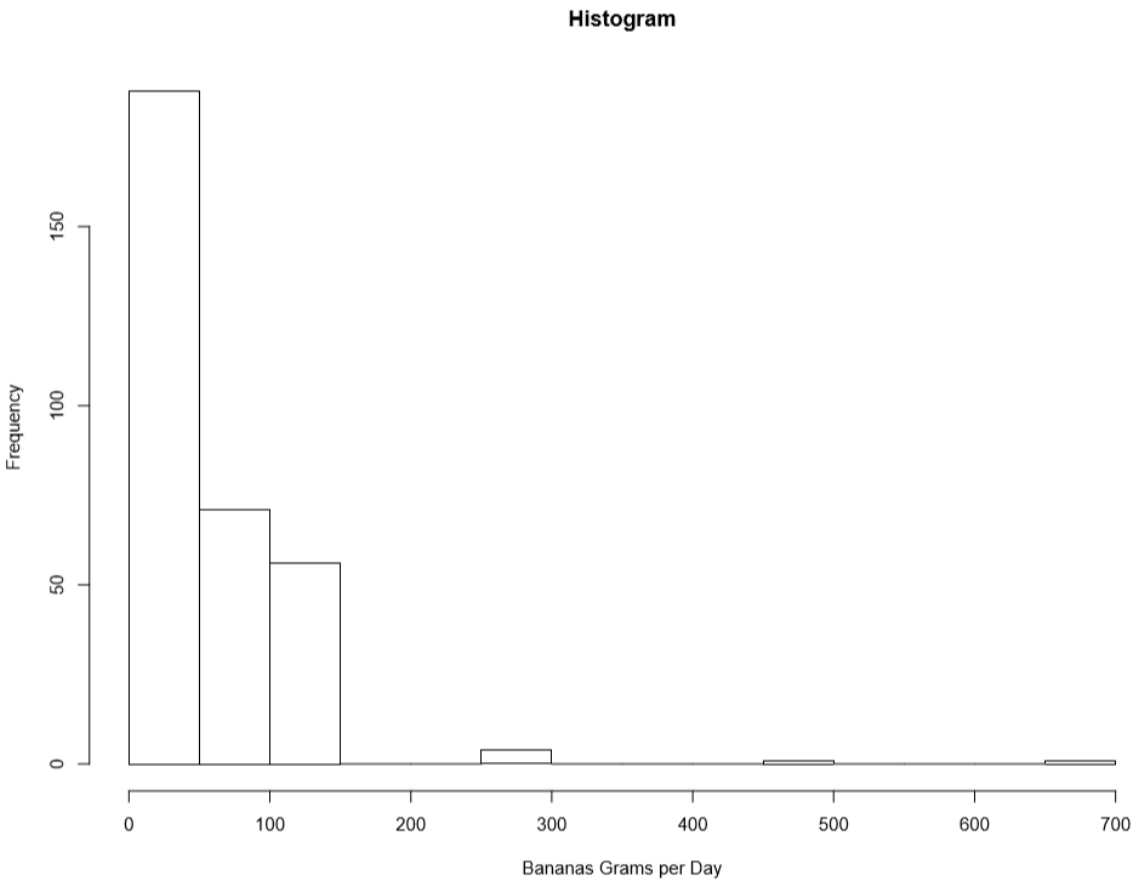


Figure 4.5: Histogram of bananas intake in grams per day.

I repeated the decoupled shrinkage and selection procedure explained in section 4.3. I varied the parameter ρ, and kept γ equal to the 9th grid value i.e. $\gamma = 1.01$ in $\boldsymbol{fanc}$ algorithm (Figure: 4.6). The final posterior factor loading matrices were selected using my proposed DSS sparse factor model. The last red dot within the blue area represented the selected factor loading matrix. The final selected factor loading matrix had 71 non-zero, and 41 zero factor loading out of 112 possible factor loading making it 37% sparse.

Table: 4.4 displays the factor loading matrix selected for the food questionnaire data using my proposed DSS sparse factor model. The consumption of processed food, drinks, and meat are making the **1st** factor. Fruits and vegetables are making the **2nd** factor. My proposed DSS sparse factor model has the advantage that the foods

loading on each factor are selected automatically. Principal components loading are typically subjected to an arbitrary threshold prior to interpretation (Mumme et. al, 2019).

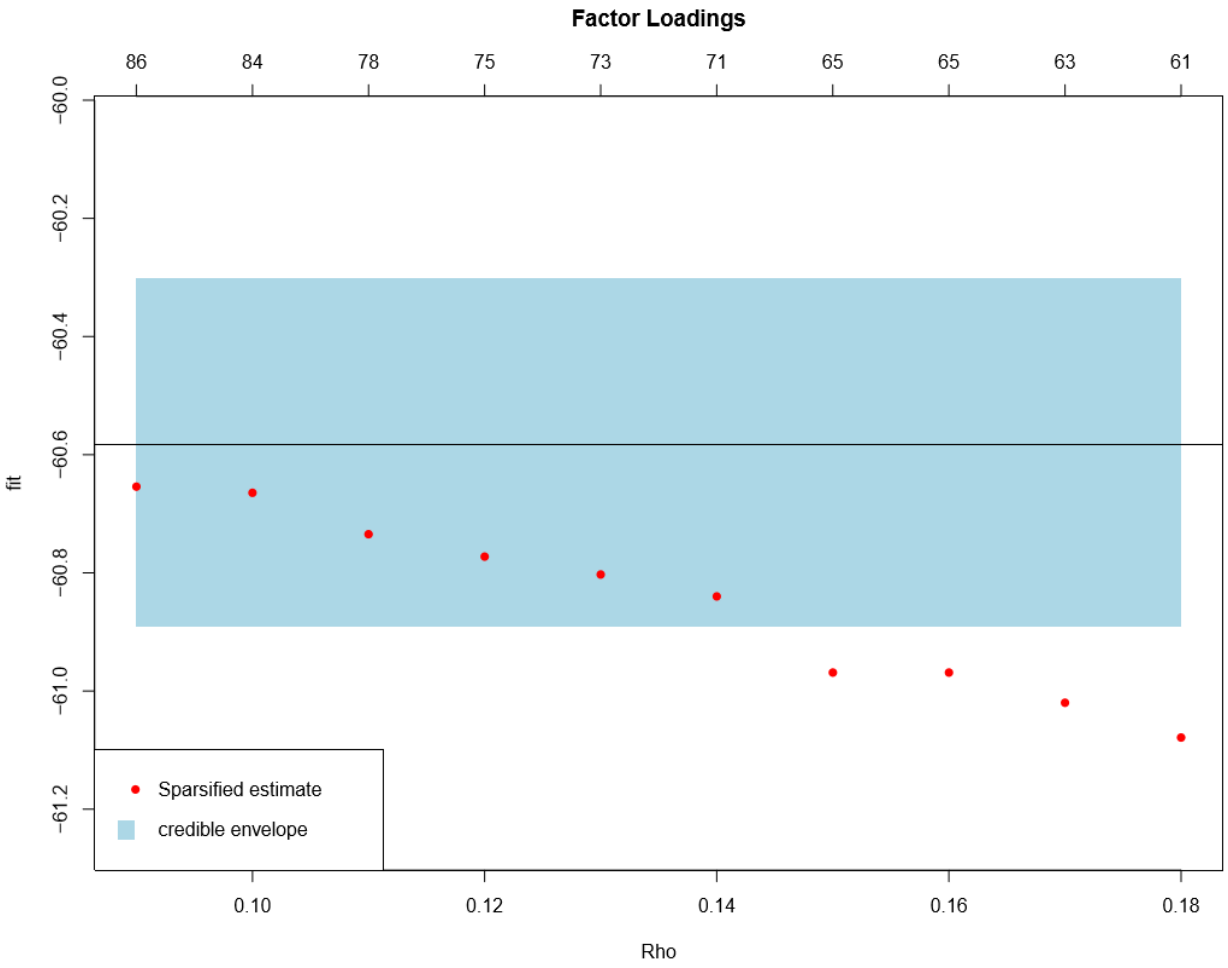


Figure 4.6: Expected fit for sparse summaries of factor loading for Food Questionnaire data considering $k = 2$.

