ON THE MODELLING OF HIGH DIMENSIONAL TIME SERIES

CHUNG KAM HIN

PhD

The Hong Kong Polytechnic University

2023

The Hong Kong Polytechnic University

Department of Applied Mathematics

On the Modelling of High Dimensional Time Series

Chung Kam Hin

A thesis submitted in partial fulfilment of the requirements

for the degree of Doctor of Philosophy

September 2022

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____(Signed)

Chung Kam Hin (Name of student)

Dedication

I would like to dedicate my works to my parents and my brother.

Thanks God for his verse in German:

> "Er aber sprach: Was bei den Menschen unmöglich ist, das ist bei Gott möglich." (Lukas 18:27)

The verse in English:

> Jesus replied, "What is impossible with man is possible with God." (Luke 18:27)

I would like to thank my relatives and friends in Hong Kong, Macau, Europe and Canada, especially Family Hung, Family Filzmoser, Mr Avery Ching, Ms Rebecca Chiu, Ms Sharon Leung, Mr Steve Wong, Rev. Karl Flückiger and Rev. Ralph Müller for their encouragement.

Abstract

Graphical models for high dimensional time series data visualize dynamics relationships among variables which however need a huge number of parameters. Some of them, indeed, fail to be estimated in some cases. To tackle these problems, this thesis introduces a new sparse graphical vector, new matrix and new sparse graphical matrix time series models, and studies an estimation problem encountered for modelling the inverse of the high dimensional covariance matrix. Our first model, the sparse graphical vector autoregressive (VAR) model, combines the sparse VAR model and the sparse Gaussian graphical model to visualize causal dynamics and conditional independence among variables. Its autoregressive (AR) coefficient matrix indicates causal dynamics of variables, while a sparse entry of the inverse of the covariance matrix (precision matrix) characterizes conditional independence between variables. Three penalized likelihood estimation methods are used and a final model is selected from all possible sparse graphical VAR models based on the Bayesian Information Criterion (BIC). Under this setting, the model has the optimal sparsity combination of the AR coefficient and precision matrices and its sparsity pattern is more robust than that in existing sparse models, which require the sparsity pre-determined. We develop an algorithm and prove that it is convergent. We also prove that the penalized maximum likelihood estimators of the model are consistent and asymptotically normally distributed. A simulation study shows that our minimax concave penalized (MCP) graphical VAR models always have smaller BIC than the popular LASSO

penalized models and their estimates are unbiased. We apply our models to the Pearl River Delta air pollution data for illustration and compare the results with the existing models. Our MCP model has the minimum BIC. We, then, extend the vector AR model to a matrix AR model with a precision matrix. This model has the AR coefficient matrices in the bilinear form. The left and right coefficient matrices explain the row-wise interaction and the column-wise dependence respectively. This analyzes the structured information under two categorical variables and reduces dramatically the number of parameters. In connection with the sparse Gaussian graphical model, we consider a sparse graphical matrix AR model with the optimal sparsity obtained in the same way as our first model. Two algorithms are proposed and we proved that they are convergent. A simulation study shows satisfactory results. Economic indicator data are fitted for illustration. We compare our model results with their corresponding existing models in the literature. The prediction sum of squared errors of our models are smaller. Finally, the thesis resolves the precision matrix problem encountered in the model estimation. The positive definiteness of the matrix is not easily transformed as constraints in the maximization likelihood estimation. During the estimation, a non-positive definite matrix iterate might be obtained after some iterations and causes numerical errors in the log-likelihood function value calculation. It leads to an estimation failure. We construct two algorithms to keep the precision matrix iterates falling into the positive definite cone in the estimation process. We test our algorithms on the failure cases of the constrained VAR model and our second model. Both are found successful in estimation.

Acknowledgements

I would like to express my deep and sincere gratitude to my supervisor, Prof. Cedric Ka-fai Yiu for his guidance, patience, and continuous support for my study. I would also like to take this chance to show my reverence to Prof. Heung Wong, who was my ex-co-supervisor and retired during my study period. I appreciate that he has given me lots of insights and guidance even after his retirement. Besides this, I participated in a medical study led by Dr. Angela Wang at Queen Mary Hospital. Thanks for her guidance and for giving me valuable exposure to medical research. In addition, I would like to thank Dr. Ting-kei Pong for his valuable discussions and ingenious suggestions on my works and Dr. Alex Kin-yau Wong for his research tips.

I also thank my colleagues and friends in the department for their kindnesses and assistance.

Contents

List of Figures

List of Tables

Notation

$\mathbf{A}^T$	The transpose of matrix $\mathbf{A}$.
a^T	The transpose of vector a.
$\det(\mathbf{M})$	The determinant of a square matrix $\mathbf{M}$
$\mathbf{A}^{-1}$	The inverse of non-singular matrix $\mathbf{A}$.
$\mathbf{I}_n$	An identity matrix of dimension n
$\mathbf{\Sigma} = (\sigma_{ij})$	A covariance matrix
$\mathbf{\Sigma} = (\sigma_{ij})_{i,j=1,\ldots,n}$	A $n \times n$ covariance matrix
$\mathbf{\Theta} = (\theta_{ij})$	A precision matrix
$\mathbf{\Theta} = (\theta_{ij})_{i,j=1,\ldots,n}$	A $n \times n$ precision matrix
$a_{i,j}$	The (i, j) entry of matrix $\mathbf{A}$
$a_{ij,l}$	The (i, j) entry of matrix $\mathbf{A}_l$
$\|\mathbf{A}\|$	Frobenius norm of matrix $\mathbf{A}$
l	A log-likelihood function
f	An objective function
$y^{(k)}$	The k-th iterate of y in an iterative optimization algorithm
$\mathbf{E}_t = (e_{ij,t})_{\substack{i=1,\ldots,m \\ j=1,\ldots,n}}$	A $m \times n$ matrix with time index t.
K	Dimension of vectors used in Chapters 2 and 4

Chapter 1

Introduction

1.1 Background

In the era of big data, data are high-dimensional and complex and models are getting complicated. For example, an environmental study in Hu et al. (2016) aims to identify the core locations of pollutants for management and to improve the air quality in Hong Kong. Four air pollutants over three air pollutant monitoring stations for 1491 days were collected. To investigate the inter-relationship between the air pollutants and between the locations, a vector autoregressive (VAR) model is sufficient, but Hu et al. (2016) applied a more advanced model, a graphical model for multivariate time series to visualize the inter-relationship. The most fundamental and widely used multivariate time series model for multiple time series is the VAR model. Consider a K-dimensional VAR process of lag order p

$$\mathbf{y}_t = \boldsymbol{\nu} + \mathbf{A}_1\mathbf{y}_{t-1} + \ldots + \mathbf{A}_p\mathbf{y}_{t-p} + \mathbf{u}_t,$$

where $\mathbf{y}_t = (y_{1,t}, \ldots, y_{K,t})^T$ is a length K column vector, $\mathbf{A}_l, l = 1, \ldots p$, are $K \times K$ autoregressive coefficient matrices, ν is a length K column vector of intercepts, $\mathbf{u}_t = (u_{1,t}, \ldots, u_{K,t})^T$ is a K-dimensional Gaussian innovation vector with mean $\mathbf{0}$ and a $K \times K$ covariance matrix, $\boldsymbol{\Sigma}_u$ and $t = 1, \ldots, T$. The model assumes that the innovation $\mathbf{u}_t$ is uncorrelated with all past innovations $\mathbf{u}_{t-i}$ and observations $\mathbf{y}_{t-i}$ for

$i = 1, 2, \ldots$, and the process can be viewed as a lagged autoregression of the current value from the past p observations. Data can be modelled by a sufficiently high order p.

The recent development on reducing the model complexity of the VAR(p) process focuses on a sparse VAR(p) model. When the dimension, K, is large and the lag order, p, is high, the model has a very huge number of parameters but it is not necessary because insignificant values in the autoregressive coefficient and covariance matrices can be replaced by zeros in the model. There are two existing approaches for the act in the literature: a traditional approach which determines the sparsity of a sparse VAR model using some statistics, such as partial correlation or spectral coherence; and a sparse modelling approach which determines the sparsity using penalties in the estimation objective functions. Both approaches aim to reduce an unnecessary number of parameters in the VAR model.

The traditional approach for a sparse VAR model requires two stages, sparsity structure identification and constrained optimization for the estimation process. Songsiri et al. (2009b) made use of the spectral density matrix to obtain the conditional independence pairs of variables and apply convex relaxation to estimate sparse autoregressive (AR) coefficient matrices of the VAR model. Davis et al. (2016) determined the sparse coefficients by partial spectral coherence and conducted constrained maximum likelihood estimation. The selected VAR model is then refined to remove spurious non-zero AR coefficients in the second stage. However, the sparsity structure of the inverse of the covariance matrix for undirected graphs is not identified in Songsiri et al. (2009b) and Davis et al. (2016).

Another approach for constructing sparse VAR models is sparse modelling. It is a penalized estimation method and was firstly proposed byTibshirani (1996) for regression. A review of these sparse regression methods can be found in Filzmoser et al. (2012). Hsu et al. (2008), Ren et al. (2013) and Songsiri et al. (2009a) imposed

penalties on the ordinary least squares of residuals to obtain sparse coefficients. However, Song and Bickel (2011) discussed that this linear regression approach ignored the contemporaneous dependence structure in time series.

To visualize the conditional dependence relationship among contemporaneous variables, there exist many graphical models (Pearl et al. (1988), Lauritzen and Wermuth (1989),Whittaker (1990),Wermuth and Lauritzen (1990)). A Gaussian graphical model gives an undirected graph on conditional independence among variables. It consists of a set of edges and a set of vertices, where vertices represent variables and an edge connecting the vertices represents that the corresponding variables are conditional dependent. The graphical model originated from Dempster (1972)'s covariance selection problem. An iterative method is used to find out any conditional independence between variables and the conditional independence between a pair of variables is represented by a zero in the inverse of the covariance matrix (precision matrix). Darroch et al. (1980) linked up the graphical models with log-linear models for discrete data. Detailed introduction to graphical modelling can be found in Edwards (1995) and Lauritzen (1996).

A recent development in graphical models is sparse modelling. Champion et al. (2017) estimated sparse directed acyclic graphs via lasso penalized likelihood. Dahl et al. (2008) suggested an iterative algorithm for covariance selection with nonchordal graphs, where the sparsity can be expressed as conditional independence constraints in the maximization problem of the likelihood function of Gaussian graphical models.

Several researchers have linked the graphical models with time series. Brillinger (1996) discussed how to explore the interrelationship among variables in the time series process by graphs. Dahlhaus (2000) extended the undirected graphical models to explore the condition dependence between variables of a multivariate time series

process. Another VAR model, a structural vector autoregressive model, is considered

$$\mathbf{A}_0\mathbf{y}_t = \mathbf{A}_1\mathbf{y}_{t-1} + \ldots + \mathbf{A}_p\mathbf{y}_{t-p} + \mathbf{e}_t,$$

where K is the dimension, $\mathbf{y}_t = (y_{1,t}, \ldots, y_{K,t})^T$ is a length K column vector, $\mathbf{A}_l, l = 0, 1, \ldots p$, are $K \times K$ autoregressive coefficient matrices, $\mathbf{e}_t = (e_{1,t}, \ldots, e_{K,t})^T$ is a K-dimensional Gaussian innovation vector with mean $\mathbf{0}$ and a $K \times K$ diagonal covariance matrix, $\mathbf{D}_u$ and $t = 1, \ldots, T$. It gives a direct acyclic graph for presenting the causal dynamics between contemporaneous and past variables. When it is transformed into a canonical form, the covariance matrix is converted to a generally structured covariance and may not be sparse. Without the sparse entries in the covariance matrix, the conditional independence between contemporaneous variables could not be identified. The more recent development of graphical time series models can be found in Tunnicliffe Wilson et al. (2015).

A sparse graphical VAR model is needed for illustrating the high-dimensional lagged variable dynamics and the contemporaneous variable relationship. Yuen et al. (2018) proposed a constrained graphical time series model which initially identified the sparsity by partial cross-correlation in the time domain approach and partial spectral coherence the in frequency domain approach. The identified sparsity of the AR coefficient matrices becomes a sparsity constraint for the inverse of the innovation matrix. As a result, both the estimated AR coefficients and the inverse of innovation covariance matrices are sparse. However, it does not allow other sparsity patterns for the parameter matrices in the VAR model and may not generate the best model in some situations. We, therefore, apply the sparse modelling concept to the likelihood function to develop a novel sparse graphical vector autoregressive model. The resulting model is a combined version of the sparse VAR(p) process and the sparse graphical model. The sparse pattern is more robust and flexible. More details will be discussed in Chapter 2.

The VAR(p) model is likely to have an over-parametrization problem and is, indeed, not capable to capture the structured information between two categorical variables. For example, in a study of economic indicators over five countries, data are collected under two classifications. A traditional method is to analyze the data by two autoregressive time series models. And a convention for handling the matrix-variate observations is to treat these multiple observations as a vector. However, the number of parameters is large and the relationship between the indicators among countries is very complicated. Other traditional data analysis methods include dynamic factor analysis (Bai and Ng (2011), Forni et al. (2000); Lam et al. (2011)). Tsai and Tsay (2010) added group constraints in a factor model for the time series. Hallin and Liška (2011) decomposed the time series into blocks and conducted factor analysis.

There is a need on keeping the structural information and further dimension reduction to avoid over-parameterization.

An extension to a matrix autoregressive (MAR) model is a new direction. Chen et al. (2021) proposed a very simple matrix time series model by applying a bilinear regression concept to the autoregressive time series models. The bilinear regression comes from a growth curve model and the bilinear form allows complete interpretability over the original matrix structure. von Rosen (2018) gave the theoretical details. The matrix time series model has a product of three matrices as a similar bilinear form of bilinear regression and an innovation matrix. The middle matrix of the bilinear form is a data matrix at time point $\mathbf{X}_t$. The left matrix is a coefficient matrix which investigates the row-wise interactions, while the right matrix is another coefficient matrix which examines the column-wise dependence. The existing MAR model is yet not adequate for a more complicated data structure. Chen et al. (2021) developed the model up to lag order one. In addition, the MAR model further dramatically reduces the number of parameters by introducing the structured innovation covariance tensor. i.e. The model assumes row-wise and column-wise

innovation relationship covariance matrices. This might be too restrictive in many applications.

A more general matrix time series model is proposed in Chapter 3 to fill the gaps. The matrix time series model needs a more general covariance structure and assumes innovations which may not exhibit completely independent row-wise and column-wise relationships. A graphical presentation will be investigated. In addition, a sparse graphical version is explored in the chapter for model visualization of high-dimensional data.

Our last topic to be covered relates to the estimation of a high-dimensional covariance matrix, which plays, in whatever models, an important role in presenting variable correlations and dependence.

Lots of papers in the literature present a solution to handle the cases when the covariance is not positive definite or it is low-ranked. Bai and Shi (2011) reviewed various methods of reducing the number of parameters for covariance matrix estimation. In the context of asset return, the shrinkage method gives a covariance estimator based on the linear combination of a single index model and the sample covariance. In a factor model, the estimated factor vector and the estimated factor covariance matrix are used to construct a quadratic form and this quadratic form added with a diagonal matrix of the variance of the noise estimates the sample covariance matrix. It can be observed that the above two estimation methods estimate the sample covariance matrix with a lower dimension full rank symmetric positive definite matrix and a diagonal positive matrix. Then the inverse of the covariance matrix could be calculated.

Another similar problem, called the nearest correlation problem, arises in finance. It is one of the matrix nearness problems. A survey on this problem can be found in Higham (1999). Higham (1998) first studied a positive approximant for any arbitrary matrix and this matrix is the nearest symmetric positive semi-definite (psd) matrix.

It is used to modify an indefinite Hessian matrix in the Newton method. Higham (1998) showed that the closest psd matrix was uniquely based on the Frobenius norm, while the uniqueness does not hold based on the shortest 2-norm distance. Higham (2002) examined a particular type of psd matrix, namely the correlation matrix. He computed a symmetric psd matrix for a correlation matrix with zero or negative eigenvalues by using the modified alternating projection method.

Boyd and Lin (2005) studied the least-squares covariance adjustment problem (LSCAP) and it was solved via its dual problem using some standard optimization methods. The LSCAP aims to find the nearest symmetric psd matrix using the least squares sense in the Frobenius norm. In addition, linear equalities and inequalities can be imposed on the problem. Same to the correlation matrix approach, the resultant matrix is a projection on the positive semidefinite cone and is the optimal adjustment. The rank of the optimal adjustment was studied. Qi and Sun (2006) developed a quadratically convergent Newton method to find the nearest correlation matrix and the algorithm is extended to find the nearest covariance matrix.

The estimation of the covariance/precision matrix is problematic, especially in high-dimensional modelling. In a traditional vector model estimation, vectorization of the covariance or the precision matrix destructs its positive definite property. It is difficult to express the positive definite property of these matrices as linear equality or inequality constraints in optimization problems using vector or matrix forms. Computation of $\log(\det(\boldsymbol{\Sigma}))$ gives a numerical error when a non-positive definite covariance iterate, $\boldsymbol{\Sigma}^{(k)}$, is generated in the algorithm. The algorithm stops and the estimation is not successful.

We, in particular, focus on the estimation of the inverse of the innovation covariance matrix, the innovation precision matrix for time series models, because the precision matrix is always estimated for graphical time series models. We avoid implementing positive definite constraints but adopt the nearest correlation matrix

concept to keep the precision matrix iterates in positive definite cones for nonlinear optimization. The details will be discussed in Chapter 4.

1.2 Research Contributions

1. Chapter 2 gives a sparse graphical vector time series model in a new sense. It combines a sparse graphical model and a sparse vector autoregressive (VAR) model. Three penalized likelihood estimation methods are considered and ranges of AR coefficient and precision matrix regularizers are used to estimate all possible sparse graphical VAR models. A final model is selected based on the minimum Bayesian Information Criterion (BIC). As a result, the sparse patterns of autoregressive coefficient and precision matrices are simultaneously selected and are optimal. The final model is much more robust than the traditional sparse time series model, which requires to identify the sparsity, based on partial correlations or other statistics, in the first stage. In addition, the model enables us to plot a mixed graph, which contains a set of nodes and a set of edges for variables and a relationship between variables. The sparse AR matrix gives significant causal dynamics by a directed edge, while a sparse entry of the precision matrix presents a missing undirected edge between variables. The model visualizes significant causal dynamics and conditional dependence among variables. In addition, our model overcomes the difficulty of using a non-convex penalty and the sparse models obtained minimax concave penalized (MCP) estimation method are empirically the best and the model estimates are unbiased. It is much better than the traditional LASSO penalized sparse model because their estimates are biased. An estimation algorithm is developed. We proved that the algorithm is convergent. Consistency and asymptotic normality of the estimator are established.The simulation study

shows satisfactory results. The Pearl River Delta air pollution data is fitted with our models for illustration and it is compared with existing models in the literature. The MCP sparse model has the lowest BIC and therefore, it is the best.

2. Chapter 3 extends the VAR model to a matrix AR (MAR) model, which caters for fitting data under two categorical variables. The proposed MAR model has AR matrices in bilinear forms and a generally structured precision matrix. The model allows us to extract structured information under two classifications and gives a graphical representation of conditional dependence among variables. In addition, it reduces the number of parameters used dramatically. To further reduce the number of parameters, the same penalized estimation methodology as in Chapter 2 is applied to obtain a new sparse graphical MAR model. For simplicity, the popular penalty, LASSO, is used for estimating all possible sparse graphical MAR models. Again, BIC is used to select the model with optimal sparsity combination between AR coefficient and precision matrices. Two corresponding algorithms are developed and we proved that they are convergent. A simulation study shows satisfactory results. In addition, the proposed MAR model tackles well in more situations than the existing MAR model. An economic indicator example is used to illustrate the proposed MAR and the proposed sparse MAR models. Comparison is made with the existing models in the literature. Both proposed models have smaller prediction sums of squared errors. Therefore, they are better.

3. Chapter 4 resolves the problem arising from the positive definite property of the covariance $\mathbf{\Sigma}$ or precision $\mathbf{\Theta}$ matrices estimation. This property is not easily transformed as equality and inequality constraints in the maximization of a log-likelihood function. During the estimation process, there might exist an it-

erate of covariance or precision matrix being non-positive definite or very close to zero, calculation of log-likelihood functions involve the term $\log(\det(\mathbf{\Sigma})$ or $\log(\det(\mathbf{\Theta})$, which causes numerical errors. We construct two algorithms for a vector time series and a matrix time series. Both algorithms keep every covariance/precision matrix iterate in the positive definite cone. We have extracted some estimation failure cases from a constrained graphical VAR model and a MAR model in Chapter 3 for testing. Both algorithms are found successful in estimation. In addition, it has been discussed that our algorithms are descent and convergent under some regularity conditions.

As a result, the following four papers are being prepared in conjunction with this thesis:

1. Dorothy Kam-hin Chung, Cedric Ka-fai Yiu and Heung Wong, "On sparse graphical modelling in time series". Comments were received. The paper has been revised and will be re-submitted to "Journal of Computational Statistics".

2. Dorothy Kam-hin Chung, Cedric Ka-fai Yiu and Heung Wong, "Modelling graphical matrix time series". To be submitted.

3. Dorothy Kam-hin Chung, Cedric Ka-fai Yiu and Heung Wong,"A new method to handle positive definiteness of covariance/precision matrices in constrained vector model estimation". To be submitted.

4. Dorothy Kam-hin Chung, Cedric Ka-fai Yiu and Heung Wong,"Handling positive definiteness of covariance/precision matrices in matrix model estimation". To be submitted.

Parts of the results of Chapter 2 was presented in a talk of conference:

1. Dorothy Kam-hin Chung and Cedric Ka-fai Yiu, "Sparse graphical time seres models", NACA-ICOTA 2019, Hakodate, Japan, August 26-31, 2019.

1.3 Thesis Outline

Here is the outline of the thesis.

1. Chapter 2 reviews the graphical models, the sparse vector autoregressive models and introduces a new sparse model, which combines the features of these two models. This new model is estimated by penalized log-likelihood estimation. Three different penalties are proposed. An algorithm is set up and it is proved that it is convergent.

2. Chapter 3 extends the bilinear regression to matrix auto-regressive time series model using structured covariance tensor to a general version of MAR(p) model with a free structured covariance and a higher lag order. To cater for the need for parameter reduction, a sparse version of the MAR(p) model using a precision matrix is proposed. Together with the sparse precision matrix, the graphical structure of the conditional dependence between variables would be visualized.

3. Chapter 4 explores the problems encountered during covariance/precision matrix estimation in time series when the positive definiteness property is not easily imposed as constraints in the estimation. Two new algorithms are proposed, where one is for vector time series while another one is for matrix time series. They both remedy the problematic situation by replacing the non-positive definite covariance/precision matrix with their closest covariance/precision matrix. The convergence of algorithms is discussed.

4. Chapter 5 gives the conclusions and explores the future works.

Chapter 2

Vector Autoregressive Graphical Time Series Models

High dimensional time series data are always available in finance, economics, environmental science and many other areas. The model complexity is getting higher and higher. To make the models better and easier for understanding, visualization and sparse model estimation techniques are important.

A mixed graph can be used to visualize causal dynamics and conditional independence between time series variables. It contains a set of edges and a set of vertices. Each vertex represents a variable. Each directed edge represents influence in the past from one variable to another variable, while each undirected edge represents conditional dependence between variables. A missing directed edge between vertices indicates a null value in the autoregressive coefficient matrix for the corresponding pair of variables, while a missing undirected edge between vertices indicates a null value in the precision matrix between the corresponding pair of variables.

Estimating an inverse of covariance matrix (precision matrix) with null values is found in Dempster (1972). It aims to use a minimum number of parameters to estimate the precision matrix based on normality assumption and is converted as a mixed graph for viewing the conditional independence between variables. It is developed as a Gaussian graphical model. Detailed introduction of graphical modelling

can be found in Edwards (1995) and Lauritzen (1996).

In the recent development of a sparse version of the Gaussian graphical model, Dahl et al. (2008) made use of the projection of a sample covariance matrix onto the set of symmetric matrices, having the same sparsity structure as the constraints for maximum likelihood estimation, to convert the constrained optimization problem into an unconstrained optimization problem. Banerjee and d'Aspremont (2007) developed a block coordinate algorithm to estimate an inverse of covariance matrix from the LASSO penalized likelihood function. Friedman et al. (2008) developed a graphical LASSO algorithm for the model.

In the recent development of a sparse time series model. Davis et al. (2016) proposed a sparse VAR model based on two-stage estimation. The first stage identifies the insignificant conditional dependence between pairs of variables by partial spectral coherence. These insignificant relations, represented by sparsity in temporal variables, are converted as zero AR coefficients constraints for maximum likelihood estimation for models at different lag orders. Models corresponding to different lag orders are constructed and Bayesian information criterion (BIC) is used to choose the 'best' model. This selected model is then refined to remove spurious non-zero small AR coefficients based on t-statistics at the second stage. However, the precision matrix of this sparse model is likely to contain a few very small insignificant values and may not achieve the most parsimonious form of conditional independence among variables.

Yuen et al. (2018) extended the sparse VAR model to a constrained graphical sparse VAR (CGsVAR) model by adding the conditional independence feature among variables in the precision matrix and their model generates a mixed graph. Again, the identification of the sparse structure of both AR coefficient matrices and the inverse of the innovation covariance matrix is required. Zero constraints for likelihood estimation represent conditional independence, which is obtained from a partial

correlation graph for the time domain approach and partial spectral coherence for the frequency approach. Then the CGsVAR model is estimated by an iterative algorithm, incorporating alternating constrained optimization solved by alternate convex search. Therefore, the both AR and precision matrices' sparse structure of the model is restricted by conditional independence constraints and the AR coefficient matrices may not have the best sparsity pattern in practice.

The application of a graph model onto a time series model is suggested by Brillinger (1996). He discussed how to explore the interrelationship between variables in the time series process by graphs. Dahlhaus (2000) extended the undirected graphical models to explore the conditional dependence between variables of a multivariate time series process. The recent development of graphical time series models can be found in Tunnicliffe Wilson et al. (2015).

In this chapter, we consider combining a sparse Gaussian graphical model with a sparse VAR model. In literature, this sparse model requires two-stage estimation. The novelty of our sparse model is that we select one optimal sparse model from all possible sparse combinations of sparse autoregressive coefficient and sparse precision matrices. Minimum BIC is used as a selection criterion. Penalized estimation is conducted with the popular LASSO penalty and two other unbiased penalties, the smoothly clipped absolute deviation (SCAD) penalty and the minimax concave penalty (MCP). In fact, they are oracle but non-convex in nature. We overcome the challenge to use the non-convex penalty and developed an algorithm for the proposed model. We proved that the algorithm is convergent. Simulated data shows a promising result using the MCP estimation. An environmental application in Pearl River Delta is used to demonstrate our model and the result is compared with the existing sparse graphical model. Our model has minimum BIC and this confirms that our proposed sparse graphical model is useful in time series.

This chapter is organized as follows. Section 2.1 discusses how a sparse graphical

VAR model is built via a penalized likelihood estimation problem. A simple and new iterative alternating algorithm is developed in Section 2.1. A simulation study is performed for various VAR models in Section 2.3. Section 2.4 illustrates our LASSO, SCAD and MCP sparse models with the Pearl River Delta air pollution data and compares the results with the existing sparse model methods. Section 2.5 gives the conclusion.

2.1 Sparse Graphical Time Series Model

We aim to produce an optimal sparse time series model so that its presentation in a mixed graph is the simplest. The directed and undirected edges represent temporal structure and the conditional dependence between variables.

Finding null values in the precision matrix is the covariance selection problem (Dempster (1972)). A covariance matrix might be low-ranked and Dempster (1972) aimed to use a minimum number of parameters by setting null values to the elements between the independent component pairs of the precision matrix iteratively. A sequence of hypothesis testings on finding null elements is conducted on the sample data with normal assumption. This method was then developed as a graphical model.

Mathematically, the covariance selection problem can be expressed in the framework of maximum likelihood estimation. Consider a K-dimensional random variable $\mathbf{X} \sim N(\mathbf{0}, \mathbf{\Sigma})$. The precision matrix $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$ satisfies

$$\begin{aligned} \max \quad & \log\det(\mathbf{\Theta}) - \mathrm{tr}(\mathbf{S\Theta}) \\ \text{subject to} \quad & \begin{cases} \mathbf{\Theta}_{ij} = 0, \ \ (i,j) \in \Omega, \\ \mathbf{\Theta} \succ 0, \end{cases} \end{aligned} \tag{2.1}$$

where $\mathbf{S}$ is the sample covariance matrix, Ω is an ultimate set of conditionally independent node pairs in the final step and the last condition on the $\mathbf{\Theta}$ matrix guarantees the matrix is positive definite.

When the method is applied to a time series model, the conditional independence between the contemporaneous variables is investigated. The null values in the precision matrix from the innovations terms are found after fitting the data with a time series model. A null value between two nodes (variables) represent missing edges between the two variables in a conditional independence graph (CIG). On the contrary, a non-null value between two variables indicates they are conditionally dependent. Thus, the two variables are linked with an edge in the CIG. The contemporaneous interrelationship among variables in a time series model would be visualized.

In the development of large scaled high dimensional Gaussian graphical model problems, parsimonious models with a minimum number of parameters are aimed at data analysis and this induces the sparse model estimation in the covariance and precision matrix. Dahl et al. (2008) made use of the projection of a sample covariance matrix onto the set of a symmetric matrix having the same sparsity structure as the constraints of the maximum likelihood estimation to convert the constrained optimization problem into an unconstrained optimization problem. Banerjee and d'Aspremont (2007) and Friedman et al. (2008) applied the LASSO penalty to likelihood function onto inverse covariance matrix estimation for a sparse graph. The penalized log-likelihood is

$$\begin{aligned} \max \quad & \log\det(\boldsymbol{\Theta}) - \mathrm{tr}(\mathbf{S}\boldsymbol{\Theta}) - \lambda\|\boldsymbol{\Theta}\|_1 \\ \text{subject to} \quad & \boldsymbol{\Theta} \succ 0, \end{aligned} \tag{2.2}$$

where $\mathbf{S}$ is the sample covariance matrix, λ is a non-negative regularization parameter and $\|\cdot\|_1$ is the l_1 norm.

The commonly used vector time series model is the vector autoregressive (VAR) model and it can be visualized the causal relationship among the lagged variables. Existence of a causal relationship would be indicated by a directed edge in a mixed graph. The sparse VAR models have not yet been well estimated by a sophisticated

penalized approach and require sparsity structure identification in AR matrices before model estimation. Song and Bickel (2011) attempted to construct a sparse VAR model by LASSO penalized regression methods via ordinary least squares. However, such an approach does not exploit the covariance matrix and ignores the temporal and contemporaneous dependence structure of time series data. Another sparse VAR model, proposed by Davis et al. (2016) , is a two-stage sparse VAR model. The first stage determines the sparsity structure of the AR matrices by partial spectral coherence and fits several VAR models up to a pre-specified lag order. A final model is then selected by minimum BIC. In the second stage, the model is fine-tuned by removing small AR coefficients using t-tests. However, such a sparse model does not achieve the most parsimonious form of precision matrix.

2.1.1 Problem formulation

Consider a K-dimensional vector autorgressive model, VAR(p):

$$\mathbf{y}_t = \boldsymbol{\nu} + \mathbf{A}_1\mathbf{y}_{t-1} + \ldots + \mathbf{A}_p\mathbf{y}_{t-p} + \mathbf{u}_t,$$

where p is a pre-determined lag order, $\mathbf{y}_t = (y_{1,t}, \ldots, y_{K,t})^T$ is a length K column observation vector, $\mathbf{A}_l, l = 1, \ldots p$, are $K \times K$ autoregressive (AR) coefficient matrices, ν is a length K column vector of intercepts, $u_t = (u_{1,t}, \ldots, u_{K,t})^T$ is a K-dimensional Gaussian innovation vector with mean $\mathbf{0}$ and a $K \times K$ positive definite covariance matrix, $\boldsymbol{\Sigma}_u$ and $t = 1, \ldots, T$. We estimate the innovation precision matrix $\boldsymbol{\Theta} = \boldsymbol{\Sigma}_u^{-1}$, because the innovation precision matrix indicates directly the conditional dependence between the variables and the partial correlation coefficients can be easily calculated by the $\rho_{ij} = -\frac{\theta_{ij}}{\sqrt{\theta_{ii}\theta_{jj}}}$ for $i, j = 1, \ldots, K$, where θ_{ij} is the (i, j)-th entry of $\boldsymbol{\Theta}$. We then rewrite the model as multivariate regression model, $\mathbf{Y} = \mathbf{BZ} + \mathbf{U}$ and the likelihood function becomes

$$l(\mathbf{B}, \boldsymbol{\Theta}) = -\frac{KT}{2}\log 2\pi + \frac{T}{2}\log\det\boldsymbol{\Theta} - \frac{1}{2}\text{trace}\left((\mathbf{Y} - \mathbf{BZ})^T\boldsymbol{\Theta}(\mathbf{Y} - \mathbf{BZ})\right) \quad (2.3)$$

where $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_T)$, $\mathbf{B} = (\boldsymbol{\nu}, \mathbf{A}_1, \ldots, \mathbf{A}_p)$ is a $K \times (Kp+1)$ matrix, $\mathbf{Z} = (\mathbf{z}_0, \ldots, \mathbf{z}_{T-1})$ is a $(Kp+1) \times T$ matrix with $\mathbf{z}_t = (1, \mathbf{y}_t^T, \ldots, \mathbf{y}_{t-p+1}^T)^T$, a length $(Kp+1)$ column vector, $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_T)$. For simplicity, we assume $\boldsymbol{\nu}$ is a zero vector and drop the $\boldsymbol{\nu}$ vector in the $\mathbf{B}$ matrix and the first row vector of ones in $\mathbf{Z}$ for discussion afterwards. The maximum likelihood estimators (MLE) of $\mathbf{B}$ and $\boldsymbol{\Theta}$ are

$$\hat{\mathbf{B}} = \mathbf{Y}\mathbf{Z}^T(\mathbf{Z}\mathbf{Z}^T)^{-1} \quad \text{and} \quad \hat{\boldsymbol{\Theta}} = T\left((\mathbf{Y} - \hat{\mathbf{B}}\mathbf{Z})(\mathbf{Y} - \hat{\mathbf{B}}\mathbf{Z})^T\right)^{-1}. \tag{2.4}$$

Chapter 3 of Lütkepohl (2005) gave the estimation of the VAR process in detail.

A traditional way for sparse estimation is to apply a penalty on the residual sum of squares, as in the regression approach, but this method ignores contemporaneous dependence structure in time series. Basu and Michailidis (2015) provided empirical results that estimation via the log-likelihood with penalty gave better results than the penalized ordinary least squares method. Friedman et al. (2008) applied the LASSO penalty onto the log-likelihood function for the inverse covariance matrix estimation to create a sparse graph.

We conduct penalized likelihood estimation and impose a penalty on the coefficients and innovation precision matrices to the VAR(p) likelihood function (2.3) and our optimization problem becomes:

$$\arg\min_{\mathbf{B},\boldsymbol{\Theta}} F(\mathbf{B}, \boldsymbol{\Theta}) \quad := \quad -l(\mathbf{B}, \boldsymbol{\Theta}) + T\sum_{i,j} p_{\lambda_{\mathbf{B}}}(|b_{ij}|) + T\sum_{i \neq j} p_{\lambda_{\boldsymbol{\Theta}}}(|\theta_{ij}|), \tag{2.5}$$

where $p_{\lambda_{\mathbf{B}}}(.)$ and $p_{\lambda_{\boldsymbol{\Theta}}}(.)$ are penalty functions, with $\lambda_{\mathbf{B}}$ and $\lambda_{\boldsymbol{\Theta}}$ being regularization parameters, for $K \times Kp$ coefficient matrix $\mathbf{B} = (b_{ij})$ and $K \times K$ innovation precision matrix, $\boldsymbol{\Theta} = (\theta_{ij})$, n is sample size and p is the number of lag. Note that penalty is applied to all elements of coefficient matrix $\mathbf{B}$ and all off-diagonal elements of the precision matrix $\boldsymbol{\Theta}$.

Under this setting, we can obtain all possible sparse models with different sparse combinations of coefficient and precision matrices. Minimum BIC is used to select the model to optimal sparseness.

2.2 Estimation

We propose an iterative algorithm for estimating the sparse graphical VAR model by the penalized likelihood. LASSO penalty is widely used in Gaussian graphical and VAR models (Meinshausen and Bühlmann (2006); Friedman et al. (2007); Song and Bickel (2011)) because it keeps the convexity nature of the penalized estimation problem and is robust. However, LASSO penalized estimator is not unbiased (Fan and Li (2001)).

2.2.1 Penalties

A good penalty is a function, which should result in an estimator with three properties. The first important property is "unbiasedness". i.e. The resulting estimator is nearly unbiased when the true unknown parameter is large to avoid unnecessary modelling bias. The second property is "sparsity". The resulting estimator sets a thresholding rule, which automatically sets small estimated coefficients to zero to reduce model complexity. The last one is "continuity". The resulting estimator is continuous to avoid instability in model prediction. A penalty possessing these three properties is oracle.

Three penalties are considered. The first one is the least absolute shrinkage and selection operator (LASSO) technique, proposed by Tibshirani (1996). It is applied to Gaussian log-likelihood function to estimate undirected graphs. It is equivalent to an l_1 norm penalty and is widely used. Banerjee et al. (2008) and Friedman et al. (2008) applied the Lasso penalty. Champion et al. (2017) estimated sparse directed acyclic graphs via lasso penalized likelihood. LASSO is popular

and can be used as a benchmark for penalized estimation. The second penalty is the smoothly clipped absolute deviation (SCAD) penalty. Fan and Li (2001) claimed that the SCAD penalty was better at selecting significant variables than the LASSO method and its estimator was unbiased. Zhang (2010a) proposed the minimax concave penalty (MCP) because it is a fast, continuous, nearly unbiased and accurate penalized variable selection method. Indeed, the SCAD and MCP are oracle penalties. The sparse model estimated with the LASSO penalty is used as a benchmark. The formulas of these three penalties are given in Table 2.1 and the penalty functions are given in Figure 2.1.

Table 2.1: LASSO, SCAD and MCP regularizers, which are continuous but non-smooth and non-convex, are expressed as a difference between two convex functions, i.e. $p_\lambda(\mathbf{w}) = p_{\lambda,1}(\mathbf{w}) - p_{\lambda,2}(\mathbf{w})$. Here, λ is the regularization parameter, w_i is the i-th element of $\mathbf{w}$ and $[x]_+ = max(0, x)$.

Penalty	$p_\lambda(w_i)$	$p_{\lambda,2}(w_i)$
LASSO	$\lambda\|w_i\|$	0
SCAD with parameter ϕ	$\lambda\int_0^{\|w_i\|}\min\left(1,\frac{[\phi\lambda-x]_+}{(\phi-1)\lambda}\right)dx \ (\phi>2)$ $=\begin{cases}\lambda\|w_i\| & \text{if } \|w_i\|\le\lambda,\\ \frac{-w_i^2+2\phi\lambda\|w_i\|-\lambda^2}{2(\phi-1)} & \text{if } \lambda<\|w_i\|\le\phi\lambda,\\ \frac{(\phi+1)\lambda^2}{2} & \text{if } \|w_i\|>\phi\lambda.\end{cases}$	$\lambda\int_0^{\|w_i\|}\frac{[\min(\phi\lambda,x)-\lambda]_+}{(\phi-1)\lambda}dx \ (\phi>2)$ $=\begin{cases}0 & \text{if } \|w_i\|\le\lambda,\\ \frac{w_i^2-2\phi\lambda\|w_i\|+\lambda^2}{2(\phi-1)} & \text{if } \lambda<\|w_i\|\le\phi\lambda,\\ \lambda\|w_i\|-\frac{(\phi+1)\lambda^2}{2} & \text{if } \|w_i\|>\phi\lambda.\end{cases}$
MCP with parameter ϕ	$\lambda\int_0^{\|w_i\|}\left[1-\frac{x}{\phi\lambda}\right]_+dx \ (\phi>0)$ $=\begin{cases}\lambda\|w_i\|-\frac{w_i^2}{2\phi} & \text{if } \|w_i\|\le\phi\lambda,\\ \frac{\phi\lambda^2}{2} & \text{if } \|w_i\|>\phi\lambda.\end{cases}$	$\lambda\int_0^{\|w_i\|}\min\left(1,\frac{x}{\phi\lambda}\right)dx \ (\phi>0)$ $=\begin{cases}\frac{w_i^2}{2\phi} & \text{if } \|w_i\|\le\phi\lambda,\\ \frac{\lambda\|w_i\|-\phi\lambda^2}{2} & \text{if } \|w_i\|>\phi\lambda.\end{cases}$

Remark: $p_{\lambda,1}(w_i) = \lambda|w_i|$ for LASSO, SCAD and MCP.

We expect that the MCP performs the best among the three penalties, its formula

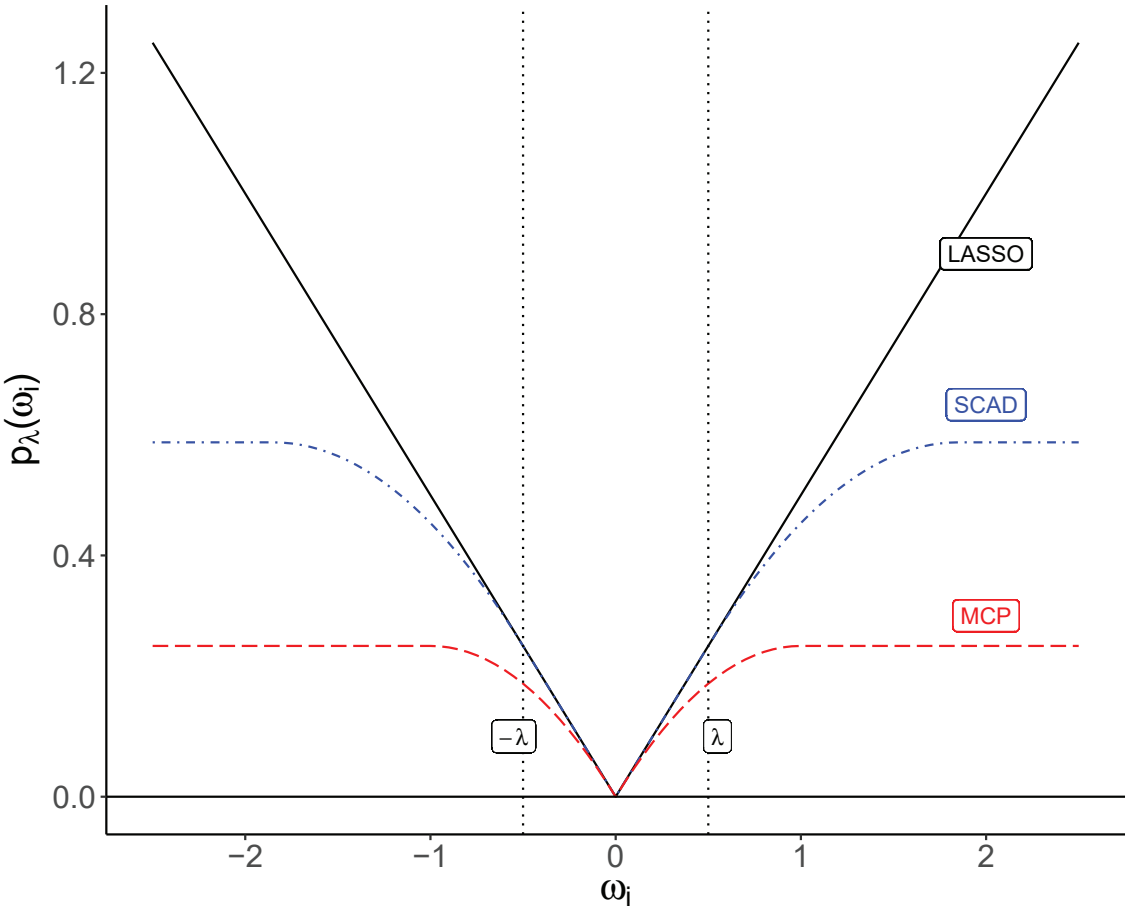


Figure 2.1: LASSO, SCAD and MCP penalties

is examined as follows:

$$p_{SCAD,\lambda}(\omega_i) = \begin{cases} p_{LASSO,\lambda} & \text{if } |\omega_i| \leq \lambda, \\ p_{LASSO,\lambda} - \frac{(\lambda - |\omega_i|)^2}{2(\phi-1)} & \text{if } \lambda < |\omega_i| \leq \phi\lambda, \quad \text{and} \\ \frac{(\phi+1)\lambda^2}{2} & \text{if } |\omega_i| > \phi\lambda, \end{cases}$$

$$p_{MCP,\lambda}(\omega_i) = \begin{cases} p_{LASSO,\lambda} - \frac{\omega_i^2}{2\phi} & \text{if } |\omega_i| \leq \phi\lambda, \\ \frac{\phi\lambda^2}{2} & \text{if } |\omega_i| > \phi\lambda, \end{cases},$$

where the LASSO penalty is $p_{LASSO,\lambda} = \lambda|\omega_i|$ for all ω_i.