4.5.2 NZ Reef Fish Abundance Data (Discrete Case)

I applied my DSS sparse factor model to a dataset of occurrences of NZ reef fish (Smith, Duffy and Leathwick, 2013). The data were collected for $p = 158$ species at $n = 467$ surveys by scuba divers, made around coastal New Zealand between 1986 and 2004.

I selected the fish that were observed at least 20 times across the 467 locations, reducing the data to $p = 72$ variables. The non-zero entries in the data were converted

Table: 4.4 (Factor Loading Matrix for Food Questionnaire Data)

Food Groups	Factor1	Factor2
All other fruit	-	0.57
Alliums	-	0.61
Alternate	(0.21)	0.41
Apple/Pear	-	0.36
Banana	-	-
Beer	0.53	-
Berry fruits	-	0.54
Biscuits, cakes and pastries	0.66	(0.18)
Bran cereal	-	-
Breakfast cereal	0.34	-
Carrots	-	0.52
Cheese and creamy dairy	0.45	0.23
Chocolate	0.34	0.21
Citrus etc	-	0.30
Confectionery	0.52	(0.22)
Cruciferous	-	0.48
Diet drinks	0.37	(0.18)
Dressings	0.57	0.41
Dried Fruit	-	0.35
dried Legumes	-	0.45
Eggs	-	0.36
fresh/frozen Legumes	-	0.25
Green leafy cruciferous	-	0.83
Juices	0.23	-
Milk	0.27	-
Nuts and seeds	(0.23)	0.62
Oily fish	-	0.51
Olives and Avocados	-	0.60
Other Alcohol	0.17	-
Other milks etc	(0.19)	0.32
Other sweetened Dairy	0.56	-
Other vegetables	-	0.83
Poultry	0.21	-
Processed fish	0.44	-
Processed Meats	0.83	(0.25)
Red Meat	0.46	-
Red Wine	0.29	-
Refined grain	0.45	-
Root/starchy vegetables	0.27	0.32
Salad vegetables	-	0.74
Sauces, chutneys etc	0.66	0.44
Savoury	0.69	(0.32)
SFA	-	0.22
Shellfish	0.19	0.33
Soup	-	-
Spices	-	0.51
Stone fruit	0.22	0.30
sugared drinks	0.29	(0.17)
Tea and coffee	0.23	-
Tomatoes	0.20	0.49
USFA	0.31	-
Water	(0.17)	0.48
White fish	-	0.42
Wholegrain	0.23	0.25
Yeast spread	-	-
Yoghurt	-	0.31

to 1 to make the dataset multivariate binary. I used NZ reef fish data as an input in the bfa_mixed package and used "Pointmass prior" to obtain the posterior samples of factor loading matrices. I kept different number of factors such as $k = 2, k = 3, k = 4, k = 5$ in bfa_mixed package separately to check which number of factors input was producing the most meaningful interpretations of the factor loading matrices for the fish data. The bfa_mixed package with $k = 2$ produced the most interpretable results. I repeated the decoupled shrinkage and selection procedure explained in section 4.3. I varied the parameter ρ, and kept γ equal to the 9th grid value i.e. $\gamma = 1.01$ (Hirose and Yamamoto, 2015) in $\boldsymbol{fanc}$ algorithm (Figure: 4.7). The final posterior factor loading matrices were selected using our proposed DSS sparse factor model. The last

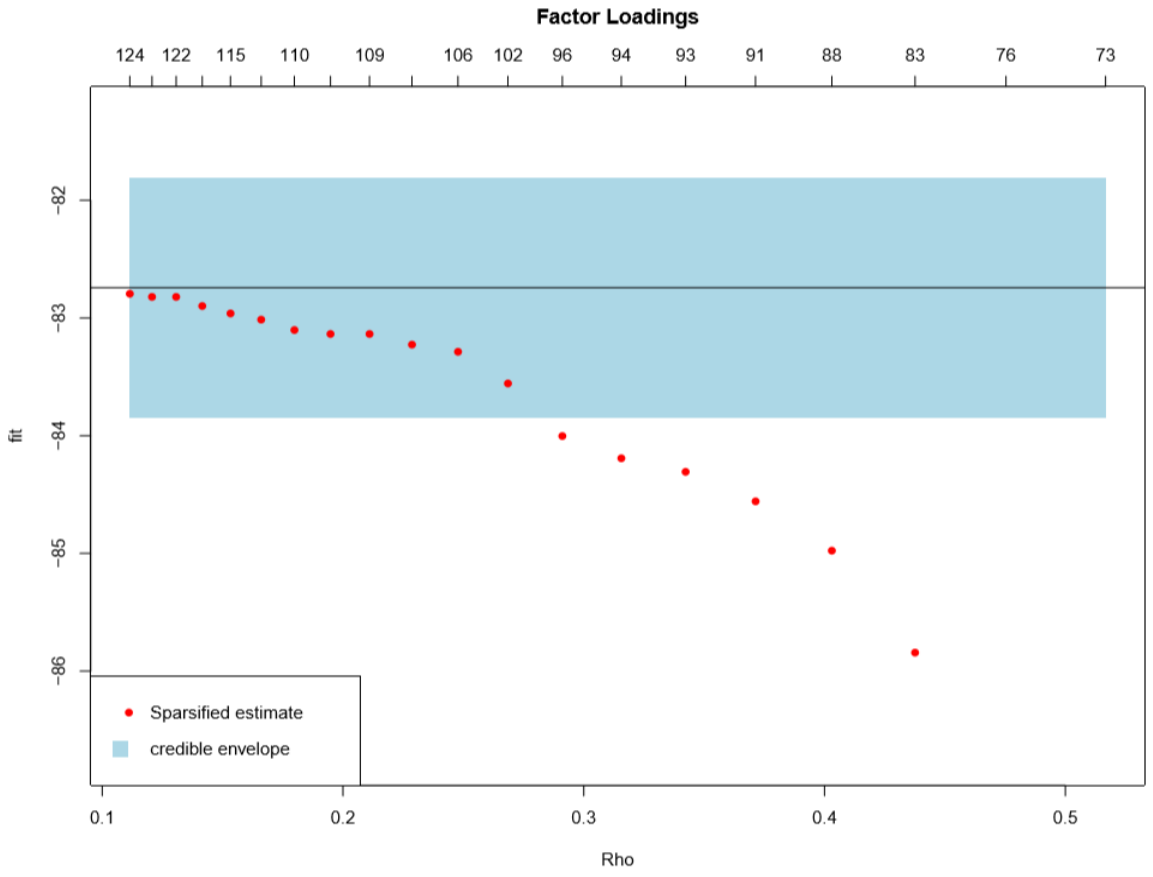


Figure 4.7: Expected fit for sparse summaries of factor loading for NZ Reef Fish data considering $k = 2$.

red dot within the blue area represented the selected factor loading matrix. The final selected factor loading matrix had 102 non-zero, and 42 zero factor loading out of 144 possible factor loading making it 29% sparse.