We discuss theoretically the performance of the TPR based on the formula in Table 2.1 and Figure 2.1. The TPR measures the true positive rates, i.e. the proportion of non-zero elements to be estimated as non-zero. When $\omega_i > \phi\lambda$, the MCP

method adds a constant weight to ω_i for estimation. This increases the chances of non-zero elements being estimated as non-zero. However, the LASSO method exhibits a penalty being proportional to ω_i, i.e. the larger the absolute value of ω_i, the larger the penalty. This action is likely to vanish the estimates. Then the MCP method keeps more non-zero elements than the LASSO method. This explains why the MCP method has a better TPR than the LASSO method. The SCAD penalty has a bigger weight, compared with MCP. Based on a similar argument as above, the SCAD penalty performs in the middle position.

We then discuss theoretically the performance of the TNR using the LASSO and MCP methods. When $0 < \omega_i \leq \phi\lambda$, $p_{MCP,\lambda}(\omega_i) < p_{LASSO,\lambda}$. Therefore, the MCP method is more sensitive to the small true values and it vanishes the estimates of the small values, while the LASSO method is likely to overweight the estimates of small true values. Therefore, the MCP method tends to have more zero estimates for zero true values than the LASSO method. Under this range of ω_i, the SCAD penalty has two parts (refer to Table 2.1). The first part is the same as the LASSO penalty, while the second part is a smooth concave curve, $p_{LASSO,\lambda} - \frac{(\lambda-|\omega_i|)^2}{2(\phi-1)}$. It connects $\lambda|\omega_i|$ and $\frac{(\phi+1)\lambda^2}{2}$. The last part is bigger than the last part of the MCP penalty. Therefore, the SCAD penalty performs in the middle position using a similar argument as above.

It is expected that the MCP method would give a better sparsity performance and hence gives a more accurate performance, in terms of bias, variance and MSE. As discussed above, the TNR and the TPR of the MCP method are better than that of the LASSO method. This indicates that the MCP method produces fewer estimation errors for zero and non-zero true values and has a smaller bias and a smaller MSE.

2.2.2 Proposed algorithm

The joint likelihood function (2.5) of $(\mathbf{B}, \boldsymbol{\Theta})$ is not convex. Many researchers developed a different alternating algorithm to solve the multivariate regression problem with LASSO penalty, such as multivariate regression with covariance estimation of Rothman et al. (2010) and Lee and Liu (2012)'s coordinate-descent algorithm for Gaussian multivariate regression.

Sofer et al. (2014) introduced a so-called "two-stage procedure" for the penalized likelihood estimation with other penalties for multivariate regression. In particular, Yuen et al. (2018) proved that this optimization problem was 'biconcave'. This indicates that the alternating algorithm can solve the estimation problem well. Therefore, we adopt their similar approach and split the estimation into two alternating steps: coefficient matrix, $\mathbf{B}$, and precision matrix, $\boldsymbol{\Theta}$, estimation.

The SCAD and MCP penalized likelihood functions for estimating $\mathbf{B}$ and $\boldsymbol{\Theta}$ are non-convex and non-smooth. Indeed, they are continuously differentiable with Lipschitz continuous gradient and bounded below. A common approach to solve these functions is to use the Multi-Stage convex relaxation or difference of convex functions programming (Zhang (2010b)) which relaxes the original non-convex problem to a sequence of convex problems. Thus this requires high computation costs for large-scale problems. Gong et al. (2013) proposed a more efficient algorithm, called the general iterative shrinkage thresholding (GIST) algorithm, which iteratively solves a proximal operator problem with closed-form solutions for a large class of non-convex penalties, including LASSO, SCAD and MCP. The line search step size in the algorithm is initialized with Barzilai-Borwein (BB) rule (Barzilai and Borwein (1988)) at each outer iteration, which greatly accelerates the convergence speed. Indeed, a nonmonotone line search can be used to further speed up the convergence speed. Since our estimation subproblems fulfil the assumptions of using the GIST

algorithm, we use it for solving $\mathbf{B}$ and $\mathbf{\Theta}$ in our unconstrained optimization part.

The proposed algorithm for solving $(\mathbf{B}, \mathbf{\Theta})$ is the following:

1: Set $i_{\mathbf{B}} = i_{\mathbf{\Theta}} = 1$ and the regularization parameter pair $(\lambda_{\mathbf{B}}, \lambda_{\mathbf{\Theta}})$ to (0.01,0.01).

2: *(Initialization of parameters* $\mathbf{B}$ *and* $\mathbf{\Theta}$*)* For each $(\lambda_{\mathbf{B}}, \lambda_{\mathbf{\Theta}})$ set of given values, set the outer iteration counter, m, to 1. When $i_{\mathbf{B}} = i_{\mathbf{\Theta}} = 1$, set the initial values of $\mathbf{B}$ and $\mathbf{\Theta}$ as $\mathbf{B}^{(0)}$ and $\mathbf{\Theta}^{(0)}$, which are the maximum likelihood estimates of (2.3), otherwise use a warm start by setting the initial values as previous $\mathbf{B}_{(i_{\mathbf{B}}-1, i_{\mathbf{\Theta}})}$ and $\mathbf{\Theta}_{(i_{\mathbf{B}}-1, i_{\mathbf{\Theta}})}$, when $i_{\mathbf{B}} > 1$ and $i_{\mathbf{\Theta}} = 1$; and $\mathbf{B}_{(i_{\mathbf{B}}, i_{\mathbf{\Theta}}-1)}$ and $\mathbf{\Theta}_{(i_{\mathbf{B}}, i_{\mathbf{\Theta}}-1)}$, when $i_{\mathbf{B}} = 1, \ldots, 100$ and $i_{\mathbf{\Theta}} > 1$.

3: *(Block Coordinate Gradient Descent Algorithm)* Given $\mathbf{\Theta}^{(m-1)}$, solve $\mathbf{B}^{(m)}$ from the following by an algorithm given in next section.

$$\mathbf{B}^{(m)} = \arg\min_{\mathbf{B}} -l(\mathbf{B}, \mathbf{\Theta}^{(m-1)}) + T \sum_{i,j} p_{\lambda_{\mathbf{B}}}(|b_{ij}|) \tag{2.6}$$

4: Given $\mathbf{B}^{(m-1)}$, solve $\mathbf{\Theta}^{(m)}$ from the following by the algorithm given in next section.

$$\mathbf{\Theta}^{(m)} = \arg\min_{\mathbf{\Theta}} -l(\mathbf{B}^{(m)}, \mathbf{\Theta}) + T \sum_{i \neq j} p_{\lambda_{\mathbf{\Theta}}}(|\theta_{ij}|) \tag{2.7}$$

5: If $\dfrac{\|\nabla_{\mathbf{B}} l(\mathbf{B}^{(m)}, \mathbf{\Theta}^{(m)}) - \nabla_{\mathbf{B}} l(\mathbf{B}^{(m)}, \mathbf{\Theta}^{(m-1)})\|}{T \cdot \max(1, \|(\mathbf{B}^{(m)}, \mathbf{\Theta}^{(m)})\|)} \geq 10^{-4}$, set m to $m+1$ and go to Step 3.

6: Set the solutions $\mathbf{B}_{(i_{\mathbf{B}}, i_{\mathbf{\Theta}})} = \mathbf{B}^{(m)}$, $\mathbf{\Theta}_{(i_{\mathbf{B}}, i_{\mathbf{\Theta}})} = \mathbf{\Theta}^{(m)}$ and set $(i_{\mathbf{B}}, i_{\mathbf{\Theta}})$ to next grid value by $i_{\mathbf{B}} = i_{\mathbf{B}} + 1$ and/ or $i_{\mathbf{\Theta}} = i_{\mathbf{\Theta}} + 1$ and go to Step 2. Repeat Steps 2 to 6 until $i_{\mathbf{B}} = i_{\mathbf{\Theta}} = 100$.

7: The final model is selected based on minimum BIC among the 10,000 grid estimates.

The size of regularization parameters $\lambda_{\mathbf{B}}$ and $\lambda_{\mathbf{\Theta}}$ controls the sparsity of the elements of AR coefficient and innovation precision matrix estimates. So a range of regularization parameters generates different estimated models. The remaining task is to choose the final model among the penalized models. When a coefficient estimate is zero, one less parameter is used in the model. Then we may use some well-known model selection techniques, such as Akaike Information Criterion (AIC) (Akaike (1973)), Bayesian Information Criterion (BIC) (Schwarz (1978)) , Hannan-Qunin Criterion (HQC) (Hannan and Quinn (1979)) and cross-validation method for choosing the final model. In practice, models are obtained by maximization of penalized likelihood function based on different $(\lambda_{\mathbf{B}}, \lambda_{\mathbf{\Theta}})$, each ranging from 0 to 1 with 100 equal divisions. BIC is easy to compute and is a commonly used technique for model selection, therefore, we use the minimum BIC approach to choose the final model.

The proposed algorithm uses a traditional way to obtain the final model based on penalized models estimated in full ranges of the regularization parameters $\lambda_{\mathbf{B}}$ and $\lambda_{\mathbf{\Theta}}$ from 0.01 to 1 with 100 divisions. This sets up a grid sized 10,000 points for model running. Thus, it is time consuming.

To speed up the estimation process, we use smaller ranges of the regularization parameters $\lambda_{\mathbf{B}}$ and $\lambda_{\mathbf{\Theta}}$ by a strategic way. The first stage to set up a frame with an initial step size of 0.05 used in the estimation algorithm. i.e. change 0.01 to 0.05 in Step 1 and change 100 in the Step 2 and the Step 6 as 20. Then the grid size is reduced from 10,000 points to 400 points in Step 7. Run the estimation algorithm and obtain an approximate optimal pair of $(\lambda_{\mathbf{B}}^{(a)}, \lambda_{\mathbf{\Theta}}^{(a)})$. Then, in the second stage, the step size is set to 0.01 in Step 1 and the maximum values for $\lambda_{\mathbf{B}}$ and $\lambda_{\mathbf{\Theta}}$ are set to slightly larger values of $\lambda_{\mathbf{B}}^{(a)}$ and $\lambda_{\mathbf{\Theta}}^{(a)}$ or up to $2 \times \lambda_{\mathbf{B}}^{(a)}$ and $2 \times \lambda_{\mathbf{\Theta}}^{(a)}$, and calculate the no. of division required in the Steps 2 and 6. Rerun the estimation algorithm again until none of the optimal value of $\lambda_{\mathbf{B}}$ and $\lambda_{\mathbf{\Theta}}$ lies on their maximum grid values on

the current frame used. If either the optimal regularization parameter value lies on the frame, there might exist a model outside the grid having a lower BIC than our selected model. So it is better to set the regularization parameter grid ranges wider after getting the approximated optimal $(\lambda_{\mathbf{B}}^{(a)}, \lambda_{\mathbf{\Theta}}^{(a)})$.

Other good features of the proposed algorithm are simple and efficient. First of all, our algorithm does not need local linear or quadratic approximation to the nonconvex penalty, SCAD and MCP, and provides elementwise closed-form solutions for Steps 3 and 4 in each iteration. The complexity of the algorithm would have not much increase with the dimension of the time series data. In addition, the step size of the line search in these two steps is initialized with Barzilai-Borwein (BB) rule (Barzilai and Borwein (1988)) at each outer iteration, which greatly accelerates the convergence speed. The stopping criterion of Step 5 is derived under the assumptions that $\mathbf{B}^{(m)}$ and $\mathbf{\Theta}^{(m)}$ satisfy the first order stationary conditions in Steps 3 and 4 and penalized likelihood function (2.5). It guarantees the proposed algorithm generates a stationary point for the $\mathbf{B}$ and $\mathbf{\Theta}$ pair because $(\mathbf{B}^{(m)}, \mathbf{\Theta}^{(m)})$ is kept updating until $\nabla l(\mathbf{B}^{(m)}, \mathbf{\Theta}^{(m)})$ converges to 0. The proposed algorithm is, in fact, a block coordinate gradient descent method.

2.2.3 Modified GIST algorithm

An extrapolation technique, proposed by Yu and Pong (2019) is incorporated with the non-monotone line search in the GIST algorithm in Gong et al. (2013) for further improvement of the convergence. Let the objective function to be minimized be $f(\mathbf{w}) = -l(\mathbf{w}) + r(\mathbf{w})$, where $l(.)$ is the likelihood function of a graphical VAR model, $r(.)$ is a penalty function, $\mathbf{w} = (w_{ij})$ and $r(\mathbf{w}) = \sum_{i,j} r_{ij}(w_{ij})$. The modified GIST algorithm is as follows.

1. Take $\xi = 10^{-4}$ for the tolerance parameter, as stated in Gong et al. (2013) and $m_s = 5$ for the number of iterations used in the line search criterion. Initialize

iteration counter k as 0 and a bounded starting point $\mathbf{w}^{(0)}$.

2. Set $t^{(k)} \in [10^{-8}, 10^{8}]$.

3. Solve

$$\mathbf{w}^{(k+1)} = \arg\min_{\mathbf{w}} l(\mathbf{w}^{(k)}) - \langle \nabla l(\mathbf{w}^{(k)}), \mathbf{w} - \mathbf{w}^{(k)} \rangle + \frac{t^{(k)}}{2} \|\mathbf{w} - \mathbf{w}^{(k)}\|^2 + r(\mathbf{w}) \quad (2.8)$$

4. Set $t^{(k)} = 2\, t^{(k)}$.

5. Go to Step 3 until this line search criterion is satisfied:

$$f(\mathbf{w}^{(k+1)}) \leq \max_{i \in \{\max(0, k-m_s+1), \ldots, k\}} f(\mathbf{w}^{(i)}) - \frac{\xi}{2} t^{(k)} \|\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)}\|^2,$$

   where m_s is set to 1 and to a value greater than 1 for monotone and nonmonotone line search in Step 1 respectively.

6. Set $k = k + 1$.

7. Go to Step 2 until the number of iterations (i.e. k) reaches 5000 or the following stopping criterion is satisfied:

$$\frac{\|\nabla l(\mathbf{w}^{(k)}) - \nabla l(\mathbf{w}^{(k+1)})\| + \frac{1}{t^{(k)}} \|\mathbf{w}^{(k)} - \mathbf{w}^{(k+1)}\|}{\max(1, \|\mathbf{w}^{(k+1)}\|)} < 10^{-4}.$$

Let $\mathbf{u}^{(k)} = (u_{ij}^{(k)}) = \mathbf{w}^{(k)} + \frac{\nabla l(\mathbf{w}^{(k)})}{t^{(k)}}$, where $\frac{1}{t^{(k)}}$ is the step size for the line search in the k-th iteration. The problem (2.8) can be solved dimension by dimension and the solutions can be obtained via the following elementwise univariate optimization problems:

$$w_{ij}^{(k+1)} = \arg\min_{w_{ij}} h_{ij} = \frac{1}{2}(w_{ij} - u_{ij}^{(k)})^2 + \frac{1}{t^{(k)}} r_{ij}(w_{ij}), \quad \text{for all } i, j.$$

To simplify the notation, we remove the subscripts ij and superscripts (k) in $u_{ij}^{(k)}$ for the following elementwise closed-form solutions for LASSO, SCAD and MCP:

(i) LASSO penalty

$$w^{(k+1)} = \text{sign}(u)\max(0, |u| - \frac{\lambda}{t}) \tag{2.9}$$

(ii) SCAD penalty

$$w^{(k+1)} = \arg\min_{y} h_{ij}(y) \text{ such that } y \in \{x_1, x_2, x_3\} \tag{2.10}$$

where

$$\begin{aligned} x_1 &= \text{sign}(u)\min\big(\lambda, \max(0, |u| - \lambda/t)\big), \\ x_2 &= \text{sign}(u)\min\left(\phi\lambda, \max(\lambda, \frac{t|u|(\phi-1) - \phi\lambda)}{t(\phi-1)-1})\right), \\ x_3 &= \text{sign}(u)\max(\phi\lambda, |u|) \end{aligned}$$

(iii) MCP penalty

$$w^{(k+1)} = \begin{cases} x_1, & \text{if } h_{ij}(x_1) \le h_{ij}(x_2) \\ x_2, & \text{otherwise.} \end{cases} \tag{2.11}$$

where

$$\begin{aligned} x_1 &= \text{sign}(u)\arg\min_{w\in\mathcal{C}} \frac{1}{2}(w - |u|)^2 + \frac{\lambda}{t}|w| - \frac{w^2}{2\phi t} \text{ with} \\ & \mathcal{C} = \begin{cases} \{0, \phi\lambda, \min(\phi\lambda, \max(0, \frac{\phi(t|u|-\lambda)}{\phi t-1}))\} & \text{if } \phi t - 1 \neq 0 \\ \{0, \phi\lambda\} & \text{otherwise.} \end{cases} \\ x_2 &= \text{sign}(u)\max(\phi\lambda, |u|), \end{aligned}$$

It is obvious that $\mathbf{w}$ and $\nabla l(\mathbf{w})$ refer to $\mathbf{B}$ and $\nabla l(\mathbf{B}, \mathbf{\Theta}^{(k-1)})$ for $\mathbf{B}$ estimation and to $\mathbf{\Theta}$ and $\nabla l(\mathbf{B}^{(k-1)}, \mathbf{\Theta})$ for $\mathbf{\Theta}$ estimation in the k-th iteration.

2.2.4 Convergence of the algorithm

In order to prove the convergence of the algorithm, we reformulate our proposed iterative alternating estimation algorithm in the framework of block coordinate gradient descent method using monotone line search and followed by non-monotone line search.

We aim to express the minimisation problem (2.5) as

$$\arg\min_{\mathbf{B},\mathbf{\Theta}} F(\mathbf{B},\mathbf{\Theta}) := f(\mathbf{B},\mathbf{\Theta}) + P_{\mathbf{B},1}(\mathbf{B}) + P_{\mathbf{\Theta},1}(\mathbf{\Theta}), \tag{2.12}$$

where $f(\mathbf{B},\mathbf{\Theta})$ is a continuously differentiable function with locally Lipschitz gradient, $P_{\mathbf{B},1}(\mathbf{B})$ and $P_{\mathbf{\Theta},1}(\mathbf{\Theta})$ are proper closed convex non-negative functions.

Lasso penalty is a convex non-negative function, but the SCAD penalty and MCP functions considered are not convex. Gong et al.(2013) express them as difference of two proper closed convex functions, listed in Table 2.1. Let

$$P_{\lambda_{\mathbf{B}},1}(|b_{ij}|) = \lambda_{\mathbf{B}}|b_{ij}|, \;\; P_{\lambda_{\mathbf{\Theta}},1}(|\theta_{ij}|) = \lambda_{\mathbf{\Theta}}|\theta_{ij}|.$$

They are l_1-norm functions. Then we assume that $P_{\lambda_{\mathbf{B}}}(|b_{ij}|) = P_{\lambda_{\mathbf{B}},1}(|b_{ij}|) - P_{\lambda_{\mathbf{B}},2}(|b_{ij}|)$ and $P_{\lambda_{\mathbf{\Theta}}}(|\theta_{ij}|) = P_{\lambda_{\mathbf{\Theta}},1}(|\theta_{ij}|) - P_{\lambda_{\mathbf{\Theta}},2}(|\theta_{ij}|)$, where $P_{\lambda_{\mathbf{B}},2}(|b_{ij}|)$ and $P_{\lambda_{\mathbf{\Theta}},2}(|\theta_{ij}|)$ are proper closed convex functions as a polynomials of degree two in b_{ij} and θ_{ij} respectively. As listed in Table 2.1, their function forms varies with the location of b_{ij} and θ_{ij}.

Define

$$P_{\mathbf{B},1}(\mathbf{B}) = T\sum_{i,j} P_{\lambda_{\mathbf{B}},1}(|b_{ij}|), \;\; P_{\mathbf{B},2}(\mathbf{B}) = T\sum_{i,j} P_{\lambda_{\mathbf{B}},2}(|b_{ij}|), \text{ and} \tag{2.13}$$

$$P_{\mathbf{\Theta},1}(\mathbf{\Theta}) = T\sum_{i\neq j} P_{\lambda_{\mathbf{\Theta}},1}(|\theta_{ij}|), \;\; P_{\mathbf{\Theta},2}(\mathbf{\Theta}) = T\sum_{i\neq j} P_{\lambda_{\mathbf{\Theta}},2}(|\theta_{ij}|) \tag{2.14}$$

Together with the above expressions (2.13) and (2.14) , we can define the functions

in the problem (2.12) as follows.

$$\begin{aligned} f(\mathbf{B},\mathbf{\Theta}) &= -l(\mathbf{B},\mathbf{\Theta}) - P_{\mathbf{B},2}(\mathbf{B}) - P_{\mathbf{\Theta},2}(\mathbf{\Theta}) \\ P_{\mathbf{B},1}(\mathbf{B}) &= T\sum_{i,j}\lambda_{\mathbf{B}}|b_{ij}| \\ P_{\mathbf{\Theta},1}(\mathbf{\Theta}) &= T\sum_{i\neq j}\lambda_{\mathbf{\Theta}}|\theta_{ij}| \end{aligned}$$

We simplify the notation of $\mathbf{B}$ and $\mathbf{\Theta}$ by denoting x_1 as $vec(\mathbf{B})$, x_2 as $vec(\mathbf{\Theta})$, $\mathbf{x} = (x_1, x_2)$, $P_1(x_1)$ as $P_{\mathbf{B},1}(\mathbf{B})$ and $P_2(x_2)$ as $P_{\mathbf{\Theta},1}(\mathbf{\Theta})$. The problem (2.12) becomes:

$$\arg\min_{\mathbf{x}=(x_1,x_2)} F(x_1,x_2) := f(x_1,x_2) + P_1(x_1) + P_2(x_2)$$

and rewrite the alternating estimation algorithm into the following framework of block coordinate descent gradient algorithm based on monotone line search:

1: Choose parameter $\gamma > 1$, $c > 0$ and $L_{\min}, L_{\max}$ with $0 < L_{\min} < L_{\max}$. Initialise iteration counter $k \leftarrow 0$.