Table: 4.5 (Factor Loading Matrix for NZ Reef Fish Data)

NZ Reef Fish	Factor 1	Factor 2
Dasyatis brevicaudata	-	0.51
Myliobatis tenuicaudatus	-	0.29
Gymnothorax nubilus	-	2.09
Gymnothorax prasinus	0.71	1.66
Conger verreauxi	0.52	-
Lotella rhacina	0.67	-
Pseudophycis barbata	0.44	-
Optivus elongatus	0.94	0.86
Paratrachichthys trailli	0.86	-0.63
Centroberyx affinis	-	1.05
Zeus faber	0.50	0.72
Helicolenus percoides	0.32	-1.10
Scorpaena cardinalis	0.45	2.78
Scorpaena papillosa	0.55	-
Caesioperca lepidoptera	0.63	-
Caprodon longimanus	-	1.84
Hypoplectrodes huntii	0.81	-0.30
Hypoplectrodes dimidius	0.33	2.79
Decapterus koheru	0.69	1.52
Pseudocaranx dentex	-	0.95
Seriola lalandi	-	0.90
Trachurus novaezelandiae	-	-
Arripis trutta	-	-
Pagrus auratus	0.58	1.63
Upeneichthys porosus	0.31	0.55
Pempheris adspersa	1.06	1.92
Atypichthys latus	-0.67	1.32
Scorpis lineolata	0.70	0.60
Scorpis violacea	-	1.75
Kyphosus sydneyanus	-	0.59
Girella cyanea	-0.61	1.41
Girella tricuspidata	0.67	0.45
Amphichaetodon howensis	-0.82	1.92
Chromis dispilus	0.52	2.98
Parma alboscapularis	-	1.87
Chironemus marmoratus	0.64	0.70
Aplodactylus arctidens	0.69	-
Cheilodactylus spectabilis	0.84	0.76
Nemadactylus douglasii	0.69	1.87
Nemadactylus macropterus	0.56	-0.48
Latridopsis ciliaris	0.46	-0.43
Latris lineata	-	-1.32
Mendosoma lineatum	0.40	-1.92
Aldrichetta forsteri	-0.30	-0.56
Notolabrus celidotus	-	-0.89
Notolabrus cinctus	-	-2.33
Notolabrus fucicola	0.90	-
Notolabrus inscriptus	-0.53	1.38
Pseudolabrus luculentus	-	3.24
Pseudolabrus miles	0.98	-0.42
Coris sandeyeri	-	2.83
Suezichthys aylingi	-	1.47
Bodianus unimaculatus	1.00	2.85
Odax pullus	0.72	-
Parapercis colias	-	-1.10
Forsterygion flavonigrum	0.43	-
Forsterygion lapillum	-0.29	-0.49
Forsterygion malcolmi	0.98	-0.66
Forsterygion varium	0.31	-0.87
Forsterygion gymnotum	-	-
Karalepis stewarti	0.63	-
Notoclinops caerulepunctus	0.81	-
Notoclinops segmentatus	0.73	-
Notoclinops yaldwyni	0.64	0.78
Forsterygion maryannae	0.81	-
Ruanoho decemdigitatus	-	-0.42
Ruanoho whero	0.69	-
Parablennius laticlavius	-	1.43
Plagiotremus tapeinosoma	-0.50	2.02
Thalasseleotris adela	-	-
Meuschenia scaber	0.51	0.64
Canthigaster callisterna	-	1.96

Factor **1** (Table: 4.5) identifies species that inhabit deeper, exposed, high-current, temperate sites, such as the outer Marlborough Sounds and outer Hauraki Gulf. The species whose loadings are set to zero for factor 1 are those typically found in shallow water and/or subtropical climates.

Factor **2** (Table: 4.5) species with positive values are those that occur in the north, and with negative values are those that occur in the south. The species for which the loadings are set to zero are those that do not follow any clear latitudinal gradient.

4.6 Conclusions and Recommendations

My proposed decoupled shrinkage and selection sparse factor model shrinks some elements (correlations between observed variables and factors) of the factor loading matrix to exact zeros. The performance comparisons based on the true positive rate, true negative rate, and true discovery rate between my proposed DSS sparse factor model and $\boldsymbol{fanc}$ – based methods (Hirose and Yamamoto, 2015) for continuous case in simulation studies indicated that DSS sparse factor model had the higher true negative rate and true discovery rate than $\boldsymbol{fanc}$ – based methods for a low dimensional data i.e. $p = 30$ for most of the cases, and the highest true negative rate than $\boldsymbol{fanc}$ – based methods for all the cases for a high dimensional data i.e. $p = 100$.

Simulation studies for discrete case showed that DSS sparse factor model using "Pointmass prior" had the highest true positive rate, true negative rate, and true discovery rates for most of the cases for a low dimensional data i.e. $p = 30$, and the highest true negative, and true discovery rates for a high dimensional data i.e. $p = 100$.

DSS sparse factor model outperformed $\boldsymbol{fanc}$ – based methods in identifying the correct number of factors k as in the true factor model both for low dimensional i.e. $p = 30$, and moderate dimensional i.e. $p = 100$ data.

DSS sparse factor model applied on real continuous (food) (Mumme et. al, 2019) and discrete (fish) (Smith, Duffy and Leathwick, 2013) data produced reasonably sparse i.e. 37% and 29% sparse factor models for food and fish data respectively. The low number of factors is another form of simplicity/sparsity of the final factor loading matrices, allowing for more meaningful interpretations of the factors for both the food and the fish data.

My proposed DSS sparse factor model is different from $\boldsymbol{fanc}$ – based methods since it picks the final factor loading matrix using a range of tuning parameters (ρ, γ) to produce covariance matrices whose expected fit lies within the 90% credible interval of fit. The proposed DSS sparse factor model has an advantage over principal components analysis, or a maximum likelihood factor fit in automatically selecting the non-zero factor loadings.

Using bfa as the source of posterior samples to feed into DSS has a disadvantage of needing to pre-specify the number of factors. However, simulation studies showed that DSS sparse factor model frequently identified the correct numbers of factors, as long as a "large enough" value was provided to bfa (Table: 4.2) that is its desirable property. For future research, DSS sparse factor models may be extended to two groups' cases where important similarities and differences between the factor loading matrices may be modelled.

Chapter - 05

General Conclusions and Recommendations

Significant work has been done in recent years to estimate covariance matrices when the dimension $\boldsymbol{p}$ is greater than the number of observations $\boldsymbol{n}$ i.e. $\boldsymbol{p} > \boldsymbol{n}$. However, there has always been a strong demand to develop methodologies to shrink some of the elements of the inverse covariance matrices and factor loading matrices to exact zeros in the context of multivariate analyses, such as Gaussian graphical models and factor analysis respectively. Sparse models allow for easier interpretations of the inverse covariance matrices (GGMs) and the factor loading matrices (sparse factor models).

In this thesis, my main objectives were to develop methods to sparsify the inverse covariance matrix for one population, to sparsify the differences between two inverse covariance matrices from two populations, and to sparsify the factor loading matrix in the context of factor analysis. I am proposing a sparse, interpretable summary of a complex posterior distribution. I apply various shrinkage techniques to the relevant parameterization of the posterior mean covariance, and the uncertainty represented by the posterior distribution provides the relevant scale to understand when I have shrunk enough versus too much. My proposed Bayesian DSS approaches reduce the computational burden, are subjective but intuitive methods to select the level of sparsity, and are new and original. The proposed Bayesian DSS approaches can easily be extended to other circumstances such as two group case, sparse factor models, and non-normal cases. In this chapter, the findings regarding the three proposed methods are summarized in sections 5.1, 5.2, and 5.3 respectively. The recommendations and future directions are given in section 5.4.

5.1 Decoupled Shrinkage and Selection for One Population

I proposed a DSS method to shrink some elements of an inverse covariance matrix to exact zeros. My proposed DSS method is an extension of DSS applied to the regression setting by Hahn & Carvalho (2015). Sparse estimates of desired parameters from the posterior distribution are produced using my proposed DSS

method. My proposed DSS method generally produced sparser graphs than a range of existing sparsification strategies such as thresholding the partial correlations, credible interval, adaptive graphical $Lasso$, and ratio selection, while maintaining good fit based on the log-likelihood. The DSS credible interval approach produced the sparsest graph for a real metabolites dataset (chapter 2), as compared to the graphs produced by other DSS-based methods. In simulation experiments, my DSS-based methods had better sensitivity and specificity for detecting true edges for cases with high dimension $\boldsymbol{p}$ and large sample size $\boldsymbol{n}$. For low $\boldsymbol{p}$, DSS had comparable performance to the alternative methods. My proposed DSS approach still requires a subjective decision by the user in specifying the credible region corresponding to the "acceptable" loss of fit.

5.2 Decoupled Shrinkage and Selection for Two Populations

In Chapter 3, I extended my DSS approach to detect sparse sets of differences between two inverse covariance matrices from two samples. Application of the DSS method to the two-population case was motivated by Danaher et al.'s (2013) suggestion that the selection of the penalty parameter be guided by "practical considerations", since overly dense models are produced by conventional criterion such as AIC, BIC and cross-validation. The main contribution I made was using the combined posterior covariance mean $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{c}}$ instead of $\boldsymbol{S}$ in fused Joint Graphical $Lasso$ ($\boldsymbol{FGL}$; Danaher et al, 2013), modifying the graphical $Lasso$ penalty to adaptive graphical $Lasso$ penalty, and making some elements identical by modifying the combined posterior mean inverse covariance matrix $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{c}}^{-\mathbf{1}}$. There were no existing Bayesian methods to infer identical elements (Peterson et al, 2015) between precision matrices, and my proposed DSS method filled this gap. Existing frequentist approaches produce progressively sparser sets of differences between the precision matrices, but they do not provide a criterion for selecting the values of tuning parameters. My proposed DSS method fills this gap as well through a data-driven approach to penalty selection. An advantage of my proposed DSS method for one population is its easy extension to the DSS method for two populations by introducing a new prior, without any separate implementation. Detecting a sparse set of differences was challenging for dense matrices for moderate sample sizes. I further demonstrate my DSS method to detect sparse sets of differences between the inverse covariance matrices with the cases and control metabolites datasets (chapter 3).

5.3 Decoupled Shrinkage and Selection Sparse Factor Models

I applied the DSS method to factor analysis, to shrink some of the elements of the factor loading matrix to exact zeros. Based on the true positive rate, true negative rate, and true discovery rate for simulated data, my proposed DSS sparse factor model performed well overall for both continuous and discrete cases, in comparison with the existing $\boldsymbol{fanc}$-based methods (Hirose and Yamamoto, 2015). Moreover, my proposed DSS sparse factor model outperformed the $\boldsymbol{fanc}$-based methods in identifying the correct number of factors (Table: 4.2) as in the true factor loading matrices. Selection of the number of factors was challenging in both my proposed DSS sparse factor model and principal component analysis. However, DSS performed well if the upper limit on the number of factors provided to the algorithm was large enough. The proposed DSS sparse factor model produced meaningful factor loading matrices both for continuous food frequency questionnaire data (Mumme et al, 2019), and discrete fish abundance data (Smith, Duffy and Leathwick, 2013). Pre-specifying the number of factors is a limitation of my proposed DSS sparse factor model.

5.4 Future Work

Possible extensions to my three proposed methods are as follows:

1. My proposed DSS approach for one and two populations could be extended to cases where the variables are observed as a multivariate time series rather than independently and identically distributed observations.

2. For more than two populations, modifications in the joint graphical $Lasso$ algorithm could be done to penalize sum of differences or absolute sum of differences among three or more inverse covariance matrices.

3. The copula approach could be used to make my sparse GGM method suitable for non-normal data.

4. I used $\boldsymbol{fanc}$ in my proposed DSS sparse factor model to shrink some of the elements of the factor loading matrix to exact zeros. Future research could use the $\boldsymbol{OFAL}$ penalty (Hui et al, 2018) to shrink some of the entire columns of the factor loading matrix to exact zeros.

5. My method for non-normal responses could be extended to specify a parametric marginal distribution such as the negative Binomial.

6. The DSS sparse factor model for one population could be extended to develop a DSS sparse factor model for two populations to detect sparse sets of differences between two factor loading matrices.

Appendix "A"

True Factor Loadings Matrix for $p = 30, k = 2$ & $k = 5$.

Variables	$k = 2$		$k = 5$				
	Factor1	Factor2	Factor1	Factor2	Factor3	Factor4	Factor5
X1	0.95	0	0.90	0	0	0	0
X2	0.90	0	0.85	0	0	0	0
X3	0.85	0	0.80	0	0	0	0
X4	0.80	0	0.75	0	0	0	0
X5	0.75	0	0.70	0	0	0	0
X6	0.70	0	0.65	0	0	0	0
X7	0.65	0	0	0.65	0	0	0
X8	0.60	0	0	0.70	0	0	0
X9	0.55	0	0	0.75	0	0	0
X10	0.50	0	0	0.80	0	0	0
X11	0.45	0	0	0.85	0	0	0
X12	0.40	0	0	0.90	0	0	0
X13	0.35	0	0	0	0.60	0	0
X14	0.30	0	0	0	0.55	0	0
X15	0.25	0	0	0	0.50	0	0
X16	0	0.25	0	0	0.45	0	0
X17	0	0.30	0	0	0.40	0	0
X18	0	0.35	0	0	0.35	0	0
X19	0	0.40	0	0	0	0.35	0
X20	0	0.45	0	0	0	0.40	0
X21	0	0.50	0	0	0	0.45	0
X22	0	0.55	0	0	0	0.50	0
X23	0	0.60	0	0	0	0.55	0
X24	0	0.65	0	0	0	0.60	0
X25	0	0.70	0	0	0	0	0.30
X26	0	0.75	0	0	0	0	0.25
X27	0	0.80	0	0	0	0	0.20
X28	0	0.85	0	0	0	0	0.15
X29	0	0.90	0	0	0	0	0.10
X30	0	0.95	0	0	0	0	0.05