2: Pick $i_k \in \{1,2\}$, $L_0^{(k)} \in [L_{\min}, L_{\max}]$ and a bounded starting point $\mathbf{x}^{(0)}$. Set $\tilde{L} = L_0^{(k)}$.

3: Solve

$$\tilde{x}_{i_k} = \arg\min_{x_{i_k}}\{\langle\nabla_{i_k} f(\mathbf{x}^{(k)}), x_{i_k} - x_{i_k}^{(k)}\rangle + P_{i_k}(x_{i_k}) + \frac{\tilde{L}}{2}\|x_{i_k} - x_{i_k}^{(k)}\|^2\} \qquad (2.15)$$

and obtain $\tilde{\mathbf{x}} := (\tilde{x}_{i_k}, x^{(k)}_{\{1,2\}\setminus i_k})$.

4: If $F(\tilde{\mathbf{x}}) > F(\mathbf{x}^{(k)}) - \frac{c}{2}\|\tilde{x}_{i_k} - x_{i_k}^{(k)}\|^2$, update $\tilde{L} \leftarrow \gamma\tilde{L}$ and go to Step 3 and else go to Step 5.

5: Set $\mathbf{x}^{(k+1)} = \tilde{\mathbf{x}}$ and $L^{(k)} = \tilde{L}$. Update $k \leftarrow k+1$ and go to Step 2.

In our two steps estimation algorithm, $\gamma = 2$, $c = 10^{-4}$, $L_{\min} = 10^{-8}$, $L_{\max} = 10^{8}$ and $L_0^{(k)} = t^{(0)}$. Recall the objective function given in GIST algorithm in Gong et al. (2013),

$$\begin{aligned}\mathbf{w}^{(k+1)} &= \arg\min_{\mathbf{w}} \; l(\mathbf{w}^{(k+1)}) + < \nabla l(\mathbf{w}^{(k)}), \mathbf{w} - \mathbf{w}^{(k)} > + \frac{t^{(k)}}{2}\|\mathbf{w} - \mathbf{w}^{(k)}\|^2 \\ &\quad + p(\mathbf{w}), \end{aligned} \tag{2.16}$$

where $l(\mathbf{w})$, $p(\mathbf{w})$ and k are likelihood function, non-smooth and non-convex continuous penalty function and k is the inner iteration counter within GIST algorithm respectively. To ease for the convergence analysis, this function for $\mathbf{B}$ and $\boldsymbol{\Theta}$ in Steps 2 and 3 in the two steps alternating estimation algorithm is required to be converted into the minimisation problem (2.15) in Step 3 above. Here are the details of the conversion. From Equation (2.16),we have

$$\begin{aligned}\mathbf{B}^{(k+1)} &= \arg\min_{\mathbf{B}} \; -l(\mathbf{B}^{(k)}, \boldsymbol{\Theta}^{(k)}) - \left\langle \nabla_{\mathbf{B}} l(\mathbf{B}^{(k)}, \boldsymbol{\Theta}^{(k)}), \mathbf{B} - \mathbf{B}^{(k)} \right\rangle \\ &\quad + \frac{t^{(k)}}{2}\|\mathbf{B} - \mathbf{B}^{(k)}\|^2 + P_{\mathbf{B},1}(\mathbf{B}) - P_{\mathbf{B},2}(\mathbf{B}) \\ &= \arg\min_{\mathbf{B}} \; -l(\mathbf{B}^{(k)}, \boldsymbol{\Theta}^{(k)}) - P_{\mathbf{B},2}(\mathbf{B}^{(k)}) \\ &\quad - \left\langle \nabla_{\mathbf{B}} l(\mathbf{B}^{(k)}, \boldsymbol{\Theta}^{(k)}), \mathbf{B} - \mathbf{B}^{(k)} \right\rangle - \left\langle \nabla P_{\mathbf{B},2}(\mathbf{B}^{(k)}), \mathbf{B} - \mathbf{B}^{(k)} \right\rangle \\ &\quad + \frac{c_{\mathbf{B}}}{2}\|\mathbf{B} - \mathbf{B}^{(k)}\|^2 + \frac{t^{(k)}}{2}\|\mathbf{B} - \mathbf{B}^{(k)}\|^2 + P_{\mathbf{B},1}(\mathbf{B}) \\ &= \arg\min_{\mathbf{B}} \; \left\langle -\nabla_{\mathbf{B}} l(\mathbf{B}^{(k)}, \boldsymbol{\Theta}^{(k)}) - \nabla P_{\mathbf{B},2}(\mathbf{B}^{(k)}), \mathbf{B} - \mathbf{B}^{(k)} \right\rangle \\ &\quad + \frac{c_{\mathbf{B}} + t^{(k)}}{2}\|\mathbf{B} - \mathbf{B}^{(k)}\|^2 + P_{\mathbf{B},1}(\mathbf{B}), \end{aligned} \tag{2.17}$$

$$
\begin{aligned}
\boldsymbol{\Theta}^{(k+1)} = & \arg\min_{\boldsymbol{\Theta}} \quad -l(\mathbf{B}^{(k+1)}, \boldsymbol{\Theta}^{(k)}) - \left\langle \nabla_{\boldsymbol{\Theta}} l(\mathbf{B}^{(k+1)}, \boldsymbol{\Theta}^{(k)}), \boldsymbol{\Theta} - \boldsymbol{\Theta}^{(k)} \right\rangle \\
& \qquad + \frac{t'^{(k)}}{2} \|\boldsymbol{\Theta} - \boldsymbol{\Theta}^{(k)}\|^2 + P_{\boldsymbol{\Theta},1}(\boldsymbol{\Theta}) - P_{\boldsymbol{\Theta},2}(\boldsymbol{\Theta}) \\
= & \arg\min_{\boldsymbol{\Theta}} \quad -l(\mathbf{B}^{(k+1)}, \boldsymbol{\Theta}^{(k)}) - P_{\boldsymbol{\Theta},2}(\boldsymbol{\Theta}^{(k)}) \\
& \qquad - \left\langle \nabla_{\boldsymbol{\Theta}} l(\mathbf{B}^{(k+1)}, \boldsymbol{\Theta}^{(k)}), \boldsymbol{\Theta} - \boldsymbol{\Theta}^{(k)} \right\rangle - \left\langle \nabla P_{\boldsymbol{\Theta},2}(\boldsymbol{\Theta}^{(k)}), \boldsymbol{\Theta} - \boldsymbol{\Theta}^{(k)} \right\rangle \\
& \qquad + \frac{t'^{(k)}}{2} \|\boldsymbol{\Theta} - \boldsymbol{\Theta}^{(k)}\|^2 + \frac{c_{\boldsymbol{\Theta}}}{2} \|\boldsymbol{\Theta} - \boldsymbol{\Theta}^{(k)}\|^2 + P_{\boldsymbol{\Theta},1}(\boldsymbol{\Theta}) \\
= & \arg\min_{\boldsymbol{\Theta}} \quad \left\langle -\nabla_{\boldsymbol{\Theta}} l(\mathbf{B}^{(k+1)}, \boldsymbol{\Theta}^{(k)}) - \nabla P_{\boldsymbol{\Theta},2}(\boldsymbol{\Theta}^{(k)}), \boldsymbol{\Theta} - \boldsymbol{\Theta}^{(k)} \right\rangle \\
& \qquad + \frac{t'^{(k)} + c_{\boldsymbol{\Theta}}}{2} \|\boldsymbol{\Theta} - \boldsymbol{\Theta}^{(k)}\|^2 + P_{\boldsymbol{\Theta},1}(\boldsymbol{\Theta}), \qquad (2.18)
\end{aligned}
$$

where $t^{(k)}, t'^{(k)}$ are the step sizes used in GIST algorithm and $c_{\mathbf{B}}$ and $c_{\boldsymbol{\Theta}}$ are some constants derived from Taylor's exapansion. The second equalities of Equations (2.17) and (2.18) follows from Taylor's expansion and the fact that $P_{\mathbf{B},2}(\mathbf{B})$ and $P_{\boldsymbol{\Theta},2}(\boldsymbol{\Theta})$ for Lasso penalty, SCAD penalty and MCP are polynomials up to degree two in b_{ij} and θ_{ij} respectively. It is clear that Equations (2.17) and (2.18) are equivalent to Equation (2.15) in our GIST algorithm. Therefore, we can use the block coordinate descent algorithm approach to discuss the convergence. Lemma 2.1 guarantees finite number of steps in the line search and Lemma 2.2 proves the convergence of the algorithm using monotone line search. Lemma 2.3 extends the results to non-monotone line search.

Lemma 2.1. *Assume that f is continuously differentiable with locally Lipschitz gradient (i.e. for any compact set, there exists a L_f such that $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L_f \|\mathbf{x} - \mathbf{y}\|$) and is bounded below and $P_{i_k}(x_{i_k})$ for $i_k \in \{1, 2\}$ are proper closed convex non-negative functions. Then the line search in Step 4 for the equation (2.15) is well-defined.*

Proof We prove by induction. Suppose $\mathbf{x}^{(k)}$ is well-defined. Since $\tilde{x}_{i_k}$ is a minimizer

of problem (2.15), we have

$$\langle \nabla_{i_k} f(\mathbf{x}^{(k)}), \tilde{x}_{i_k} - x_{i_k}^{(k)} \rangle + \frac{\tilde{L}}{2} \|\tilde{x}_{i_k} - x_{i_k}^{(k)}\|^2 + P_{i_k}(\tilde{x}_{i_k}) \leq P_{i_k}(\tilde{x}_{i_k}^{(k)}),$$

for any $\tilde{L}$. Using the inequality of $\langle \mathbf{u}, \mathbf{v} \rangle \geq -\|\mathbf{u}\| \cdot \|\mathbf{v}\|$ and the fact that $P_{i_k}(\cdot) \geq 0$, we have

$$\frac{\tilde{L}}{2} \|\tilde{x}_{i_k} - x_{i_k}^{(k)}\|^2 - \|\nabla_{i_k} f(\mathbf{x}^{(k)})\| \|\tilde{x}_{i_k} - x_{i_k}^{(k)}\| \leq P_{i_k}(\tilde{x}_{i_k}^{(k)}),$$

Therefore $\|\tilde{x}_{i_k} - x_{i_k}^{(k)}\|$ is bounded by $\frac{\|\nabla_{i_k} f(\mathbf{x}^{(k)})\| + \sqrt{\|\nabla_{i_k} f(\mathbf{x}^{(k)})\|^2 + 2\tilde{L} P_{i_k}(x_{i_k}^{(k)})}}{\tilde{L}}$. When $\tilde{L}$ is large enough, i.e. $\tilde{L} > \hat{L}_k$ for some large $\hat{L}_k$, then, by locally Lipschitz gradient property of f, we can assume that $\tilde{\mathbf{x}}$ and $\mathbf{x}^{(k)}$ both lie in a ball on which ∇f is Lipschitz continuous with modulus $L_{f,k}$.

Using Taylor's inequality, we have

$$\begin{aligned}
F(\tilde{\mathbf{x}}) &\leq f(\mathbf{x}^{(k)}) + \langle \nabla f(\mathbf{x}^{(k)}), \tilde{\mathbf{x}} - \mathbf{x}^{(k)} \rangle + \frac{L_{f,k}}{2} \|\tilde{\mathbf{x}} - \mathbf{x}^{(k)}\|^2 + P_1(\tilde{x}_1) + P_2(\tilde{x}_2) \\
&= f(\mathbf{x}^{(k)}) + \langle \nabla_{i_k} f(\mathbf{x}^{(k)}, \tilde{x}_{i_k} - x_{i_k}^{(k)} \rangle + \frac{\tilde{L}}{2} \|\tilde{x}_{i_k} - x_{i_k}^{(k)}\|^2 + P_1(\tilde{x}_1) + P_2(\tilde{x}_2) \\
&\qquad + \frac{L_{f,k} - \tilde{L}}{2} \|\tilde{x}_{i_k} - x_{i_k}^{(k)}\|^2 \\
&\leq F(\mathbf{x}^{(k)}) + \frac{L_{f,k} - \tilde{L}}{2} \|\tilde{x}_{i_k} - x_{i_k}^{(k)}\|^2 \qquad (2.19)
\end{aligned}$$

The last inequality follows by the fact that $\tilde{\mathbf{x}}$ is minimizer of Step 3. By defining $\tilde{L} > \max\{\hat{L}_k, L_{f,k} + c\}$, the line search holds. □

Lemma 2.2. *Under the same assumption of Lemma 2.1 and boundedness of* $\{\mathbf{x}^{(k)}\}_{k \in \mathcal{K}}$, *any accumulation point* $\mathbf{x}^*$ *of the sequence* $\{\mathbf{x}^{(k)}\}_{k \in \mathcal{K}}$ *generated by the above algorithm is a stationary point of* $F(\mathbf{x})$.

Proof From the monotone line search criterion used in Step 4,

$$F(\mathbf{x}^{(k+1)}) \leq F(\mathbf{x}^{(k)}) - \frac{c}{2} \|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|^2, \qquad (2.20)$$

we have

$$\sum_{k=0}^{\infty} \|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| \leq \frac{2}{c}\left(F(\mathbf{x}^{(0)}) - \inf F\right) < \infty.$$

This implies that $\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \to 0$. i.e. there exists a positive integer ϵ and M such that $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| > \epsilon$ for $k \leq M$. Together with the boundedness of $\{x^{(k)}\}_{k\in\mathcal{K}}$, we can find a bounded set S such that $x^{(k)} \in S$ for all $k = 1, 2, \ldots$. Since f has locally Lipschitz gradients, we can find a positive constant L_S ($\geq L_{f,k}$ for all $0 < k \leq M$) such that $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| < L_S\|\mathbf{x} - \mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in S$. In this case, we can argue in a similar way as in Lemma 2.1 that the line search criterion (2.20) holds whenever $\tilde{L} > L_S + c$.

Since the choice of i_k is essentially cyclic, i.e. there exists a T such that $\{1, 2\} \subseteq \{i, i+1, \ldots, i+T-1\}$. From Equation (2.15), we have

$$0 \in \nabla_{i_k} f(\mathbf{x}^{(k)}) + \partial P_{i_k}(\mathbf{x}^{(k+1)}) + L^{(k)}(x_{i_k}^{(k+1)} - x_{i_k}^{(k)}) \tag{2.21}$$

for all k.

We now claim that $L^{(k)}$ is bounded for all $k \geq 0$. Let n_k be the number of inner iterations for the k-th outer iteration. By the definition of $L^{(k)}$, $L_{\min}\gamma^{n_k-1} \leq L_0^{(k)}\gamma^{n_k-1}$, and by the locally Lipschitz property of f on a bounded set S, we have $\tilde{L} > L_S + c$, we have $L^{(k)}/\gamma < L_S + c$ or $L^{(k)} < \gamma(L_S + c)$. Then we have

$$L_{\min}\gamma^{n_k-1} \leq L_0^{(k)}\gamma^{n_k-1} < \gamma(L_S + c).$$

Hence, $n_k \leq \left\lceil \frac{\log(L_S+c) - \log L_{\min}}{\log \gamma} + 2 \right\rceil$. Then $L^{(k)} = L_0^{(k)}\gamma^{\left\lceil \frac{\log(L_S+c)-\log L_{\min}}{\log\gamma}+1\right\rceil}$, and hence is bounded.

Since $\mathbf{x}^*$ be an accumulation point of the sequence $\{\mathbf{x}^{(k)}\}$, then there exists a further subsequence of $\{i_{k_j}\}$ converging to either 1 or 2. Since $\{L^{(k)}\}$ is bounded and $\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \to 0$, we have $x_{i_{k_j}}^{(k_j+1)} \to x_{i_{k_j}}^{(k_j)}$ and the term $L^{(k_j)}(x_{i_{k_j}}^{(k_j+1)} - x_{i_{k_j}}^{(k_j)}) \to 0$.

Since the choice of i_k is essentially cyclic, $\nabla f_{i_{k_j}}(.)$ and $P_{i_k}(.)$ are Lipschitz continuous, $\mathbf{x}^{(k_j+l)} \to \mathbf{x}^*$ holds for $l = i, i+1, \ldots, i+T-1$ and Statement (2.21) holds for $k = k_j + l$ and $l = i, i+1, \ldots, i+T-1$. Therefore, taking limits on both sides of statement (2.21), we have

$$0 \in \nabla_1 f(\mathbf{x}^*) + \partial P_1(x_1^*) \tag{2.22}$$

$$0 \in \nabla_2 f(\mathbf{x}^*) + \partial P_2(x_2^*) \tag{2.23}$$

Thus, by combining (2.22) and (2.23), we have

$$0 \in \nabla f(\mathbf{x}^*) + \begin{bmatrix} \partial P_1(x_1^*) \\ \partial P_2(x_2^*) \end{bmatrix}.$$

Therefore $\mathbf{x}^*$ is a stationary point. □

Now we move on the algorithm using nonmonotonic line search criterion and Steps 1 and 4 are revised accordingly in the following:

1: Choose parameter $\gamma > 1$, $m_s > 1$, $c > 0$ and $L_{\min}, L_{\max}$ with $0 < L_{\min} < L_{\max}$. Initialise iteration counter $k \leftarrow 0$

4: If

$$F(\tilde{\mathbf{x}}) > \max_{\max\{0,k-m_s+1\} \le i \le k} F(\mathbf{x}^{(i)}) - \frac{c}{2}\|\tilde{x}_{i_k} - x_{i_k}^{(k)}\|^2,$$

update $\tilde{L} \leftarrow \gamma \tilde{L}$ and go to Step 3 and else go to Step 5.

The monotone line search criterion is more stringent than the nonmonotone line search criterion. Therefore, the nonmonotone line search criterion is well-defined.

Lemma 2.3. *Under the same assumptions of Lemma 2.1 and boundedness of* $\{\mathbf{x}^{(k)}\}$, *any accumulation point* $\mathbf{x}^*$ *of the sequence* $\{\mathbf{x}^{(k)}\}_{k \in K}$ *generated by the algorithm using nonmonotone line search is a stationary point of* $F(\mathbf{x})$.

Proof Let $s(k)$ be an integer such that $\max\{0, k-m_s+1\} \le s(k) \le k$ for all $k \ge 0$. Define

$$F(\mathbf{x}^{(s(k))}) = \max\left\{F(\mathbf{x}^{(i)}) : \max\{0, k-m_s+1\} \le s(k) \le k\right\} \text{ for all } k \ge 0.$$

From the nonmonotone line search criterion,

$$F(\mathbf{x}^{(k+1)}) \le \max_{\max\{0,k-m_s+1\}\le i\le k} F(\mathbf{x}^{(i)}) - \frac{c}{2}\|x_{i_k}^{(k+1)} - x_{i_k}^{(k)}\|^2, \tag{2.24}$$

we have

$$F(\mathbf{x}^{(k+1)}) \le F(\mathbf{x}^{(s(k))}) \text{ for all } k \ge 0.$$

Consider

$$\begin{aligned}
& F(\mathbf{x}^{(s(k+1))}) - F(\mathbf{x}^{(s(k)}))\\
= \;& \max\left\{F(\mathbf{x}^{(k+1)}), \max\left\{F(\mathbf{x}^{(i)}) : \max\{0, k-m_s+2\} \le s(k) \le k\right\}\right\} - F(\mathbf{x}^{s(k)})\\
\le \;& \max\left\{F(\mathbf{x}^{(k+1)}), F(\mathbf{x}^{(s(k))})\right\} - F(\mathbf{x}^{(s(k))})\\
\le \;& \max\left\{F(\mathbf{x}^{(s(k))}), F(\mathbf{x}^{(s(k))})\right\} - F(\mathbf{x}^{(s(k))})\\
= \;& 0
\end{aligned}$$

for any $k \ge 0$. This indicates that $\{F(\mathbf{x}^{(s(k))})\}_{k\in K}$ is a monotonic nonincreasing sequence. Since F is bounded below, there exists a constant F^* such that

$$\lim_{k\to\infty} F(\mathbf{x}^{(s(k))}) = F^*.$$

By replacing k with $s(k)-1$ in (2.24), we have

$$F(\mathbf{x}^{(s(k))}) < F(\mathbf{x}^{(s(s(k)-1))}) - \frac{c}{2}\|x_{i_k}^{(s(k))} - x_{i_k}^{(s(k)-1)}\|^2.$$

Since $\lim_{k\to\infty} F(\mathbf{x}^{(s(s(k)-1))}) = \lim_{k\to\infty} F(\mathbf{x}^{(s(k))}) = F^*$ and we take limit on both side of the above inequality, then

$$\lim_{k\to\infty} c\|x_{i_k}^{(s(k))} - x_{i_k}^{(s(k)-1)}\|^2 = 0.$$

Indeed, c is a a positive constant. Then we have

$$\lim_{k\to\infty} x_{i_k}^{(s(k))} - x_{i_k}^{(s(k)-1)} = 0. \tag{2.25}$$

Hence, $\lim_{k\to\infty} \mathbf{x}^{(s(k))} - \mathbf{x}^{(s(k)-1)} = 0$. In addition, by continuity of F, we have

$$\begin{aligned} F^* &= \lim_{k\to\infty} F(x^{(s(k))}) \\ &= \lim_{k\to\infty} F(x^{(s(k)-1)} + x^{(s(k))} - x^{(s(k)-1)}) \\ &= \lim_{k\to\infty} F(x^{(s(k)-1)}) \end{aligned} \tag{2.26}$$

We will now prove by induction that

$$\lim_{k\to\infty} x_{i_k}^{(s(k)-j+1)} - x_{i_k}^{(s(k)-j)} = 0 \quad \text{and} \quad \lim_{k\to\infty} F(x^{(s(k)-j)}) = F^* \quad \text{for } j = 1, 2, \ldots \tag{2.27}$$

For the case of $j = 1$, the results follow from (2.25) and (2.26). Now assume that the limits in (2.27) hold for j and we are going to prove that they hold for the case $j + 1$.