True Factor Loadings Matrix for $p = 30, k = 10$

Variables	Factor1	Factor2	Factor3	Factor4	Factor5	Factor6	Factor7	Factor8	Factor9	Factor10
X1	0.90	0	0	0	0	0	0	0	0	0
X2	0.85	0	0	0	0	0	0	0	0	0
X3	0.80	0	0	0	0	0	0	0	0	0
X4	0	0.80	0	0	0	0	0	0	0	0
X5	0	0.85	0	0	0	0	0	0	0	0
X6	0	0.90	0	0	0	0	0	0	0	0
X7	0	0	0.75	0	0	0	0	0	0	0
X8	0	0	0.70	0	0	0	0	0	0	0
X9	0	0	0.65	0	0	0	0	0	0	0
X10	0	0	0	0.65	0	0	0	0	0	0
X11	0	0	0	0.70	0	0	0	0	0	0
X12	0	0	0	0.75	0	0	0	0	0	0
X13	0	0	0	0	0.60	0	0	0	0	0
X14	0	0	0	0	0.55	0	0	0	0	0
X15	0	0	0	0	0.50	0	0	0	0	0
X16	0	0	0	0	0	0.50	0	0	0	0
X17	0	0	0	0	0	0.55	0	0	0	0
X18	0	0	0	0	0	0.60	0	0	0	0
X19	0	0	0	0	0	0	0.45	0	0	0
X20	0	0	0	0	0	0	0.40	0	0	0
X21	0	0	0	0	0	0	0.35	0	0	0
X22	0	0	0	0	0	0	0	0.35	0	0
X23	0	0	0	0	0	0	0	0.40	0	0
X24	0	0	0	0	0	0	0	0.45	0	0
X25	0	0	0	0	0	0	0	0	0.30	0
X26	0	0	0	0	0	0	0	0	0.25	0
X27	0	0	0	0	0	0	0	0	0.20	0
X28	0	0	0	0	0	0	0	0	0	0.20
X29	0	0	0	0	0	0	0	0	0	0.25
X30	0	0	0	0	0	0	0	0	0	0.30

True Factor Loadings Matrix for $p = 100, k = 2$ & $k = 5$.

Variables	$k = 2$		$k = 5$				
	Factor1	Factor2	Factor1	Factor2	Factor3	Factor4	Factor5
X1	0.99	0	0.99	0	0	0	0
X2	0.98	0	0.98	0	0	0	0
X3	0.97	0	0.97	0	0	0	0
X4	0.96	0	0.96	0	0	0	0
X5	0.95	0	0.95	0	0	0	0
X6	0.94	0	0.94	0	0	0	0
X7	0.93	0	0.93	0	0	0	0
X8	0.92	0	0.92	0	0	0	0
X9	0.91	0	0.91	0	0	0	0
X10	0.90	0	0.90	0	0	0	0
X11	0.89	0	0.89	0	0	0	0
X12	0.88	0	0.88	0	0	0	0
X13	0.87	0	0.87	0	0	0	0
X14	0.86	0	0.86	0	0	0	0
X15	0.85	0	0.85	0	0	0	0
X16	0.84	0	0.84	0	0	0	0
X17	0.83	0	0.83	0	0	0	0
X18	0.82	0	0.82	0	0	0	0
X19	0.81	0	0.81	0	0	0	0
X20	0.80	0	0.80	0	0	0	0
X21	0.79	0	0	0.80	0	0	0
X22	0.78	0	0	0.81	0	0	0
X23	0.77	0	0	0.82	0	0	0
X24	0.76	0	0	0.83	0	0	0
X25	0.75	0	0	0.84	0	0	0
X26	0.74	0	0	0.85	0	0	0
X27	0.73	0	0	0.86	0	0	0
X28	0.72	0	0	0.87	0	0	0
X29	0.71	0	0	0.88	0	0	0
X30	0.70	0	0	0.89	0	0	0
X31	0.69	0	0	0.90	0	0	0
X32	0.68	0	0	0.91	0	0	0
X33	0.67	0	0	0.92	0	0	0
X34	0.66	0	0	0.93	0	0	0
X35	0.65	0	0	0.94	0	0	0
X36	0.64	0	0	0.95	0	0	0
X37	0.63	0	0	0.96	0	0	0
X38	0.62	0	0	0.97	0	0	0
X39	0.61	0	0	0.98	0	0	0
X40	0.60	0	0	0.99	0	0	0
X41	0.59	0	0	0	0.79	0	0
X42	0.58	0	0	0	0.78	0	0
X43	0.57	0	0	0	0.77	0	0
X44	0.56	0	0	0	0.76	0	0
X45	0.55	0	0	0	0.75	0	0
X46	0.54	0	0	0	0.74	0	0
X47	0.53	0	0	0	0.73	0	0
X48	0.52	0	0	0	0.72	0	0
X49	0.51	0	0	0	0.71	0	0
X50	0.50	0	0	0	0.70	0	0

X51	0	0.50	0	0	0.69	0	0
X52	0	0.51	0	0	0.68	0	0
X53	0	0.52	0	0	0.67	0	0
X54	0	0.53	0	0	0.66	0	0
X55	0	0.54	0	0	0.65	0	0
X56	0	0.55	0	0	0.64	0	0
X57	0	0.56	0	0	0.63	0	0
X58	0	0.57	0	0	0.62	0	0
X59	0	0.58	0	0	0.61	0	0
X60	0	0.59	0	0	0.60	0	0
X61	0	0.60	0	0	0	0.60	0
X62	0	0.61	0	0	0	0.61	0
X63	0	0.62	0	0	0	0.62	0
X64	0	0.63	0	0	0	0.63	0
X65	0	0.64	0	0	0	0.64	0
X66	0	0.65	0	0	0	0.65	0
X67	0	0.66	0	0	0	0.66	0
X68	0	0.67	0	0	0	0.67	0
X69	0	0.68	0	0	0	0.68	0
X70	0	0.69	0	0	0	0.69	0
X71	0	0.70	0	0	0	0.70	0
X72	0	0.71	0	0	0	0.71	0
X73	0	0.72	0	0	0	0.72	0
X74	0	0.73	0	0	0	0.73	0
X75	0	0.74	0	0	0	0.74	0
X76	0	0.75	0	0	0	0.75	0
X77	0	0.76	0	0	0	0.76	0
X78	0	0.77	0	0	0	0.77	0
X79	0	0.78	0	0	0	0.78	0
X80	0	0.79	0	0	0	0.79	0
X81	0	0.80	0	0	0	0	0.59
X82	0	0.81	0	0	0	0	0.58
X83	0	0.82	0	0	0	0	0.57
X84	0	0.83	0	0	0	0	0.56
X85	0	0.84	0	0	0	0	0.55
X86	0	0.85	0	0	0	0	0.54
X87	0	0.86	0	0	0	0	0.53
X88	0	0.87	0	0	0	0	0.52
X89	0	0.88	0	0	0	0	0.51
X90	0	0.89	0	0	0	0	0.50
X91	0	0.90	0	0	0	0	0.49
X92	0	0.91	0	0	0	0	0.48
X93	0	0.92	0	0	0	0	0.47
X94	0	0.93	0	0	0	0	0.46
X95	0	0.94	0	0	0	0	0.45
X96	0	0.95	0	0	0	0	0.44
X97	0	0.96	0	0	0	0	0.43
X98	0	0.97	0	0	0	0	0.42
X99	0	0.98	0	0	0	0	0.41
X100	0	0.99	0	0	0	0	0.40

True Factor Loadings Matrix for $p = 100, k = 10$

Variables	Factor1	Factor2	Factor3	Factor4	Factor5	Factor6	Factor7	Factor8	Factor9	Factor10
X1	0.99	0	0	0	0	0	0	0	0	0
X2	0.98	0	0	0	0	0	0	0	0	0
X3	0.97	0	0	0	0	0	0	0	0	0
X4	0.96	0	0	0	0	0	0	0	0	0
X5	0.95	0	0	0	0	0	0	0	0	0
X6	0.94	0	0	0	0	0	0	0	0	0
X7	0.93	0	0	0	0	0	0	0	0	0
X8	0.92	0	0	0	0	0	0	0	0	0
X9	0.91	0	0	0	0	0	0	0	0	0
X10	0.90	0	0	0	0	0	0	0	0	0
X11	0	0.90	0	0	0	0	0	0	0	0
X12	0	0.91	0	0	0	0	0	0	0	0
X13	0	0.92	0	0	0	0	0	0	0	0
X14	0	0.93	0	0	0	0	0	0	0	0
X15	0	0.94	0	0	0	0	0	0	0	0
X16	0	0.95	0	0	0	0	0	0	0	0
X17	0	0.96	0	0	0	0	0	0	0	0
X18	0	0.97	0	0	0	0	0	0	0	0
X19	0	0.98	0	0	0	0	0	0	0	0
X20	0	0.99	0	0	0	0	0	0	0	0
X21	0	0	0.89	0	0	0	0	0	0	0
X22	0	0	0.88	0	0	0	0	0	0	0
X23	0	0	0.87	0	0	0	0	0	0	0
X24	0	0	0.86	0	0	0	0	0	0	0
X25	0	0	0.85	0	0	0	0	0	0	0
X26	0	0	0.84	0	0	0	0	0	0	0
X27	0	0	0.83	0	0	0	0	0	0	0
X28	0	0	0.82	0	0	0	0	0	0	0
X29	0	0	0.81	0	0	0	0	0	0	0
X30	0	0	0.80	0	0	0	0	0	0	0
X31	0	0	0	0.80	0	0	0	0	0	0
X32	0	0	0	0.81	0	0	0	0	0	0
X33	0	0	0	0.82	0	0	0	0	0	0
X34	0	0	0	0.83	0	0	0	0	0	0
X35	0	0	0	0.84	0	0	0	0	0	0
X36	0	0	0	0.85	0	0	0	0	0	0
X37	0	0	0	0.86	0	0	0	0	0	0
X38	0	0	0	0.87	0	0	0	0	0	0
X39	0	0	0	0.88	0	0	0	0	0	0
X40	0	0	0	0.89	0	0	0	0	0	0
X41	0	0	0	0	0.79	0	0	0	0	0
X42	0	0	0	0	0.78	0	0	0	0	0
X43	0	0	0	0	0.77	0	0	0	0	0
X44	0	0	0	0	0.76	0	0	0	0	0
X45	0	0	0	0	0.75	0	0	0	0	0
X46	0	0	0	0	0.74	0	0	0	0	0
X47	0	0	0	0	0.73	0	0	0	0	0
X48	0	0	0	0	0.72	0	0	0	0	0
X49	0	0	0	0	0.71	0	0	0	0	0
X50	0	0	0	0	0.70	0	0	0	0	0

X51	0	0	0	0	0	0.70	0	0	0	0
X52	0	0	0	0	0	0.71	0	0	0	0
X53	0	0	0	0	0	0.72	0	0	0	0
X54	0	0	0	0	0	0.73	0	0	0	0
X55	0	0	0	0	0	0.74	0	0	0	0
X56	0	0	0	0	0	0.75	0	0	0	0
X57	0	0	0	0	0	0.76	0	0	0	0
X58	0	0	0	0	0	0.77	0	0	0	0
X59	0	0	0	0	0	0.78	0	0	0	0
X60	0	0	0	0	0	0.79	0	0	0	0
X61	0	0	0	0	0	0	0.69	0	0	0
X62	0	0	0	0	0	0	0.68	0	0	0
X63	0	0	0	0	0	0	0.67	0	0	0
X64	0	0	0	0	0	0	0.66	0	0	0
X65	0	0	0	0	0	0	0.65	0	0	0
X66	0	0	0	0	0	0	0.64	0	0	0
X67	0	0	0	0	0	0	0.63	0	0	0
X68	0	0	0	0	0	0	0.62	0	0	0
X69	0	0	0	0	0	0	0.61	0	0	0
X70	0	0	0	0	0	0	0.60	0	0	0
X71	0	0	0	0	0	0	0	0.60	0	0
X72	0	0	0	0	0	0	0	0.61	0	0
X73	0	0	0	0	0	0	0	0.62	0	0
X74	0	0	0	0	0	0	0	0.63	0	0
X75	0	0	0	0	0	0	0	0.64	0	0
X76	0	0	0	0	0	0	0	0.65	0	0
X77	0	0	0	0	0	0	0	0.66	0	0
X78	0	0	0	0	0	0	0	0.67	0	0
X79	0	0	0	0	0	0	0	0.68	0	0
X80	0	0	0	0	0	0	0	0.69	0	0
X81	0	0	0	0	0	0	0	0	0.59	0
X82	0	0	0	0	0	0	0	0	0.58	0
X83	0	0	0	0	0	0	0	0	0.57	0
X84	0	0	0	0	0	0	0	0	0.56	0
X85	0	0	0	0	0	0	0	0	0.55	0
X86	0	0	0	0	0	0	0	0	0.54	0
X87	0	0	0	0	0	0	0	0	0.53	0
X88	0	0	0	0	0	0	0	0	0.52	0
X89	0	0	0	0	0	0	0	0	0.51	0
X90	0	0	0	0	0	0	0	0	0.50	0
X91	0	0	0	0	0	0	0	0	0	0.50
X92	0	0	0	0	0	0	0	0	0	0.51
X93	0	0	0	0	0	0	0	0	0	0.52
X94	0	0	0	0	0	0	0	0	0	0.53
X95	0	0	0	0	0	0	0	0	0	0.54
X96	0	0	0	0	0	0	0	0	0	0.55
X97	0	0	0	0	0	0	0	0	0	0.56
X98	0	0	0	0	0	0	0	0	0	0.57
X99	0	0	0	0	0	0	0	0	0	0.58
X100	0	0	0	0	0	0	0	0	0	0.59

Appendix “B”

DRC Forms

The signed statement of contribution to doctoral thesis containing publications is attached immediately after the references.