By replacing k with $s(k) - j - 1$ in the nonmonotone line search criterion, we have

$$F(\mathbf{x}^{(s(k)-j)}) < F(\mathbf{x}^{(s(s(k)-j-1))}) - \frac{c}{2}\|x_{i_k}^{(s(k)-j)} - x_{i_k}^{(s(k)-j-1)}\|^2.$$

Then

$$\|x_{i_k}^{(s(k))-j} - x_{i_k}^{(s(k)-j-1)}\|^2 \leq \frac{2}{c}\left(F(\mathbf{x}^{(s(s(k)-j-1))}) - F(\mathbf{x}^{(s(k)-j)})\right)$$

By letting $k \to \infty$ and using the limits of $F(\mathbf{x}^{(s(s(k)-j-1))})$ being equal to that of $F(\mathbf{x}^{(s(k)-j)})$ in (2.27), $x_{i_k}^{(s(k)-j)} - x_{i_k}^{(s(k)-j-1)} \to 0$. Then, by the continuity of F and

(2.27) again, we have

$$\begin{aligned}
&\lim_{k\to\infty} F(\mathbf{x}^{(s(k)-(j+1))}) \\
= &\lim_{k\to\infty} F\left(\mathbf{x}^{(s(k)-j)} - (x^{(s(k)-j)} - x^{(s(k)-j-1)})\right) \\
= &\lim_{k\to\infty} F(\mathbf{x}^{(s(k)-j)}) \\
= &\ F^*
\end{aligned}$$

We are going to prove $F(\mathbf{x}^{(k)}) \to F^*$.

Note that $\mathbf{x}^{s(k)} = \mathbf{x}^{(\max\{0,k-m_s+1\})} + \sum_{j=1}^{s(k)-(k-m_s+1)} \left(\mathbf{x}^{(s(k)-j+1)} - \mathbf{x}^{(s(k)-j)}\right)$ for all k. Since $\mathbf{x}^{(k)} = (x_1^{(k)}, x_2^{(k)})$ and by the limits in (2.27), we have

$$x_{i_k}^{(s(k)-j+1)} - x_{i_k}^{(s(k)-j)} \to 0 \text{ and } x_{\{1,2\}\setminus i_k}^{(s(k)-j+1)} - x_{\{1,2\}\setminus i_k}^{(s(k)-j)} = 0.$$

Thus, $\mathbf{x}^{(s(k)-j+1)} - \mathbf{x}^{(s(k)-j)} \to 0$. Then

$$\lim_{k\to\infty} \mathbf{x}^{(s(k))} - \mathbf{x}^{(k-m_s+1)} = 0.$$

Therefore, by the continuity of F,

$$\begin{aligned}
\lim_{k\to\infty} F(\mathbf{x}^{(k)}) &= F(\lim_{k\to\infty} \mathbf{x}^{(k)}) \\
&= F(\lim_{k\to\infty} \mathbf{x}^{(k-m_s+1)}) \\
&= F(\lim_{k\to\infty} \mathbf{x}^{(s(k))}) \\
&= \lim_{k\to\infty} F(\mathbf{x}^{(s(k))}) \\
&= F^*. \qquad (2.28)
\end{aligned}$$

Taking limit on the both sides of (2.24) and using (2.28), we have

$$\|x_{i_k}^{(k+1)} - x_{i_k}^{(k)}\|^2 \to 0 \text{ for all } k.$$

Using a similar argument on the boundedness of $L^{(k)}$ in **Lemma 2.2**, $L^{(k)}$ is bounded.

Assume the choice of i_k is essentially cyclic, i.e. there exists a T such that $\{1,2\} \subseteq \{i, i+1, \ldots, i+T-1\}$. From Equation (2.15), we have

$$0 \in \nabla_{i_k} f(\mathbf{x}^{(k)}) + \partial P_{i_k}(\mathbf{x}^{(k+1)}) + L^{(k)}(x_{i_k}^{(k+1)} - x_{i_k}^{(k)}) \tag{2.29}$$

for all k.

There exists a sequence $\{\mathbf{x}^{(k_j)}\}$ such that $\mathbf{x}^{(k_j)} \to \mathbf{x}^*$. There exists a further subsequence of $\{i_{k_j}\}$ converging to 1 or 2. Since $L^{(k)}$ is bounded and $\mathbf{x}_{i_k}^{(k+1)} - \mathbf{x}_{i_k}^{(k)} \to 0$, the term $L^{(k)}(x_{i_k}^{(k+1)} - x_{i_k}^{(k)}) \to 0$. Since the choice of i_k is essentially cyclic and $\mathbf{x}^{(k_j)} \to \mathbf{x}^*$, $\mathbf{x}^{(k)} \to \mathbf{x}^*$. Therefore, taking limits on both sides of statement (2.29), we have

$$0 \in \nabla_1 f(\mathbf{x}^*) + \partial P_1(x_1^*) \tag{2.30}$$

$$0 \in \nabla_2 f(\mathbf{x}^*) + \partial P_2(x_2^*) \tag{2.31}$$

Thus, by combining (2.30) and (2.31), we have

$$0 \in \nabla f(\mathbf{x}^*) + \begin{bmatrix} \partial P_1(x_1^*) \\ \partial P_2(x_2^*) \end{bmatrix}.$$

Therefore $\mathbf{x}^*$ is a stationary point. □

Theorem 2.1. *Let the sequence* $\{(\mathbf{B}^{(k)}, \mathbf{\Theta}^{(k)})\}$ *be generated by the two-stage general iterative shrinkage and thresholding algorithm. Suppose* $(\mathbf{B}^*, \mathbf{\Theta}^*)$ *is any accumulation point of* $\{(\mathbf{B}^{(k)}, \mathbf{\Theta}^{(k)})\}$. $(\mathbf{B}^*, \mathbf{\Theta}^*)$ *is a first-order stationary point of the penalized likelihood function (2.5).*

Proof: It follows from Lemma 2.3. □

2.2.5 Consistency and asymptotic normality

The last section discusses the convergence of the algorithm and this section discusses the features of the penalized maximum likelihood estimator - consistency and asymptotic normality. Chu et al. (2011a) proved that the penalized maximum likelihood estimator of their geostatistical model was consistent and converged asymptotically to a normal distribution. We consider the asymptotic framework of their geostatistical model to derive the consistency and the limiting distribution of our penalized maximum likelihood estimator.

Here we set up the notations. Let $\boldsymbol{\beta}_0 = \text{vec}(\mathbf{B}_0)$ be the true vectorized autoregressive coefficient matrix. Let $\boldsymbol{\eta}_0 = \text{vec}(\boldsymbol{\Theta})_0$ be the true precision matrix. Without the loss of generality, we assume $\boldsymbol{\beta}_0 = (\beta_{10}, \ldots, \beta_{k^2p,0})^T = (\boldsymbol{\beta}_{10}^T, \boldsymbol{\beta}_{20}^T)^T$ with $\dim(\boldsymbol{\beta}_{10}) = s$, $\dim(\boldsymbol{\beta}_{20}) = k^2p - s$ and $\boldsymbol{\beta}_{20} = \mathbf{0}$. Similarly, $\boldsymbol{\eta}_0 = (\eta_{10}, \ldots, \eta_{k^2,0})^T = (\boldsymbol{\eta}_{10}^T, \boldsymbol{\eta}_{20}^T)$ with $\dim(\boldsymbol{\eta}_{10}) = \nu$, $\dim(\boldsymbol{\eta}_{20}) = k^2 - \nu$ and $\boldsymbol{\eta}_{20} = \mathbf{0}$.

The log-likelihood function of our model is:

$$\begin{aligned} l(\boldsymbol{\beta}, \boldsymbol{\Theta}) &= -\frac{KT}{2}\log 2\pi + \frac{T}{2}\log(\det(\boldsymbol{\Theta})) \\ &\quad -\frac{1}{2}\left(\text{vec}(\mathbf{Y}) - (\mathbf{Z}^T \otimes \mathbf{I}_k)^T\boldsymbol{\beta}\right)^T (\mathbf{I}_k \otimes \boldsymbol{\Theta}) \left(\text{vec}(\mathbf{Y}) - (\mathbf{Z}^T \otimes \mathbf{I}_k)^T\boldsymbol{\beta}\right). \end{aligned} \tag{2.32}$$

and the penalized log-likelihood function (2.5) is written as

$$Q(\boldsymbol{\beta}, \boldsymbol{\eta}) = l(\boldsymbol{\beta}, \boldsymbol{\eta}) - T\sum_{i}^{k^2p} p_{\lambda_{\mathbf{B}}}(|\beta_i|) - T \sum_{\substack{i=1 \\ i \neq (r-1)k+r, r \in \{1,\ldots,k\}}}^{k^2} p_{\lambda_{\boldsymbol{\Theta}}}(|\eta_i|) \tag{2.33}$$

where $\boldsymbol{\beta} = \text{vec}(\mathbf{B})$ and $\boldsymbol{\eta} = \text{vec}(\boldsymbol{\Theta})$.

Let n be the stage of the asymptotics. In particular, we write $T_n = T$, $\lambda_{\mathbf{B},n} = \lambda_{\boldsymbol{\Theta},n} = \lambda$ and $k_n^2 = k^2$. For the quantities which depend on the stage n, n will be in either the left superscript or right subscript of the quantities.

Define penalty-related scalars: $a_{\mathbf{B},n} = \max_{1\le j\le k^2p}\{|p'_{\lambda_{\mathbf{B},n}}(|\beta_{j0}|)| : \beta_{i0} \neq 0\}$, $a_{\mathbf{\Theta},n} = \max_{1\le j\le k^2, j\neq(r-1)k+r, r\in\{1,\ldots,k\}}\{|p'_{\lambda_{\mathbf{\Theta},n}}(|\eta_{j0}|)| : \eta_{j0} \neq 0\}$, $b_{\mathbf{B},n} = \max_{1\le j\le k^2p}\{|p''_{\lambda_{\mathbf{B},n}}(\beta_{j0}|)| : \beta_{i0} \neq 0\}$ and $b_{\mathbf{\Theta},n} = \max_{1\le j\le k^2, j\neq(r-1)k+r, r\in\{1,\ldots,k\}}\{|p''_{\lambda_{\mathbf{\Theta},n}}(|\eta_{j0}|)| : \eta_{j0} \neq 0\}$, where $p'_{\lambda_{\mathbf{B},n}}(|\beta_{j0}|)$ and $p''_{\lambda_{\mathbf{B},n}}(|\beta_{j0}|)$ are the first and second order of derivatives of the penalty with respect to β_j respectively and $p'_{\lambda_{\mathbf{\Theta},n}}(|\eta_{j0}|)$, $p''_{\lambda_{\mathbf{\Theta},n}}(|\eta_{j0}|)$ are the first and second order of derivatives of the penalty with respect to η_j respectively. Here, the clause $j \neq (r-1)k+r, r \in \{1,\ldots,k\}$ is used because penalty is not applied on the diagonal elements of $\mathbf{\Theta}$. Then the first and second order of derivative vectors are defined as $\boldsymbol{\psi}_{\mathbf{B}}(\boldsymbol{\beta}) = (p'_{\lambda_{\mathbf{B},n}}(|\beta_1|)\mathrm{sgn}(\beta_1),\ldots,p'_{\lambda_{\mathbf{B},n}}(|\beta_{k^2p}|)\mathrm{sgn}(\beta_{k^2p}))^T$, $\mathbf{\Psi}_{\mathbf{B}}(\boldsymbol{\beta}) = \mathrm{diag}\{p''_{\lambda_{\mathbf{B},n}}(|\beta_1|),\ldots,p''_{\lambda_{\mathbf{B},n}}(|\beta_{k^2p}|)\}$ for $\mathbf{B}$ and $\boldsymbol{\psi}_{\mathbf{\Theta}}(\boldsymbol{\eta}) = (\psi_{\mathbf{\Theta},i}(\eta_i))_{i=1,\ldots,k^2}$ and $\mathbf{\Psi}_{\mathbf{\Theta}}(\boldsymbol{\eta}) = \mathrm{diag}\{\Psi_{\mathbf{\Theta},i}(\eta_i)\}_{i=1,\ldots,k^2}$ for $\mathbf{\Theta}$, where

$$\psi_{\mathbf{\Theta},i}(\eta_i) = \begin{cases} 0 & i = (r-1)k+r, r\in\{1,\ldots,k\} \\ p'_{\lambda_{\mathbf{\Theta},n}}(|\eta_i|)\mathrm{sgn}(\eta_i) & \text{otherwise,} \end{cases}$$

and

$$\Psi_{\mathbf{\Theta},i}(\eta_i) = \begin{cases} 0 & i = (r-1)k+r, r\in\{1,\ldots,k\} \\ p''_{\lambda_{\mathbf{\Theta},n}}(|\eta_i|) & \text{otherwise.} \end{cases}$$

We set up variables related to the precision matrix $\mathbf{\Theta}$. Define ${}^n\mathbf{\Theta}_i = \frac{\partial\, {}^n\mathbf{\Theta}}{\partial\eta_i}$ ($k\times k$ matrix) and ${}^n\mathbf{\Theta}_{ij} = \frac{\partial^2\, {}^n\mathbf{\Theta}}{\partial\eta_i\eta_j}$ ($k\times k$ matrix). Let $\mu_1 \le \ldots \le \mu_{k^2}$ be the eigenvalues of ${}^n\mathbf{\Theta}$. Let μ_l^q be the eigenvalues of ${}^n\mathbf{\Theta}_q$ such that $|\mu_1^q| \le \ldots \le |\mu_{k^2}^q|$ and let $\mu_l^{qq'}$ be the eigenvalues of ${}^n\mathbf{\Theta}_{qq'}$ such that $|\mu_1^{qq'}| \le \ldots \le |\mu_{k^2}^{qq'}|$. Define ${}^nt_{qq'} = \mathrm{trace}({}^n\mathbf{\Theta}^{-1}\, {}^n\mathbf{\Theta}_q\, {}^n\mathbf{\Theta}^{-1}\, {}^n\mathbf{\Theta}_{q'})$.

Next, the Frobenius norm, max norm and spectral norm of an matrix $\mathbf{E} = (e_{ij})_{i,j=1,\ldots,T_n}$ are defined as $\|\mathbf{E}\| = (\sum_i^{T_n}\sum_j^{T_n} e_{ij}^2)^{1/2}$, $\|\mathbf{E}\|_{\max} = \max\{|e_{ij}| : i,j = 1,\ldots,T_n\}$ and $\|\mathbf{E}\|_s = max\{|\mu_l| \text{ of } \mathbf{E} : l = 1,\ldots,T_n\}$.

Following Chu et al. (2011a), based on Equation (2.33), we have the following regularity conditions:

(A1) For any $\boldsymbol{\eta} \in \Omega$, where Ω is an open set in $\mathbb{R}^{k^2}$ such that the precision matrix $\boldsymbol{\Theta}$ with $\boldsymbol{\eta} = \text{vec}(\boldsymbol{\Theta})$ is positive definite and twice differentiable with respect to $\boldsymbol{\eta}$. Its second order derivatives is continuous and is positive definite.

(A2) There exist $C, C_q, C_{qq'} \in \mathbb{Z}^+$ such that $\lim_{n\to\infty} \mu_{k_n^2} = C < \infty$, $\lim_{n\to\infty} |\mu^q_{k_n^2}| = C_q < \infty$, $\lim_{n\to\infty} |\mu^{qq'}_{k_n^2}| = C_{qq'} < \infty$ for all $q, q' = 1, \ldots, k_n^2$.

(A3) For some $\delta > 0$, there exist positive constants $D_q, D_{qq'}$ and $D^*_{qq'}$, such that $\|^n\boldsymbol{\Theta}_q\|^{-2} = D_q T_n^{-1/2-\delta}$ for $q = 1, \ldots, k^2$;

(A4) For any $q, q' = 1, \ldots, k^2$, ${}^n e_{qq'} = \lim_{n\to\infty}\{{}^n t_{qq'}({}^n t_{qq}\ {}^n t_{q'q'})^{-1/2}\}$ exists and $\mathbf{E}_n = ({}^n e_{qq'})_{q,q'=1....,k^2}$ is nonsingular;

(A5) The matrix $\mathbf{Z}$ has full rank kp and is uniformly bounded in max norm with $\lim_{n\to\infty}(\mathbf{Z}\mathbf{Z}^T)^{-1} = \mathbf{0}$.

(A6) There exists a $C_0 \in \mathbb{Z}^+$, such that $\|^n\boldsymbol{\Theta}^{-1}\|_s < C_0 < \infty$.

(A7) For $\boldsymbol{\beta} \in \mathbb{R}^{k^2 p}$ and $\boldsymbol{\eta} \in \Omega$, $T_n^{-1}\mathcal{I}_n(\boldsymbol{\beta}) \to \mathbf{J}(\boldsymbol{\beta})$ and $T_n^{-1}\mathcal{I}_n(\boldsymbol{\eta}) \to \mathbf{J}(\boldsymbol{\eta})$, where $\mathcal{I}_n(\cdot)$ is a Fisher information matrix and $\mathbf{J}(\cdot)$ is a Jacobian matrix.

(A8) $a_{\mathbf{B},n} = O(T_n^{-1/2})$, $a_{\boldsymbol{\Theta},n} = O(T_n^{-1/2})$, $b_{\mathbf{B},n} \to 0$, as $n \to \infty$ and $b_{\boldsymbol{\Theta},n} \to 0$ as $n \to \infty$.

(A9) There exist $c_{\mathbf{B},1}, c_{\boldsymbol{\Theta},1}$, $c_{\mathbf{B},2}, c_{\boldsymbol{\Theta},2} \in \mathbb{Z}^+$ such that, $|p''_{\lambda_\mathbf{B},n}(\beta_1) - p''_{\lambda_\mathbf{B},n}(\beta_1)| \leq c_{\mathbf{B},2}|\beta_1 - \beta_2|$ for $\beta_1, \beta_2 > c_{\mathbf{B},1}\lambda_{\mathbf{B},n}$ and $|p''_{\lambda_\boldsymbol{\Theta},n}(\eta_1) - p''_{\lambda_\boldsymbol{\Theta},n}(\eta_1)| \leq c_{\boldsymbol{\Theta},2}|\eta_1 - \eta_2|$ for $\eta_1, \eta_2 > c_{\boldsymbol{\Theta},1}\lambda_{\boldsymbol{\Theta},n}$.

(A10) $\lambda_{\mathbf{B},n} \to 0$, $T_n^{1/2}\lambda_{\mathbf{B},n} \to \infty$, as $n \to \infty$ and $\lambda_{\boldsymbol{\Theta},n} \to 0$, $T_n^{1/2}\lambda_{\boldsymbol{\Theta},n} \to \infty$, as $n \to \infty$.

(A11) $\liminf_{n\to\infty} \liminf_{\beta\to 0^+} \lambda^{-1}_{\mathbf{B},n} p'_{\lambda_\mathbf{B},n}(\beta) > 0$ and $\liminf_{n\to\infty} \liminf_{\eta\to 0^+} \lambda^{-1}_{\boldsymbol{\Theta},n} p'_{\lambda_\mathbf{B},n}(\eta) > 0$.

Note that Conditions (A1) and (A5) are standard assumptions for maximum likelihood estimators and Conditions (A2), (A3), (A4) and (A6) assume the information matrix is smooth and convergent. Conditions (A8) to (A11) refer to mild regularity conditions for the penalty functions, LASSO, SCAD and MCP.

Theorem 2.2. *Under the regularity conditions (A1) to (A9), there exists a local maximizer* ${}^n\widehat{\boldsymbol{\zeta}} = ({}^n\widehat{\boldsymbol{\beta}},\ {}^n\widehat{\boldsymbol{\eta}})$ *of* $Q(\boldsymbol{\beta}, \boldsymbol{\eta})$ *such that* $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\| = O_p(T_n^{-1/2} + a_{\mathbf{B},n})$ *and* $\|\widehat{\boldsymbol{\eta}} - \boldsymbol{\eta}_0\| = O_p(T_n^{-1/2} + a_{\boldsymbol{\Theta},n})$. *Both have the probabilities tending to one. Assume* ${}^n\widehat{\boldsymbol{\beta}} = ({}^n\widehat{\boldsymbol{\beta}}_1, {}^n\widehat{\boldsymbol{\beta}}_2)$ *and* ${}^n\widehat{\boldsymbol{\eta}} = ({}^n\widehat{\boldsymbol{\eta}}_1, {}^n\widehat{\boldsymbol{\eta}}_2)$. *In addition, if (A10) to (A11) hold, we have*

(a) sparsity on ${}^n\widehat{\boldsymbol{\beta}}$ *and* ${}^n\widehat{\boldsymbol{\eta}}$.

(i) ${}^n\widehat{\boldsymbol{\beta}}_2 = \mathbf{0}$ *with probability tending to 1.*

(ii) ${}^n\widehat{\boldsymbol{\eta}}_2 = \mathbf{0}$ *with probability tending to 1.*

(b) limiting normal distributions for ${}^n\widehat{\boldsymbol{\beta}}$ *and* ${}^n\widehat{\boldsymbol{\eta}}$.

(i) $T^{1/2}\{\mathbf{J}(\boldsymbol{\beta}_{10}) + \boldsymbol{\Psi}_{\mathbf{B}}(\boldsymbol{\beta}_{10})\}[\ {}^n\widehat{\boldsymbol{\beta}}_{10} - \boldsymbol{\beta}_{10} + \{\mathbf{J}(\boldsymbol{\beta}_{10}) + \boldsymbol{\Psi}_{\mathbf{B}}(\boldsymbol{\beta}_{10})\}^{-1}\boldsymbol{\psi}_{\mathbf{B}}(\boldsymbol{\beta}_{10})] \xrightarrow{D} N_{k^2p}(\mathbf{0}, \mathbf{J}(\boldsymbol{\beta}_{10}))$,

(ii) $T^{1/2}\{\mathbf{J}(\boldsymbol{\eta}_{10}) + \boldsymbol{\Psi}_{\boldsymbol{\Theta}}(\boldsymbol{\eta}_{10})\}[\ {}^n\widehat{\boldsymbol{\eta}}_{10} - \boldsymbol{\eta}_{10} + \{\mathbf{J}(\boldsymbol{\eta}_{10}) + \boldsymbol{\Psi}_{\boldsymbol{\Theta}}(\boldsymbol{\eta}_{10})\}^{-1}\boldsymbol{\psi}_{\boldsymbol{\Theta}}(\boldsymbol{\eta}_{10})] \xrightarrow{D} N_{k^2}(\mathbf{0}, \mathbf{J}(\boldsymbol{\eta}_{10}))$,

where $\mathbf{J}(\boldsymbol{\beta}_{10})$ *and* $\mathbf{J}(\boldsymbol{\eta}_{10})$ *are the first* $k^2p \times k^2p$ *upper-left matrix and middle* $k^2 \times k^2$ *matrix of* $\mathbf{J}(\boldsymbol{\zeta}_0)$ *respectively.*

Proof: We make use the proof of Theorem 4.1 of the Technical Report of Chu et al. (2011b) for proving our results.

Refer their proof, we replace the following for proving $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\| = O_p(T_n^{-1/2} + a_{\mathbf{B},n})$ and $\|\widehat{\boldsymbol{\eta}} - \boldsymbol{\eta}_0\| = O_p(T_n^{-1/2} + a_{\boldsymbol{\Theta},n})$ with probabilities tending to one:

1. ξ_n with $\xi_{\mathbf{B},n} = T_n^{-1/2} + a_{\mathbf{B},n}$ and $\xi_{\mathbf{\Theta},n} = T_n^{-1/2} + a_{\mathbf{\Theta},n}$;

2. Equation (A.1) with

$$P\left\{\sup_{\|(\mathbf{u_B},\mathbf{u_\Theta})\|=C} Q(\boldsymbol{\beta}_0 + \xi_{\mathbf{B},n}\mathbf{u}_{\mathbf{B},n}, \boldsymbol{\eta}_0 + \xi_{\mathbf{\Theta},n}\mathbf{u}_{\mathbf{\Theta},n}) < Q(\boldsymbol{\beta}_0)\right\} \geq 1 - \epsilon;$$

3. $Q(\boldsymbol{\eta}_0 + \xi_n\mathbf{u}) - Q(\boldsymbol{\eta}_0)$ inequality with

$$\begin{aligned} Q(\boldsymbol{\beta}_0 + \xi_{\mathbf{B},n}\mathbf{u}_{\mathbf{B},n}, \boldsymbol{\eta}_0 + \xi_{\mathbf{\Theta},n}\mathbf{u}_{\mathbf{\Theta},n}) - Q(\boldsymbol{\beta}_0, \boldsymbol{\eta}_0) \\ \leq\ & l(\boldsymbol{\beta}_0 + \xi_{\mathbf{B},n}\mathbf{u}_{\mathbf{B},n}, \boldsymbol{\eta}_0 + \xi_{\mathbf{\Theta},n}\mathbf{u}_{\mathbf{\Theta},n}) - l(\boldsymbol{\beta}_0, \boldsymbol{\eta}_0) \\ & - T_n \sum_{j=1}^{k^2 p} \{p_{\lambda_{\mathbf{B}}}(|\beta_{j0} + \xi_{\mathbf{B},n}u_j|) - p_{\lambda_{\mathbf{B}}}(|\beta_{j0}|)\} \\ & - T_n \sum_{j=1}^{k^2} \{p_{\lambda_{\mathbf{\Theta}}}(|\eta_{j0} + \xi_{\mathbf{\Theta},n}u_j|) - p_{\lambda_{\mathbf{\Theta}}}(|\eta_{j0}|)\} \quad (2.34) \end{aligned}$$

The right hand side of Equation (2.34) will be expanded with Taylor's expansion. Together with their Lemma 1, the results follow.

In order to prove the sparsity of $\widehat{\boldsymbol{\beta}}$, Equation (A.3) in their report is used. We prove the sparsity of $\widehat{\boldsymbol{\eta}}$ by formulating $\frac{\partial Q(\widehat{\beta},\widehat{\eta})}{\partial \eta_j}$ in a similar way as $\frac{\partial Q(\widehat{\beta},\widehat{\eta})}{\partial \beta_j}$ in their report. Then, it is clear that $\frac{\partial Q(\widehat{\beta},\widehat{\eta})}{\partial \eta_j} < 0$ for $\hat{\eta}_j \in (0, \epsilon_n)$ and $\frac{\partial Q(\widehat{\beta},\widehat{\eta})}{\partial \eta_j} > 0$ for $\hat{\eta}_j \in (-\epsilon_n, 0)$ for $\epsilon_n > 0$. As $n \to \infty, \epsilon_n \to 0$. Then, the sparsity results follow.

To show the asymptotic normality of $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\eta}}$, we can define $s_{\mathbf{B}}$ and $s_{\mathbf{\Theta}}$ as the number of non-zero elements in $\mathbf{B}$ and $\mathbf{\Theta}$ and change the following in their proof:

1. $\frac{\partial}{\partial \eta_j} Q(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\eta}}) = -T_n \left\{p'_{\lambda_{\mathbf{\Theta}},n}(|\hat{\eta}_j|)\mathrm{sgn}(|\hat{\eta}_j|) + O_p(T_n^{-1/2})\right\}$

2. $\mathbf{U}_1 = [\mathbf{I}_{s_{\mathbf{B}}}, \mathbf{0}_{s_{\mathbf{B}} \times (k^2(p+1) - s_{\mathbf{B}})}]$

3. $\mathbf{U}_2 = [\mathbf{0}_{s_{\mathbf{\Theta}} \times k^2 p}, \mathbf{I}_{s_{\mathbf{\Theta}}}, \mathbf{0}_{s_{\mathbf{\Theta}} \times (k^2 - s_{\mathbf{\Theta}})}]$

The first order derivatives are set to zero and similarly, apply the Slutsky's theorem, results follow. □

2.3 Simulation Study

2.3.1 Data generation

In this simulation study, we consider the following four sparse stable VAR models:

Model 1: $\mathbf{y}_t = \mathbf{A}_1^{(1)}\mathbf{y}_{t-1} + \mathbf{u}_t$, with $\mathbf{u}_t \sim N(\mathbf{0}, \mathbf{\Sigma}_1)$ with matrices sparsity 0.56.

Model 2: $\mathbf{y}_t = \mathbf{A}_1^{(2)}\mathbf{y}_{t-1} + \mathbf{u}_t$, with $\mathbf{u}_t \sim N(\mathbf{0}, \mathbf{\Sigma}_2)$ with matrices sparsity being 0.72.

Model 3: $\mathbf{y}_t = \mathbf{A}_1^{(3)}\mathbf{y}_{t-1} + \mathbf{A}_2^{(3)}\mathbf{y}_{t-2} + \mathbf{u}_t$, with $\mathbf{u}_t \sim N(\mathbf{0}, \mathbf{\Sigma}_3)$, with matrices sparsity being 0.5.

Model 4: $\mathbf{y}_t = \mathbf{A}_1^{(4)}\mathbf{y}_{t-1} + \mathbf{A}_2^{(4)}\mathbf{y}_{t-2} + \mathbf{u}_t$, with $\mathbf{u}_t \sim N(\mathbf{0}, \mathbf{\Sigma}_4)$, with matrices sparsity being 0.72,

where the coefficients and innovation precision matrices are

$$\mathbf{A}_1^{(1)} = \begin{bmatrix} 0.4352 & -0.6552 & 0.4154 & 0.393 & -0.52 & 0.2256 \\ 0.1478 & -0.4932 & 0 & 0 & 0 & 0 \\ -0.794 & 0 & -0.8933 & 0 & 0 & 0 \\ 0.5894 & 0 & 0 & -0.1478 & 0 & 0 \\ -0.8009 & 0 & 0 & 0 & -0.4169 & 0 \\ 0.4197 & 0 & 0 & 0 & 0 & -0.2439 \end{bmatrix},$$

$$\mathbf{\Sigma}_1^{-1} = \begin{bmatrix} 1 & 0.4 & 0.4 & 0.4 & 0.4 & 0.4 \\ 0.4 & 1 & 0 & 0 & 0 & 0 \\ 0.4 & 0 & 1 & 0 & 0 & 0 \\ 0.4 & 0 & 0 & 1 & 0 & 0 \\ 0.4 & 0 & 0 & 0 & 1 & 0 \\ 0.4 & 0 & 0 & 0 & 0 & 1 \end{bmatrix};$$

$$\mathbf{A}_1^{(2)} = \begin{bmatrix} -0.9665 & -0.8648 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.6593 & 0.0143 & -0.7612 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5644 & 0.0363 & -0.0301 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.5107 & -0.6301 & -0.8983 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0502 & 0.1398 & -0.2802 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.4055 & 0.0806 & 0.6966 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.4182 & -0.6615 & 0.2808 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.2419 & -0.5323 & -0.3481 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4555 & 0.1713 \end{bmatrix},$$

$$\boldsymbol{\Sigma}_2^{-1} = \begin{bmatrix} 1.9934 & -0.2357 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.2357 & 1.7076 & 0.5477 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5477 & 0.8525 & -0.0117 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.0117 & 0.9293 & 0.0713 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0713 & 0.6003 & -0.0125 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.0125 & 0.5386 & 0.0281 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0281 & 1.2644 & 0.1604 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.1604 & 0.7970 & -0.1512 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.1512 & 1.6736 \end{bmatrix};$$

$$\mathbf{A}_1^{(3)} = \begin{bmatrix} -0.6 & 0.4 & 0 & 0 & 0 & 0.4 \\ 0.4 & -0.6 & 0.4 & 0 & 0 & 0 \\ 0 & 0.4 & -0.6 & 0.4 & 0 & 0 \\ 0 & 0 & 0.4 & -0.6 & 0.4 & 0 \\ 0 & 0 & 0 & 0.4 & -0.6 & 0.4 \\ 0.4 & 0 & 0 & 0 & 0.4 & -0.6 \end{bmatrix}, \mathbf{A}_2^{(3)} = \begin{bmatrix} -0.3 & 0.2 & 0 & 0 & 0 & 0.2 \\ 0.2 & -0.3 & 0.2 & 0 & 0 & 0 \\ 0 & 0.2 & -0.3 & 0.2 & 0 & 0 \\ 0 & 0 & 0.2 & -0.3 & 0.2 & 0 \\ 0 & 0 & 0 & 0.2 & -0.3 & 0.2 \\ 0.2 & 0 & 0 & 0 & 0.2 & -0.3 \end{bmatrix},$$

$$\boldsymbol{\Sigma}_3^{-1} = \begin{bmatrix} 1 & -0.3 & 0 & 0 & 0 & -0.3 \\ -0.3 & 1 & -0.3 & 0 & 0 & 0 \\ 0 & -0.3 & 1 & -0.3 & 0 & 0 \\ 0 & 0 & -0.3 & 1 & -0.3 & 0 \\ 0 & 0 & 0 & -0.3 & 1 & -0.3 \\ -0.3 & 0 & 0 & 0 & -0.3 & 1 \end{bmatrix};$$

$$\mathbf{A}_1^{(4)} = \begin{bmatrix} 0.2559 & -0.4582 & 0.2256 & 0.354 & -0.38 & -0.2754 & 0.439 & -0.5911 & 0.2157 \\ 0.3954 & 0.1506 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.401 & 0 & 0.1008 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.3762 & 0 & 0 & 0.2938 & 0 & 0 & 0 & 0 & 0 \\ -0.4588 & 0 & 0 & 0 & -0.8571 & 0 & 0 & 0 & 0 \\ 0.5327 & 0 & 0 & 0 & 0 & -0.211 & 0 & 0 & 0 \\ 0.7749 & 0 & 0 & 0 & 0 & 0 & -0.1765 & 0 & 0 \\ 0.5407 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1244 & 0 \\ 0.6038 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.5406 \end{bmatrix},$$

$$\mathbf{A}_2^{(4)} = \begin{bmatrix} 0.378 & -0.2629 & 0.78 & 0.1052 & 0.3005 & -0.382 & 0.6321 & 0.4034 & -0.3793 \\ 0.7094 & -0.5152 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.8389 & 0 & 0.4518 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.3865 & 0 & 0 & -0.5059 & 0 & 0 & 0 & 0 & 0 \\ 0.9818 & 0 & 0 & 0 & -0.5895 & 0 & 0 & 0 & 0 \\ -0.1593 & 0 & 0 & 0 & 0 & -0.2484 & 0 & 0 & 0 \\ 0.2882 & 0 & 0 & 0 & 0 & 0 & 0.7813 & 0 & 0 \\ -0.8302 & 0 & 0 & 0 & 0 & 0 & 0 & -0.2798 & 0 \\ 0.8814 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.4475 \end{bmatrix} \text{ and}$$

$$\boldsymbol{\Sigma}_4^{-1} = \begin{bmatrix} 2.9388 & -0.3055 & -0.3675 & 0.5713 & 0.3805 & 0.895 & -0.2748 & -0.5905 & -1.7305 \\ -0.3055 & 1.6715 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.3675 & 0 & 3.2985 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5716 & 0 & 0 & 2.3303 & 0 & 0 & 0 & 0 & 0 \\ 0.3805 & 0 & 0 & 0 & 3.8838 & 0 & 0 & 0 & 0 \\ 0.895 & 0 & 0 & 0 & 0 & 1.9733 & 0 & 0 & 0 \\ -0.2748 & 0 & 0 & 0 & 0 & 0 & 4.4195 & 0 & 0 \\ -0.5905 & 0 & 0 & 0 & 0 & 0 & 0 & 3.7525 & 0 \\ -1.7305 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4.866 \end{bmatrix}.$$

Here, $\mathbf{A}_1^{(1)}$ and $\boldsymbol{\Sigma}_1^{-1}$ are 6-dimensional square matrices with non-zero elements on the first row, first column and its diagonal only. So Model 1 is a six-dimensional VAR(1) model with its first node connecting to all other nodes in a mixed graph. Model 2 is a nine-dimensional VAR(1) model having a tridiagonal AR coefficient ma-

trix $\mathbf{A}_1^{(2)}$ and precision matrices $\mathbf{\Sigma}_2^{-1}$. If all nodes are located along a line, all nodes are connected only with their adjacent neighbours in a mixed graph. Model 3 is a 6-dimensional VAR(2) model having $\mathbf{A}_1^{(3)}, \mathbf{A}_2^{(3)}$ and $\mathbf{\Sigma}_3^{-1}$ as Toeplitz and tridiagonal structures, added with non-zero elements at (6,1) and (1,6) entries. Its graphical presentation is very similar to Model 2's, added with the first and last nodes being connected. Model 4 is a nine-dimensional VAR(2) model. $\mathbf{A}_1^{(4)}, \mathbf{A}_2^{(4)}$ and $\mathbf{\Sigma}_4^{-1}$ are 9-dimensional matrices have the same structure pattern as $\mathbf{A}_1^{(4)}$. Therefore, its graphical representation is a lag two extension of Model 1.

Experiments have been conducted using R for Models 1 to 4 with sample sizes (T) of 200, 500 and 2000 over 500 replicates. Their sparse VAR models are obtained from our penalized likelihood estimation method using LASSO, SCAD and MCP penalties, over all combinations of tuning regularization parameters $(\lambda_B, \lambda_\Theta)$ with each parameter being ranged from 0.01 to 1.00 under 100 equal divisions. For each replicate, the final VAR model is selected based on minimum BIC.

2.3.2 Performance evaluation measures

To evaluate the performance of the methods, we consider two types of measures: accuracy and sparsity recognition of the AR coefficients and the innovation precision matrix estimates. The former is evaluated by bias, variance and mean square errors (MSE). Bias measures the total expected absolute deviation from the true values. Variance measures the total squares of deviation from the estimated mean value while mean square errors measure the average squared deviation from the true values. The smaller these measures are, the better the model is. The metrics are defined for the AR coefficient matrices as follows. The former is defined as bias, variance and mean square error (MSE).

1. Bias of the AR coefficient estimates from the true matrix:

$$\text{Bias} = \sum_{l=1}^{p} \sum_{i,j=1}^{K} \left| E\left[(\hat{\mathbf{A}}_l)_{i,j}\right] - (\mathbf{A}_l)_{i,j} \right|$$

2. Variance of the estimate:

$$\text{Variance} = \sum_{l=1}^{p} \sum_{i,j=1}^{K} Var\left((\hat{\mathbf{A}}_l)_{i,j}\right);$$

3. Mean square error (MSE)of the estimates:

$$\text{MSE} = \sum_{l=1}^{p} \sum_{i,j=1}^{K} \left\{ \left[E\left[(\hat{\mathbf{A}}_l)_{i,j}\right] - (\mathbf{A}_l)_{i,j} \right]^2 + Var\left((\hat{\mathbf{A}}_l)_{i,j}\right) \right\},$$

The above metrics for the estimated innovation precision matrix $\hat{\mathbf{\Theta}}$ are defined similarly, by replacing $(\hat{\mathbf{A}}_l)_{i,j}$ with $\hat{\mathbf{\Theta}}_{i,j}$ and $(\mathbf{A}_l)_{i,j}$ with $\mathbf{\Theta}_{i,j}$.

The latter measure is the sparsity recognition metrics. We define the true negative rate (TNR) as the proportion of the number of zero coefficients, which are correctly estimated as zero, and the true positive rate (TPR) as the proportion of non-zero coefficients, which are correctly estimated as non-zero. The higher these two rates are, the better the model is. Here, the mean and standard deviation of the TNR and the TPR over the simulated samples are recorded.

Note that only the upper triangular part of the matrix is estimated because the innovation precision matrix is the inverse of the covariance matrix and is symmetric. Hence the number of parameters in the precision matrix is equal to the number of parameters in the upper triangular part of the matrix. This does not apply to the AR coefficient matrix/matrices, as symmetry is not assumed in AR coefficients matrices.

To visualize the performance of the three penalties, boxplots on deviations of estimated values and the true AR coefficients and innovation precision matrices are plotted. If all median lines (marked in black) lie on the line of zero value for the deviation axis in a boxplot, it indicates no or very little deviation among estimates from their true values. Boxes in cyan colour indicate sparse elements, while red colour indicates non-zero elements in the true matrices. This is useful for sparsity recognition

analysis. The smaller the interquartile range box, the smaller the standard deviation of the estimates.

To determine which penalized estimation method is the best, we count the number of each penalized model being used as a final model for each sample and calculate the proportion. The penalized estimation method with the high proportion is the best. To select a final model among the three penalized models for each sample, we adopt the commonly used variable selection criterion, the minimum Bayesian Information Criterion (BIC).

2.3.3 Results

Tables 2.2, 2.3, 2.4 and 2.5 tabulate the results of accuracy measures and sparsity recognition rates for Models 1 to 4 respectively. Bias, variance and MSE are accuracy performance measures, while true negative rates (TNRs) and true positive rates (TPRs) are the sparsity recognition measures. Plain figures under columns TNR and TPR are mean true negative and positive rates, which are used to measure the sparsity performance. Their corresponding standard deviations are below their respective mean values and are in brackets. The table also gives the mean and standard deviation of the corresponding regularization parameter values, λ_B's and λ_Θ's, for these final models selected by BIC and such lambda pairs are called optimal regularization parameters. Plain and bracketed figures below columns λ_B and λ_Θ are the averages and standard deviations of optimal regularization parameters for the AR coefficient matrix and precision matrix estimation respectively.

Table 2.2 shows that SCAD and MCP have overall good accuracy and sparsity recognition performance for Model 1. LASSO gives the largest bias and MSE for the two AR and precision matrices over $T = 200, 500$ and 2000. SCAD gives slightly smaller bias and MSE values than LASSO. MCP gives the smallest bias and MSE for the AR matrices and the second smallest bias and MSE values for the precision

matrix over $T = 200, 500$ and 2000. Note that this bias value and this MSE value of MCP for the precision matrix are very close to the smallest values. For each penalty, the bias, variance and MSE get smaller as the sample size gets larger. When the sparsity recognition is studied, the TPR for all penalties and sample sizes are very close to 1, while the TNR of MCP are better than that of SCAD and LASSO. The TNR values are greater than 0.8. However, the TNRs of LASSO penalized estimates achieve about 0.6 to 0.7 for all sample sizes.

Table 2.3 illustrates that SCAD and MCP have good performance for Model 2 in general. We look at the accuracy performance first. LASSO gives the largest bias and MSE for the AR coefficient and precision matrices, while MCP gives almost the smallest bias and MSE over the three sample sizes. SCAD gives the bias and MSE values for both matrices in the middle of three penalties for each sample size. Note that the variances of matrices for the three penalties are small and LASSO achieves the smallest. SCAD and MCP give similar values. The sparse recognition measures are TNR and TPR. The TNRs and the TPRs for these three penalties get higher as the sample size increases. MCP always gives the highest TNR for AR coefficient and precision matrices and the TNRs and TPRs of SCAD are ranked in the middle. The TNRs and TPRs of LASSO are the worst among these three penalties. MCP seems to be the best among the three penalties.

Table 2.4 shows that the MCP penalty has the best performance in accuracy and sparsity recognition for Model 3. We discuss the accuracy performance first. The LASSO penalty has the largest bias and MSE for two matrices estimation, while MCP gives the smallest bias and MSE for these three sample sizes. SCAD has values of these two metrics in the middle of the two penalties. The variances among the three penalties are similar. Then, the sparsity recognition performance is investigated. The TPRs of the three penalties are all over 0.95 and are very good in performance. When TNR is discussed, LASSO gives the lowest mean TNR of

Table 2.2: LASSO, SCAD and MCP penalized results for Model 1 (VAR(1)) over 500 replicates. Figures in brackets are the corresponding standard deviations.