References

Aloraini, A., & Sayed-Mouchaweh, M. (2014, December). Graphical model based approach for fault diagnosis of wind turbines. In 2014 13th International Conference on Machine Learning and Applications (pp. 614-619). IEEE.

Armagan, A., Dunson, D. B., & Lee, J. (2013). Generalized double Pareto shrinkage. *Statistica Sinica*, *23*(1), 119.

Arminger, G., & Muthén, B. O. (1998). A Bayesian approach to nonlinear latent variable models using the Gibbs sampler and the Metropolis-Hastings algorithm. *Psychometrika*, *63*(3), 271-300.

Bayarri, M. J., Berger, J. O., Forte, A., & García-Donato, G. (2012). Criteria for Bayesian model choice with application to variable selection. *The Annals of statistics*, *40*(3), 1550-1577.

Belloni, A., Chen, D., Chernozhukov, V., & Hansen, C. (2012). Sparse models and methods for optimal instruments with an application to eminent domain. *Econometrica*, *80*(6), 2369-2429.

Bernardo, J. M., Bayarri, M. J., Berger, J. O., Dawid, A. P., Heckerman, D., Smith, A., & West, M. (2003). Bayesian factor regression models in the “large p, small n” paradigm. *Bayesian statistics*, *7*, 733-742.

Bhattacharya, A., & Dunson, D. B. (2011). Sparse Bayesian infinite factor models. *Biometrika*, 291-306.

Cai, T. T., Li, H., Liu, W., & Xie, J. (2016). Joint estimation of multiple high-dimensional precision matrices. *Statistica Sinica*, *26*(2), 445.

Cai, T., Liu, W., & Luo, X. (2011). A constrained ℓ 1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, *106*(494), 594-607.

Carvalho, C. M., Chang, J., Lucas, J. E., Nevins, J. R., Wang, Q., & West, M. (2008). High-dimensional sparse factor modeling: applications in gene expression genomics. *Journal of the American Statistical Association*, *103*(484), 1438-1456.

Cheng, Y., & Lenkoski, A. (2012). Hierarchical Gaussian graphical models: Beyond reversible jump. *Electronic Journal of Statistics*, *6*, 2309-2331.

Chib, S., & Greenberg, E. (1998). Analysis of multivariate probit models. *Biometrika*, *85*(2), 347-361.

Chiquet, J., Grandvalet, Y., & Ambroise, C. (2011). Inferring multiple graphical structures. *Statistics and Computing*, *21*(4), 537-553.

Choi, J., Oehlert, G., & Zou, H. (2010). A penalized maximum likelihood approach to sparse factor analysis. *Statistics and its Interface*, *3*(4), 429-436.

Danaher, P., Wang, P., & Witten, D. M. (2014). The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society. Series B, Statistical methodology*, *76*(2), 373.

Dempster, A. P. (1973). Covariance selection. Biometrics, 21(1), 157–175.

Dobra, A., Hans, C., Jones, B., Nevins, J. R., Yao, G., & West, M. (2004). Sparse graphical models for exploring gene expression data. *Journal of Multivariate Analysis*, *90*(1), 196-212.

Efron, B. (1982). *The jackknife, the bootstrap and other resampling plans*. Society for industrial and applied mathematics.

Fan, J., & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, *96*(456), 1348-1360.

Fan, J., Feng, Y., & Wu, Y. (2009). Network exploration via the adaptive LASSO and SCAD penalties. *The annals of applied statistics*, *3*(2), 521.

Fitch, A. M., Jones, M. B., & Massam, H. (2014). The performance of covariance selection methods that consider decomposable models only. *Bayesian Analysis*, *9*(3), 659-684.

Fraley, C., & Raftery, A. E. (2007). Bayesian regularization for normal mixture estimation and model-based clustering. *Journal of classification*, *24*(2), 155-181.

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

Friedman, N. (2004). Inferring cellular networks using probabilistic graphical models. *Science*, *303*(5659), 799-805.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper). *Bayesian analysis*, *1*(3), 515-534.

Ghosh, J., & Dunson, D. B. (2009). Default prior distributions and efficient posterior computation in Bayesian factor analysis. *Journal of Computational and Graphical Statistics*, *18*(2), 306-320.

Giudici, P., & Spelta, A. (2016). Graphical network models for international financial flows. *Journal of Business & Economic Statistics*, *34*(1), 128-138.

Green, P. J., & Thomas, A. (2013). Sampling decomposable graphs using a Markov chain on junction trees. *Biometrika*, *100*(1), 91-110.

Guo, J., Levina, E., Michailidis, G., & Zhu, J. (2011). Joint estimation of multiple graphical models. *Biometrika*, *98*(1), 1-15.

Hahn, P. R., & Carvalho, C. M. (2015). Decoupling shrinkage and selection in Bayesian linear models: a posterior summary perspective. *Journal of the American Statistical Association*, *110*(509), 435-448.

Richard Hahn, P., Carvalho, C. M., & Scott, J. G. (2012). A sparse factor analytic probit model for congressional voting patterns. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *61*(4), 619-635.

Hastie, T., Tibshirani, R. I. (2013). *The elements of statistical learning: data mining, inference, and prediction/by Trevor Hastie, Robert Tibshirani, Jerome Friedman*, second edition, Springer.

Hirose, K., & Yamamoto, M. (2014). Estimation of an oblique structure via penalized likelihood factor analysis. *Computational Statistics & Data Analysis*, *79*, 120-132.

Hirose, K., & Yamamoto, M. (2015). Sparse estimation via nonconcave penalized likelihood in factor analysis model. *Statistics and Computing*, *25*(5), 863-875.

Hirose, K., Yamamoto, M., & Nagata, H. (2016). Fanc: Penalized Likelihood Factor Analysis via Nonconvex Penalty. *R package version*, *2*.

Hosie, G. W., Fukuchi, M., & Kawaguchi, S. (2003). Development of the Southern Ocean continuous plankton recorder survey. *Progress in Oceanography*, *58*(2-4), 263-283.

Huang, A., & Wand, M. P. (2013). Simple marginally noninformative prior distributions for covariance matrices. *Bayesian Analysis*, *8*(2), 439-452.

Hui, F. K., Tanaka, E., & Warton, D. I. (2018). Order selection and sparsity in latent variable models via the ordered factor LASSO. *Biometrics*, *74*(4), 1311-1319.

Jones, B., Carvalho, C., Dobra, A., Hans, C., Carter, C., & West, M. (2005). Experiments in stochastic computation for high-dimensional graphical models. *Statistical Science*, 388-400.

Kim, I. S., Londregan, J., & Ratkovic, M. (2014, November). Voting, Speechmaking, and the Dimensions of Conflict in the US Senate. In *Annual Meeting of the Midwest Political Science Association*.

Kundu, S., Mallick, B. K., & Baladandayuthapani, V. (2019). Efficient Bayesian regularization for graphical model selection. *Bayesian Analysis*, *14*(2), 449-476.

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

Lenkoski, A. (2013). A direct sampler for G-Wishart variates. *Stat*, *2*(1), 119-128.