T	Penalty	λ_B	AR Coefficient Matrix, B				
			Bias	Variance	MSE	TNR	TPR
200	LASSO	0.0866	0.9970	0.0557	0.1126	0.5906	0.9906
		(0.0266)				(0.1196)	(0.0237)
	SCAD	0.1251	0.4895	0.0577	0.0839	0.7587	0.9849
		(0.0325)				(0.0991)	(0.0290)
	MCP	0.1683	0.3436	0.0571	0.0725	0.8767	0.9721
		(0.0421)				(0.0786)	(0.0403)
500	LASSO	0.0609	0.6650	0.0210	0.0460	0.6040	0.9999
		(0.0173)				(0.1129)	(0.0028)
	SCAD	0.0894	0.2353	0.0251	0.0187	0.8099	0.9991
		(0.0199)				(0.0918)	(0.0074)
	MCP	0.1247	0.1189	0.0160	0.0176	0.9350	0.9991
		(0.0242)				(0.0610)	(0.0074)
2000	LASSO	0.0344	0.3287	0.0046	0.0110	0.6183	1.0000
		(0.009)				(0.1063)	(0.0000)
	SCAD	0.0650	0.0677	0.0033	0.0038	0.9367	1.0000
		(0.0106)				(0.0555)	(0.0000)
	MCP	0.0836	0.0198	0.0028	0.0029	0.9928	1.0000
		(0.0179)				(0.0192)	(0.0000)
T	Penalty	λ_Θ	Precision Matrix, Θ				
			Bias	Variance	MSE	TNR	TPR
200	LASSO	0.0466	1.203	0.1107	0.1739	0.6712	1.0000
		(0.0213)				(0.1677)	(0.0000)
	SCAD	0.0845	0.3365	0.1156	0.1236	0.9886	1.0000
		(0.0231)				(0.0407)	(0.0000)
	MCP	0.1184	0.3885	0.1273	0.1372	0.9758	1.0000
		(0.0546)				(0.0518)	(0.0000)
500	LASSO	0.0323	0.9038	0.0429	0.0797	0.6806	1.0000
		(0.0138)				(0.1604)	(0.0000)
	SCAD	0.0689	0.1333	0.0401	0.0413	0.9938	1.0000
		(0.0274)				(0.0307)	(0.0000)
	MCP	0.0967	0.1650	0.0424	0.0440	0.9864	1.0000
		(0.0623)				(0.0387)	(0.0000)
2000	LASSO	0.0191	0.5468	0.0111	0.0254	0.6772	1.0000
		(0.0075)				(0.1625)	(0.0000)
	SCAD	0.0510	0.0.373	0.0090	0.0091	0.9982	1.0000
		(0.0294)				(0.0133)	(0.0000)
	MCP	0.0849	0.0442	0.0092	0.0093	0.9950	1.0000
		(0.0715)				(0.0236)	(0.0000)

Table 2.3: LASSO, SCAD and MCP penalized results for Model 2 (VAR(1)) over 500 replicates. Figures in brackets are the corresponding standard deviations.

T	Penalty	λ_B	AR Coefficient Matrix, B				
			Bias	Variance	MSE	TNR	TPR
200	LASSO	0.1404	1.9505	0.0727	0.2660	0.7998	0.8060
		(0.0249)				(0.0785)	(0.0577)
	SCAD	0.1637	1.1983	0.0980	0.2052	0.8564	0.7994
		(0.0253)				(0.0635)	(0.0577)
	MCP	0.2127	0.8530	0.1103	0.1666	0.9310	0.7565
		(0.0311)				(0.0434)	(0.0500)
500	LASSO	0.0913	1.3124	0.0306	0.1147	0.7983	0.8620
		(0.0167)				(0.0755)	(0.0524)
	SCAD	0.1225	0.7013	0.0394	0.0777	0.8996	0.8471
		(0.0168)				(0.0490)	(0.0513)
	MCP	0.1599	0.5518	0.0338	0.0632	0.9585	0.7951
		(0.0246)				(0.0320)	(0.0465)
2000	LASSO	0.0487	0.7793	0.0077	0.0325	0.8114	0.9457
		(0.0086)				(0.0715)	(0.0381)
	SCAD	0.0745	0.3472	0.0077	0.0195	0.9420	0.9226
		(0.0122)				(0.0408)	(0.0454)
	MCP	0.0885	0.2653	0.0084	0.0155	0.9694	0.8946
		(0.0162)				(0.0287)	(0.0572)
T	Penalty	λ_Θ	Precision Matrix, Θ				
			Bias	Variance	MSE	TNR	TPR
200	LASSO	0.0909	2.1401	0.1734	0.5506	0.9767	0.5128
		(0.0323)				(0.0382)	(0.0904)
	SCAD	0.1105	1.2825	0.3079	0.4266	0.9816	0.5053
		(0.0338)				(0.0392)	(0.0926)
	MCP	0.1278	1.0413	0.3846	0.4425	0.9738	0.5426
		(0.0606)				(0.0397)	(0.0900)
500	LASSO	0.0544	1.7381	0.0817	0.3182	0.9638	0.6197
		(0.0149)				(0.0469)	(0.1271)
	SCAD	0.0674	0.8015	0.1554	0.2002	0.9769	0.6136
		(0.0290)				(0.0389)	(0.1309)
	MCP	0.0762	0.5269	0.1491	0.1641	0.9812	0.6434
		(0.0300)				(0.0299)	(0.0821)
2000	LASSO	0.0212	0.9069	0.0252	0.0835	0.9151	0.8520
		(0.0041)				(0.0628)	(0.0720)
	SCAD	0.0318	0.2022	0.0273	0.0306	0.9839	0.7902
		(0.0063)				(0.0275)	(0.0577)
	MCP	0.0405	0.1722	0.0280	0.0301	0.9914	0.7619
		(0.0097)				(0.0203)	(0.0483)

both matrices (around 0.5 to 0.7). SCAD gives better TNRs of both matrices than LASSO and MCP gives even better than SCAD. The performance of MCP seems to be the best among the three penalties.

Table 2.5 shows that the MCP penalty has the best accuracy and sparse recognition measures of Model 4. We first examine the accuracy measures. LASSO gives the largest bias, variance and MSE for AR coefficient matrix estimation for three sample sizes. SCAD and MCP give similar accuracy metrics values for the AR coefficient matrix at three sample sizes. But LASSO gives the largest bias and variance, but the smallest MSE for precision matrix estimation for three sample sizes. SCAD and MCP give similar accuracy metrics values for the precision matrix at three sample sizes. SCAD is sometimes better than MCP and MCP is sometimes better than SCAD. When the sample size is 2000, SCAD and MCP give similar accuracy metric values and their MSE values are comparable with the MSE values of LASSO for both matrices. It seems that SCAD and MCP are more accurate in general.

The next task to compare their sparse recognition ability from the TPR and TNR values. LASSO has the smallest TNRs for both matrices among the three penalties. SCAD gives larger TNRs for both matrices than LASSO and MCP has the largest TNR for both matrices. All TPRs for AR coefficient matrices are over 0.9 for the three penalties, but it is not the case for the precision matrix estimation. LASSO has the largest TPR for precision matrix estimates. SCAD and MCP have similar values and they are slightly lower than that of LASSO. Anyway, the TPR of SCAD and MCP for the precision matrix are over 0.7. There is a trade-off between TNR and TPR. When the sample size is 2000, MCP has almost the largest TNR and TPR. When the sample size is 2000, MCP has almost the largest TNR and TPR. MCP has an overall better TNR and TPR for both matrices.

When accuracy and the TNR are prioritized as the most important, the overall performance of the LASSO method is the poorest, while the MCP method is almost

Table 2.4: LASSO, SCAD and MCP penalized results for Model 3 (VAR(2)) over 500 replicates. Figures in brackets are the corresponding standard deviations.

T	Penalty	λ_B	AR Coefficient Matrix, B				
			Bias	Variance	MSE	TNR	TPR
200	LASSO	0.0535	2.6269	0.2303	0.4106	0.5341	0.9849
		(0.0142)				(0.1255)	(0.0237)
	SCAD	0.0937	1.4358	0.3419	0.4003	0.7478	0.9711
		(0.0140)				(0.0921)	(0.0333)
	MCP	0.1381	1.1414	0.3698	0.4113	0.8913	0.9324
		(0.0208)				(0.0573)	(0.0550)
500	LASSO	0.0382	1.7137	0.0834	0.1600	0.5711	0.9999
		(0.0086)				(0.1213)	(0.0012)
	SCAD	0.0762	0.7081	0.1042	0.1206	0.8394	0.9996
		(0.0105)				(0.0730)	(0.0039)
	MCP	0.1035	0.3449	0.0895	0.0933	0.9433	0.9983
		(0.0143)				(0.0450)	(0.0073)
2000	LASSO	0.0231	0.9704	0.0191	0.0440	0.6445	1.0000
		(0.0053)				(0.1137)	(0.0000)
	SCAD	0.0594	0.2571	0.0535	0.0557	0.9167	0.9986
		(0.0097)				(0.1283)	(0.0080)
	MCP	0.0756	0.0394	0.0147	0.0147	0.9984	1.0000
		(0.0102)				(0.0081)	(0.0000)
T	Penalty	λ_Θ	Precision Matrix, Θ				
			Bias	Variance	MSE	TNR	TPR
200	LASSO	0.0294	1.0486	0.1642	0.2132	0.5771	0.9993
		(0.0132)				(0.2217)	-(0.0086)
	SCAD	0.0610	0.4910	0.2068	0.2258	0.9131	0.9964
		(0.0157)				(0.1214)	(0.0196)
	MCP	0.0893	0.5594	0.1892	0.2107	0.9609	0.9909
		(0.0284)				(0.0724)	(0.0305)
500	LASSO	0.0220	0.8488	0.0559	0.0861	0.6220	1.0000
		(0.0088)				(0.1984)	(0.0000)
	SCAD	0.0567	0.2508	0.0556	0.0587	0.9887	1.0000
		(0.0142)				(0.0502)	(0.0000)
	MCP	0.0734	0.2381	0.0540	0.0581	0.9884	1.0000
		(0.0320)				(0.0374)	(0.0000)
2000	LASSO	0.0130	0.5464	0.0135	0.0262	0.6478	1.0000
		(0.0047)				(0.1781)	(0.0000)
	SCAD	0.0484	0.1318	0.0332	0.0345	0.9824	1.0000
		(0.0184)				(0.0577)	(0.0000)
	MCP	0.1051	0.0538	0.0120	0.0123	0.9922	1.0000
		(0.0428)				(0.0284)	(0.0000)

Table 2.5: LASSO, SCAD and MCP penalized results for Model 4 (VAR(2)) over 500 replicates. Figures in brackets are the corresponding standard deviations.

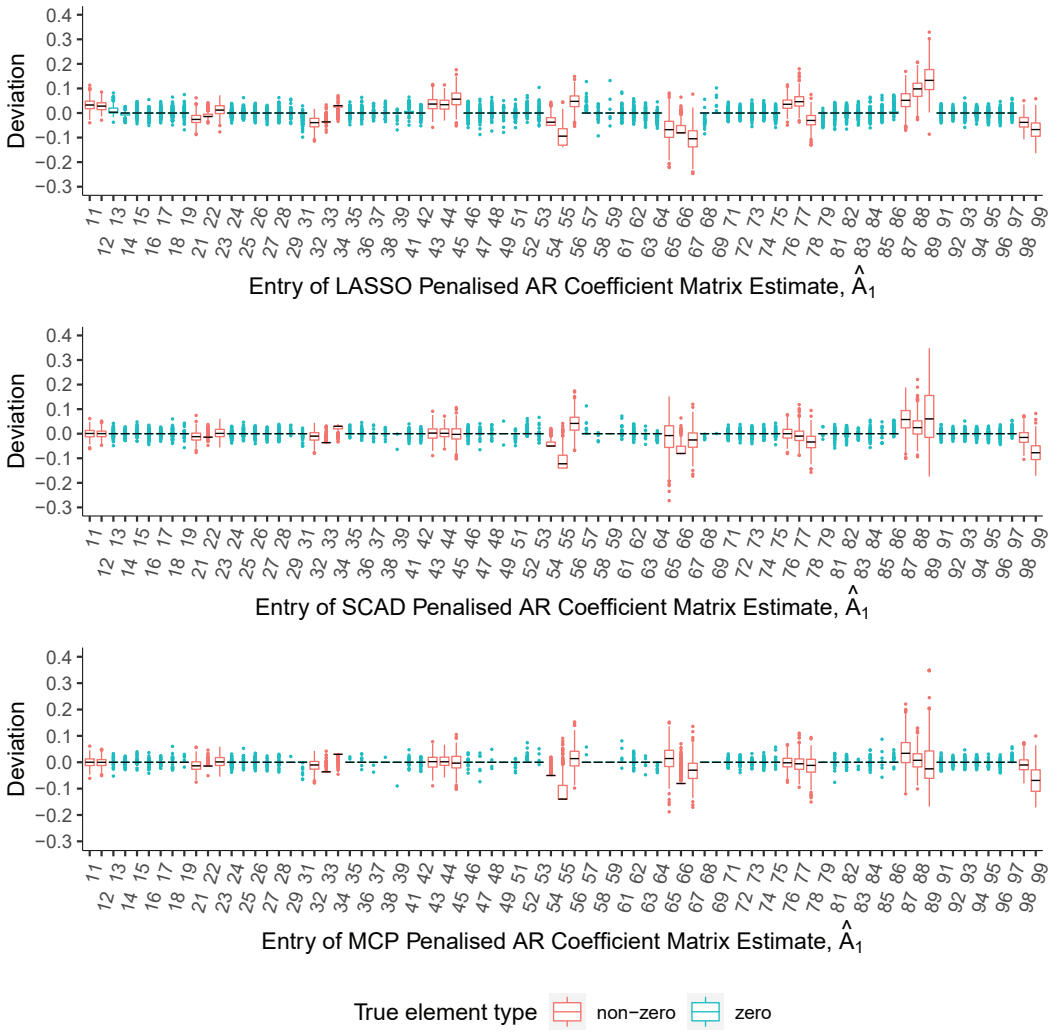
T	Penalty	λ_B	AR Coefficient Matrix, B				
			Bias	Variance	MSE	TNR	TPR
200	LASSO	0.0972	4.8085	0.2204	0.8087	0.5784	0.9652
		(0.0180)				(0.0661)	(0.0268)
	SCAD	0.1344	2.3562	0.3473	0.5036	0.7244	0.9571
		(0.0171)				(0.0485)	(0.0232)
	MCP	0.1987	1.9505	0.3512	0.4690	0.8531	0.9285
		(0.0257)				(0.0405)	(0.0316)
500	LASSO	0.0692	3.1939	0.0833	0.3570	0.5996	0.9950
		(0.0117)				(0.0561)	(0.0102)
	SCAD	0.1091	1.4797	0.1250	0.1910	0.8020	0.9824
		(0.0144)				(0.0391)	(0.0152)
	MCP	0.1535	1.2154	0.0922	0.1550	0.8989	0.9725
		(0.0244)				(0.0340)	(0.0204)
2000	LASSO	0.0393	1.6574	0.0185	0.0944	0.6265	1.0000
		(0.0066)				(0.0521)	(0.0000)
	SCAD	0.0703	0.5383	0.0147	0.0275	0.8758	0.9998
		(0.0080)				(0.0322)	(0.0020)
	MCP	0.1024	0.2975	0.0118	0.0169	0.9544	0.9986
		(0.0112)				(0.0210)	(0.0053)
T	Penalty	λ_Θ	Precision Matrix, Θ				
			Bias	Variance	MSE	TNR	TPR
200	LASSO	0.0224	7.1638	1.8266	4.1718	0.8708	0.8590
		(0.0073)				(0.1018)	(0.1006)
	SCAD	0.0894	4.4435	4.3791	5.1799	0.9311	0.7400
		(0.0485)				(0.0703)	(0.1099)
	MCP	0.1616	3.9958	4.1110	4.7309	0.9428	0.7360
		(0.0885)				(0.0643)	(0.1079)
500	LASSO	0.0128	4.4610	0.7537	1.6629	0.8349	0.9630
		(0.0046)				(0.0956)	(0.0519)
	SCAD	0.0558	1.6709	1.4082	1.5318	0.9481	0.8822
		(0.0243)				(0.0584)	(0.0872)
	MCP	0.1074	1.6303	1.3594	1.4826	0.9525	0.8778
		(0.0552)				(0.0596)	(0.0876)
2000	LASSO	0.0100	3.0324	0.1465	0.6681	0.9199	0.9989
		(0.0000)				(0.0511)	(0.0094)
	SCAD	0.0436	0.3431	0.2139	0.2222	0.9922	0.9803
		(0.0216)				(0.0233)	(0.0352)
	MCP	0.0633	0.3096	0.2223	0.2266	0.9806	0.9946
		(0.0397)				(0.0303)	(0.0202)

the best. The results in Table 2.5 are similar to that in Table 2.3, but MCP gives the smallest bias, variance and MSE in AR and precision matrix estimation in most of the sample cases. Indeed, the TNRs using MCP are higher among the others and the corresponding to TPRs are comparable to those using SCAD. As a result, MCP is the best. This result is consistent with our theoretical performance discussion in Section 2.2.1.

We further examine the entry-wise estimation performance of the AR coefficient and precision matrices for these three penalties. The deviations of each element estimate are presented in boxplots. A red-coloured box plot is used to represent the true non-zero elements while cyan coloured box plot is used to represent the true zero elements. A black horizontal line is used to represent the median. And the box plots are put together to form AR coefficients and innovation matrix estimates deviation box plots. The overall accuracy and sparsity recognition patterns for each penalty are quite similar among these four models. Therefore, we choose a VAR(1) model at a sample size of 500 and VAR(2) model at a sample size of 2000 for illustration and they are presented in Figures 2.2 and 2.3 for Model 2 ($T = 500$) and 4 ($T = 2000$) respectively.

Figure 2.2 shows the box plots for AR coefficient $\mathbf{A}_1$ and the innovation precision matrix $\mathbf{\Theta}$ for Model 2. Figure 2.2 (a) shows the red boxes of LASSO are smaller than that of SCAD and MCP. The cyan outlier spots of zero elements estimated by LASSO fluctuate more from zero than the other two penalized methods. Figure 2.2 (b) has red smaller boxes in the LASSO plot than in the SCAD and the MCP plots, but all black median lines of zero entries lie on zero for these three methods. Its MCP cyan spots lie farther away than LASSO and SCAD.

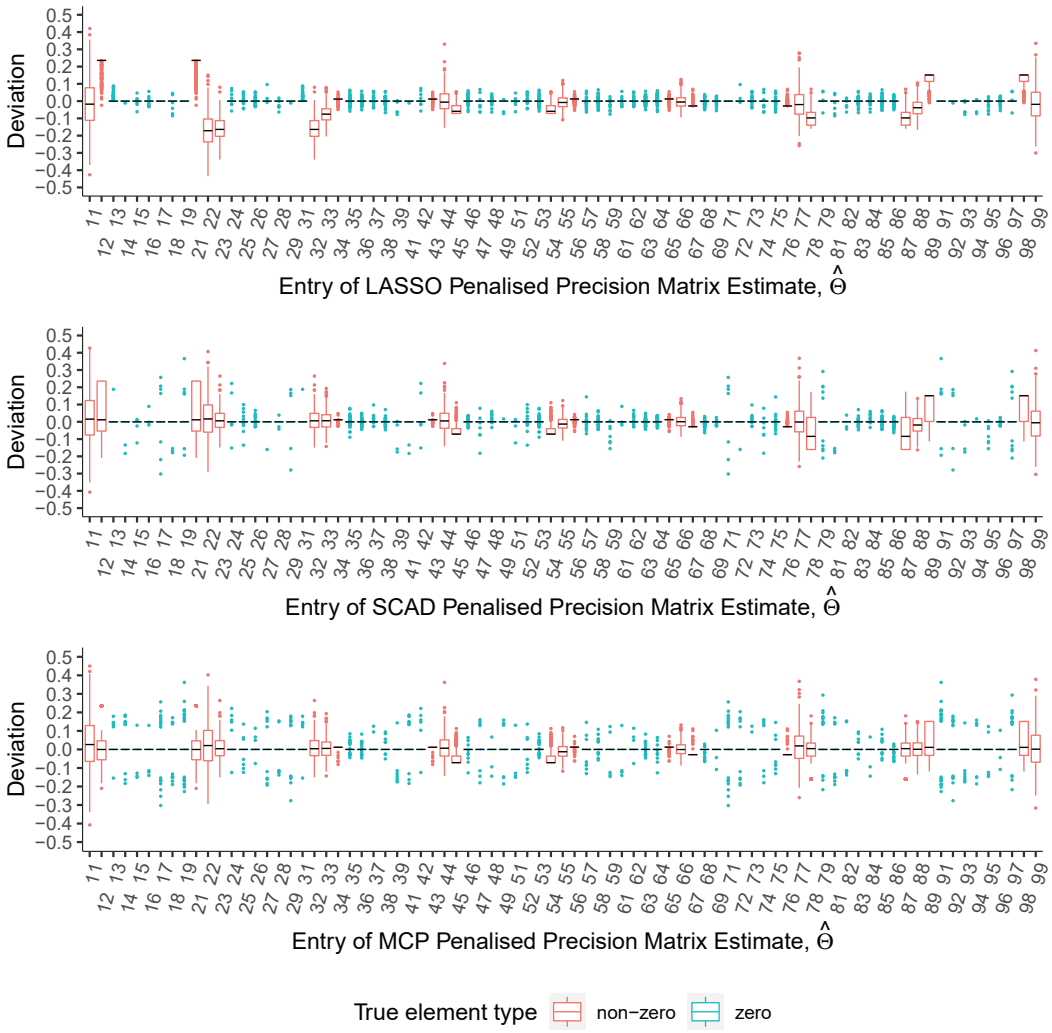
Figure 2.3 shows the box plots for AR coefficients $\mathbf{A}_1$, $\mathbf{A}_2$ and the innovation precision matrix $\mathbf{\Theta}$ for Model 4. Figure 2.3 (a) and (b) show that the red boxes of LASSO deviate more from zero than the SCAD and MCP. SCAD estimates deviate



(a) Deviation boxplot of AR coefficient matrix at lag one for Model 2 at $T = 500$

Figure 2.2: The deviation boxplot of the LASSO, SCAD and MCP penalized estimates from the true values for Model 2 ($T = 500$), first part.

more from zero than the MCP method. The cyan outlier spots of zero elements estimated by LASSO fluctuate more from zero than the other two penalized methods. All black median lines lie on zero for these three methods. Its LASSO cyan spots lie farther away than SCAD and MCP. Figure 2.3 (c) shows that the red boxes in the LASSO plot are smaller than in the SCAD and the MCP plots, and all black median lines of zero entries lie on zero for these three methods. Its MCP cyan spots



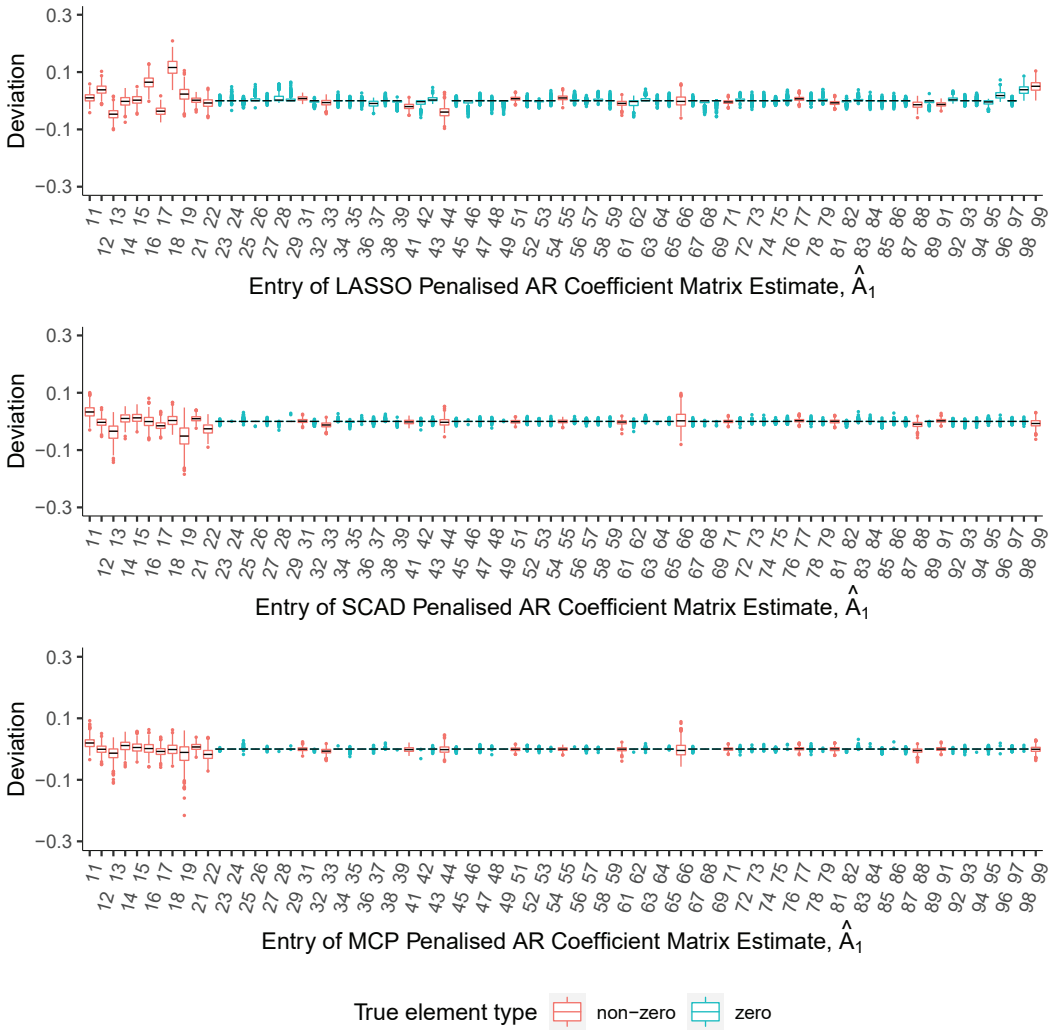
(b) Deviation boxplot of precision matrix for Model 2 at $T = 500$

Figure 2.2: The deviation boxplot of the LASSO, SCAD and MCP penalized estimates from the true values for Model 2 ($T = 500$), second part.

lie farther away than LASSO and SCAD.

All in all, the boxplots of the AR coefficients and precision matrices show that the MCP penalized estimation seems to be the best among these penalties in general.

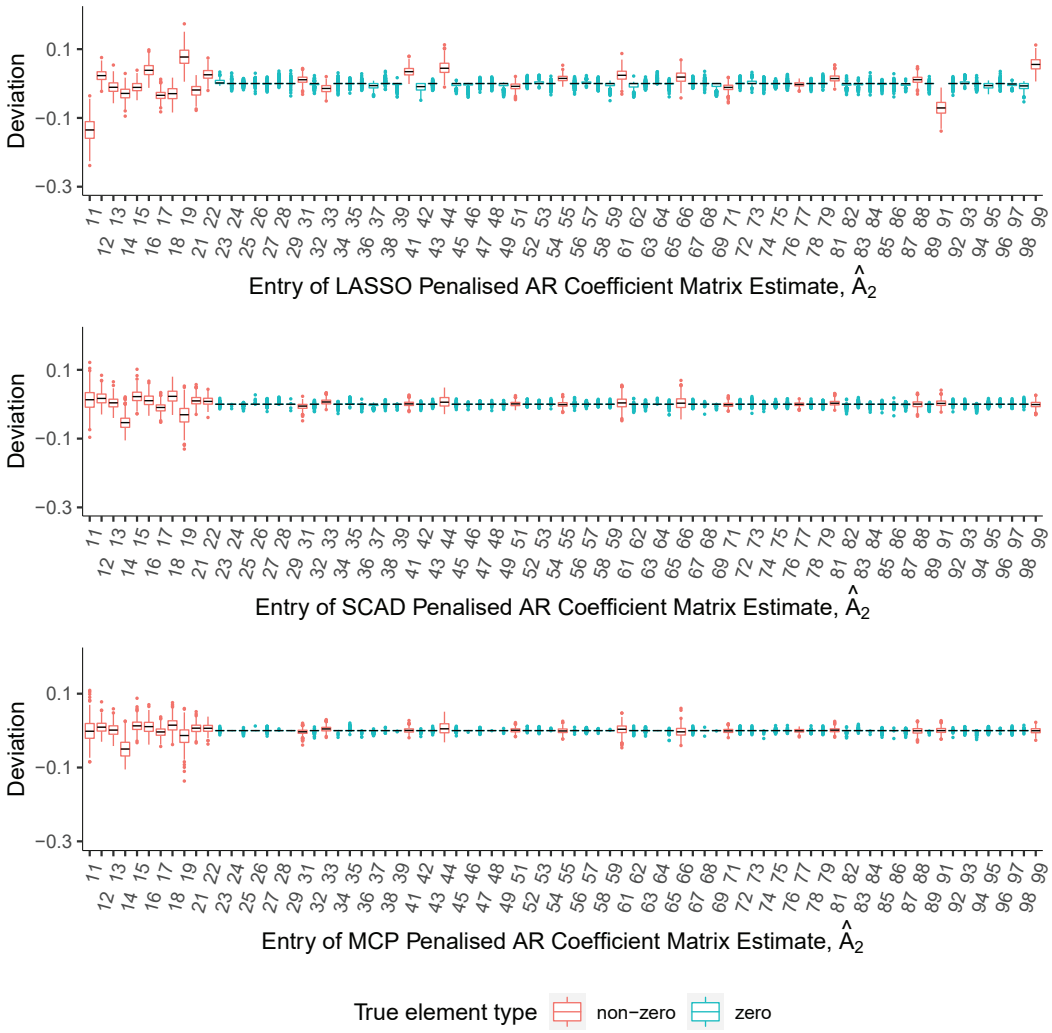
To confirm that the MCP penalized estimation method is the best, we adopt the BIC for model selection. For each sample, we choose the model with minimum BIC from the LASSO, SCAD and MCP models. We count the number of models



(a) Deviation boxplot of AR coefficient matrix at lag one for Model 4 at $T = 2000$

Figure 2.3: The deviation boxplot of the LASSO, SCAD and MCP penalized estimates from the true values for Model 4 ($T = 2000$), first part

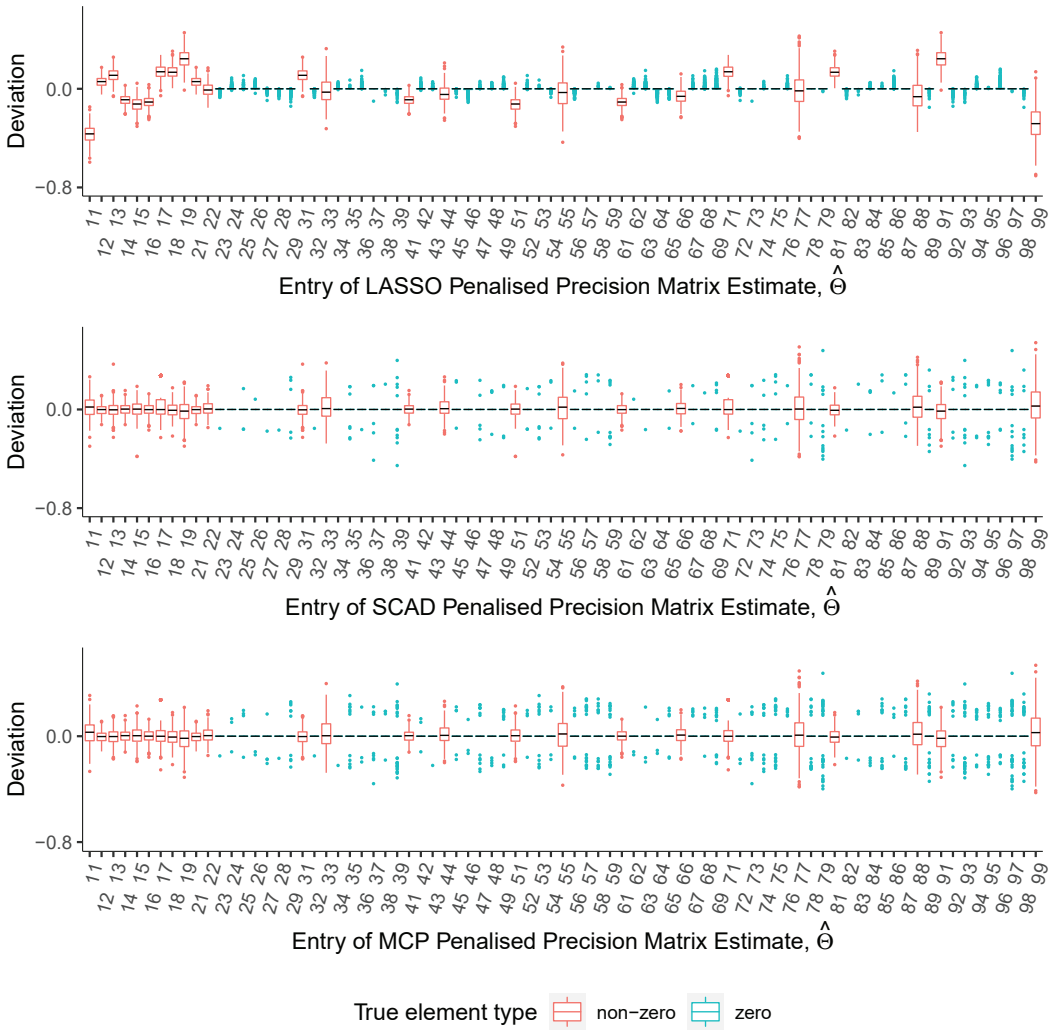
chosen as the final model to calculate the proportion of models chosen as the final model for each penalty. The results are tabulated in Tables 2.6 to 2.9 for Models 1 to 4 respectively. All models have minimum mean BICs using the MCP penalization method over all the sample sizes $T = 200, 500$ and 2000. Table 2.6 shows that 100% MCP sparse models with lengths $T = 200$ and 500 have minimum BIC, when compared with the other two penalties and are chosen as the final model for Model 1.



(b) Deviation boxplot of AR coefficient matrix at lag two for Model 4 at $T = 2000$

Figure 2.3: The deviation boxplot of the LASSO, SCAD and MCP penalized estimates from the true values for Model 4 ($T = 2000$), second part

For the case of $T = 2000$, over 99% of MCP are chosen based on minimum BIC as final models. In Tables 2.7 to 2.9, all MCP penalized sparse models are similarly selected as final models for Models 2 to 4 at lengths $T = 200, 500$ and 2000. Therefore, MCP is the best penalized method.



(c) Deviation boxplot of precision matrix for Model 4 at $T = 2000$

Figure 2.3: The deviation boxplot of the LASSO, SCAD and MCP penalized estimates from the true values for Model 4 ($T = 2000$), third part

2.3.4 Robustness Test

To test the robustness of the MCP penalty, we used the LASSO penalty as a benchmark for comparison, because the LASSO penalty is well known for its robustness against large variance. The performance evaluation measures described in Section 2.3.2 are used and we compare the pattern of the results across the change to the precision matrices.

Table 2.6: Bayesian Information Criterion values for LASSO, SCAD and MCP penalized results for Model 1 (VAR(1)) over 500 replicates.

		BIC		% of models
T	Penalty	mean	sd	having min BIC
200	LASSO	3899.36	51.99	0
	SCAD	3862.09	50.69	0
	MCP	3849.46	50.88	100
500	LASSO	9525.25	76.19	0
	SCAD	9475.27	75.94	0
	MCP	9460.63	75.57	100
2000	LASSO	37532.86	151.79	0
	SCAD	37451.75	151.46	0.4
	MCP	37443.42	151.30	99.6

Table 2.7: Bayesian Information Criterion values for LASSO, SCAD and MCP penalized results for Model 2 (VAR(1)) over 500 replicates.

		BIC		% of models
T	Penalty	mean	sd	having min BIC
200	LASSO	5334.09	62.10	0
	SCAD	5288.33	61.92	0
	MCP	5253.74	61.42	100
500	LASSO	13040.50	89.05	0
	SCAD	12966.78	89.38	0
	MCP	12929.92	89.25	100
2000	LASSO	51344.20	189.00	0
	SCAD	51224.87	189.87	0
	MCP	51193.74	189.20	100

Two VAR(1) models with sample sizes 200, 500 and 2000 and with the sparsity rates in AR coefficients being 0.56 and in the precision matrix being 0.48 are simulated for 200 replicates. They have the same AR coefficient matrix but their precision matrices are different. The precision matrix of Model 5 is one-fifth of the precision matrix of Model 6. That is, the covariance matrix of Model 6 is larger. Their details are the following:

Model 5: $\mathbf{y}_t = \mathbf{A}_1^{(5)}\mathbf{y}_{t-1} + \mathbf{u}_t$ with $\mathbf{u}_t \sim N(\mathbf{0}, \mathbf{\Sigma}_5)$,

Table 2.8: Bayesian Information Criterion values for LASSO, SCAD and MCP penalized results for Model 3 (VAR(2)) over 500 replicates.

		BIC		% of models
T	Penalty	mean	sd	having min BIC
200	LASSO	3843.74	52.66	0
	SCAD	3779.09	52.47	0
	MCP	3747.44	51.50	100
500	LASSO	9204.86	77.00	0
	SCAD	9112.98	76.37	0
	MCP	9086.50	76.06	100
2000	LASSO	35784.50	150.94	0
	SCAD	35774.85	374.45	0
	MCP	35632.64	151.15	100

Table 2.9: Bayesian Information Criterion values for LASSO, SCAD and MCP penalized results for Model 4 (VAR(1)) over 500 replicates.

		BIC		% of models
T	Penalty	mean	sd	having min BIC
200	LASSO	3777.33	64.75	0
	SCAD	3641.24	65.47	0
	MCP	3572.23	65.02	100
500	LASSO	8735.02	94.60	0
	SCAD	8538.57	91.96	0
	MCP	8477.18	90.48	100
2000	LASSO	33103.55	187.96	0
	SCAD	32806.50	186.04	0
	MCP	32742.62	188.09	100

Model 6: $\mathbf{y}_t = \mathbf{A}_1^{(5)}\mathbf{y}_{t-1} + \mathbf{u}_t$ with $\mathbf{u}_t \sim N(\mathbf{0}, \mathbf{\Sigma}_6)$,

where the coefficient and innovation precision matrices are

$$\mathbf{A}_1^{(5)} = \begin{bmatrix} 0.4352 & -0.6552 & 0.4154 & 0.3930 & -0.5200 & 0.2256 \\ 0.1478 & -0.4932 & 0 & 0 & 0 & 0 \\ -0.7940 & 0 & -0.8933 & 0 & 0 & 0 \\ 0.5894 & 0 & 0 & -0.1478 & 0 & 0 \\ -0.8009 & 0 & 0 & 0 & -0.4169 & 0 \\ 0.4197 & 0 & 0 & 0 & 0 & -0.2439 \end{bmatrix},$$

$$\boldsymbol{\Sigma}_5^{-1} = \begin{bmatrix} 0.05 & 0.02 & 0.02 & 0.02 & 0.02 & 0.02 \\ 0.02 & 0.05 & 0 & 0 & 0 & 0 \\ 0.02 & 0 & 0.05 & 0 & 0 & 0 \\ 0.02 & 0 & 0 & 0.05 & 0 & 0 \\ 0.02 & 0 & 0 & 0 & 0.05 & 0 \\ 0.02 & 0 & 0 & 0 & 0 & 0.05 \end{bmatrix} \quad \text{and}$$

$$\boldsymbol{\Sigma}_6^{-1} = \begin{bmatrix} 0.01 & 0.004 & 0.004 & 0.004 & 0.004 & 0.004 \\ 0.004 & 0.01 & 0 & 0 & 0 & 0 \\ 0.004 & 0 & 0.01 & 0 & 0 & 0 \\ 0.004 & 0 & 0 & 0.01 & 0 & 0 \\ 0.004 & 0 & 0 & 0 & 0.01 & 0 \\ 0.004 & 0 & 0 & 0 & 0 & 0.01 \end{bmatrix}.$$

The accuracy performance and sparsity recognition performance of Model 5 are reported in Table 2.10. The bias, variance and MSE of the AR coefficient matrix, $\mathbf{B}$ of MCP are much lower than that of the LASSO method and these three accuracy measures are very similar in the precision matrix estimation. The TPR of the AR coefficient matrix using both methods are extremely high, close to one, while the TNR of the MCP method is much higher than that of the LASSO method. The TNR of MCP are at least 0.88, but the LASSO method results in around 0.6 in 3 different sample sizes. When the performance of the precision matrix is examined, bias, variance and MSE of the two methods are similar and close to zero. Both methods exhibit similar TNR and TPR.

The accuracy performance and sparsity recognition performance of Model 6 are reported in Table 2.11. The bias, variance and MSE of the AR coefficient matrix, $\mathbf{B}$ of MCP are much lower than that of the LASSO method and a similar accuracy performance is found in both LASSO and MCP methods. The TPR of the AR coefficient matrix using both methods are extremely high, close to one, while the TNR of the MCP method is much higher than that of the LASSO method. The TNR of MCP are at least 0.86, but the LASSO method results in around 0.6 in 3

Table 2.10: LASSO and MCP penalized results for Model 5 (VAR(1)). Figures in brackets are the corresponding standard deviations.

T	Penalty	λ_B	AR Coefficient Matrix, $\mathbf{B}$				
			Bias	Variance	MSE	TPR	TNR
200	LASSO	0.0901	0.9960	0.0515	0.1084	0.5990	0.9906
		(0.0263)				(0.1095)	(0.0241)
	MCP	0.1709	0.3838	0.0546	0.0730	0.8825	0.9706
		(0.0432)				(0.0763)	(0.0381)
500	LASSO	0.0638	0.6748	0.0198	0.0455	0.6145	0.9997
		(0.0176)				(0.1140)	(0.0044)
	MCP	0.1230	0.1089	0.0154	0.0170	0.9388	0.9988
		(0.0227)				(0.0611)	(0.0088)
2000	LASSO	0.0363	0.3306	0.0044	0.0110	0.6363	1.0000
		(0.0092)				(0.1132)	(0.0000)
	MCP	0.0831	0.0183	0.0027	0.0028	0.9950	1.0000
		(0.0170)				(0.0159)	(0.0000)
T	Penalty	λ_Θ	Precision Matrix, $\mathbf{\Theta}$				
			Bias	Variance	MSE	TNR	TPR
200	LASSO	0.6535	0.0490	0.0003	0.0004	0.6090	1.0000
		(0.2203)				(0.1654)	(0.0000)
	MCP	0.6562	0.0472	0.0003	0.0004	0.6245	1.0000
		(0.2048)				(0.1621)	(0.0000)
500	LASSO	0.5300	0.0435	0.0001	0.0002	0.6465	1.0000
		(0.2164)				(0.1600)	(0.0000)
	MCP	0.5565	0.0415	0.0001	0.0002	0.6670	1.0000
		(0.2094)				(0.1467)	(0.0000)
2000	LASSO	0.3182	0.0244	0.0000	0.0001	0.6920	1.0000
		(0.1409)				(0.1535)	(0.0000)
	MCP	0.3075	0.0226	0.0000	0.0000	0.7070	1.0000
		(0.1326)				(0.1529)	(0.0000)

different sample sizes. When the performance of the precision matrix is examined, bias, variance and MSE of the two methods are similar and close to zero. Both methods exhibit similar TNR and TPR.

Next, we compare the difference in accuracy and sparsity recognition performance of Model 5 from Model 6.

Table 2.12 tabulates the difference in the accuracy performance of Model 5 from Model 6. The accuracy measures, bias, variance and MSE of the AR coefficients of Model 6 are expected to be larger than that of Model 5 because Model 6 is a model with diagonal variances about 5 times (larger than) the corresponding variance

Table 2.11: LASSO and MCP penalized results for Model 6 (VAR(1)). Figures in brackets are the corresponding standard deviations.

T	Penalty	λ_B	AR Coefficient Matrix, $\mathbf{B}$				
			Bias	Variance	MSE	TNR	TPR
200	LASSO	0.0954	1.0534	0.0548	0.1161	0.5953	0.9906
		(0.0318)				(0.1177)	(0.0241)
	MCP	0.1723	0.3881	0.0555	0.0755	0.8693	0.9725
		(0.0455)				(0.0834)	(