Li, S., Hsu, L., Peng, J., & Wang, P. (2013). Bootstrap inference for network construction with an application to a breast cancer microarray study. *The annals of applied statistics*, *7*(1), 391.

Li, X., Zhao, T., Yuan, X., & Liu, H. (2015). The flare package for high dimensional linear regression and precision matrix estimation in R. *Journal of machine learning research: JMLR*, *16*, 553.

Liu, H., Lafferty, J., & Wasserman, L. (2009). The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *Journal of Machine Learning Research*, *10*(10).

Liu, H., Roeder, K., & Wasserman, L. (2010). Stability approach to regularization selection (stars) for high dimensional graphical models. In *Advances in neural information processing systems* (pp. 1432-1440).

Liu, H., & Wang, L. (2012). *Tiger: A tuning-insensitive approach for optimally estimating large undirected graphs*. Technical report.

Liu, J. S., & Wu, Y. N. (1999). Parameter expansion for data augmentation. *Journal of the American Statistical Association*, *94*(448), 1264-1274.

Lucas, J., Carvalho, C., Wang, Q., Bild, A., Nevins, J. R., & West, M. (2006). Sparse statistical modelling in gene expression genomics. *Bayesian inference for gene expression and proteomics*, *1*(1).

Meinshausen, N., & Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *The annals of statistics*, *34*(3), 1436-1462.

Mitra, R., Müller, P., & Ji, Y. (2016). Bayesian graphical models for differential pathways. *Bayesian Analysis*, *11*(1), 99-124.

Mohammadi, A., & Wit, E. C. (2015). Bayesian structure learning in sparse Gaussian graphical models. *Bayesian Analysis*, *10*(1), 109-138.

Mukherjee, S., & Speed, T. P. (2008). Network inference using informative priors. *Proceedings of the National Academy of Sciences*, *105*(38), 14313-14318.

Mumme, K., Conlon, C., Hurst, P. V., Jones, M. B., Haskell-Ramsay, C., Stonehouse, W., ... & Beck, K. (2019). Dietary Patterns and Their Nutrients in Older New Zealand Adults. *Multidisciplinary Digital Publishing Institute Proceedings*, *37*(1), 38.

Murray, J. (2016). bfa: Bayesian factor analysis. R package version 0.4.

Murray, J. S., Dunson, D. B., Carin, L., & Lucas, J. E. (2013). Bayesian Gaussian copula factor models for mixed data. *Journal of the American Statistical Association*, *108*(502), 656-665.

Ning, L., & Georgiou, T. T. (2011, December). Sparse factor analysis via likelihood and ℓ 1-regularization. In *2011 50th IEEE Conference on Decision and Control and European Control Conference* (pp. 5188-5192). IEEE.

Peng, J., Wang, P., Zhou, N., & Zhu, J. (2009). Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, *104*(486), 735-746.

Peng, J., Zhu, J., Bergamaschi, A., Han, W., Noh, D. Y., Pollack, J. R., & Wang, P. (2010). Regularized multivariate regression for identifying master predictors with application to integrative genomics study of breast cancer. *The annals of applied statistics*, *4*(1), 53.

Peterson, C. B., Stingo, F. C., and Vannucci, M. (2015). Bayesian inference of multiple Gaussian graphical models. Journal of the American Statistical Association, 110(509), 159–174.

Peterson, C., Vannucci, M., Karakas, C., Choi, W., Ma, L., & Maletić-Savatić, M. (2013). Inferring metabolic networks using the Bayesian adaptive graphical lasso with informative priors. *Statistics and its Interface*, *6*(4), 547.

Quinn, K. M. (2004). Bayesian factor analysis for mixed ordinal and continuous responses. *Political Analysis*, *12*(4), 338-353.

Rencher, A. C. (2003). *Methods of multivariate analysis* (Vol. 492). John Wiley & Sons.

Scott, J. G., & Carvalho, C. M. (2008). Feature-inclusion stochastic search for Gaussian graphical models. *Journal of Computational and Graphical Statistics*, *17*(4), 790-808.

Smith, A. N., Duffy, C. A. J., & Leathwick, J. R. (2013). *Predicting the distribution and relative abundance of fishes on shallow subtidal reefs around New Zealand*. Wellington, New Zealand: Publishing Team, Department of Conservation.

Song, X. Y., & Lee, S. Y. (2001). Bayesian estimation and test for factor analysis model with continuous and polytomous data in several populations. *British Journal of Mathematical and Statistical Psychology*, *54*(2), 237-263.

Stingo, F. C., Chen, Y. A., Vannucci, M., Barrier, M., & Mirkes, P. E. (2010). A Bayesian graphical modeling approach to microRNA regulatory network inference. *The annals of applied statistics*, *4*(4), 2024.

Talluri, R., Baladandayuthapani, V., & Mallick, B. K. (2014). Bayesian sparse graphical models and their mixtures. *Stat*, *3*(1), 109-125.

Tian, D., Gu, Q., & Ma, J. (2016). Identifying gene regulatory network rewiring using latent differential graphical models. *Nucleic acids research*, *44*(17), e140-e140.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267-288.

Wang, H. (2012). Bayesian graphical lasso models and efficient posterior computation. *Bayesian Analysis*, *7*(4), 867-886.

Wang, H., & Li, S. Z. (2012). Efficient Gaussian graphical model determination under G-Wishart prior distributions. *Electronic Journal of Statistics*, *6*, 168-198.

Wilson, R. (1998). Introduction to graph theory, 4th edition, Longman group ltd.

Yuan, M. (2009). Sparse Inverse Covariance Matrix Estimation via Linear Programming. Journal of Machine Learning Research, 11, 2261–2286. [594,595]

Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *68*(1), 49-67.

Zellner, A. (1986). On assessing prior distributions and Bayesian regression analysis with g-prior distributions. *Bayesian inference and decision techniques*.

Zhang, C. H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of statistics*, *38*(2), 894-942.

Zhao, S. D., Cai, T. T., & Li, H. (2014). Direct estimation of differential networks. *Biometrika*, *101*(2), 253-268.

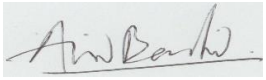
Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association*, *101*(476), 1418-1429.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, *67*(2), 301-320.

MASSEY UNIVERSITY
GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION
DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the candidate and the candidate's Primary Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate's contribution as indicated below in the *Statement of Originality*.

Name of candidate:	**Amir Bashir**
Name/title of Primary Supervisor:	**Dr. Adam Nicholas Howard Smith**
Name of Research Output and full reference:	
Bashir, A; Carvalho, C.M; Hahn, P.R; and Jones, M.B. (2019). "Post-processing posteriors over precision matrices to produce sparse graph estimates". Bayesian Analysis, Volume 14, Number 4 (2019), 1075-1090.	
In which Chapter is the Manuscript /Published work:	**Chapters 02 & 03**
Please indicate:	
☐ The percentage of the manuscript/Published Work that was contributed by the candidate:	**80%**
and	
☐ Describe the contribution that the candidate has made to the Manuscript/Published Work:	
Amir did all the data analyses, created figures and tables, and provided an initial draft of the manuscript.	
For manuscripts intended for publication please indicate target journal:	
Candidate's Signature:	
Date:	**11.05.2020**
Primary Supervisor's Signature:	
Date:	**12.05.2020**

(This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis)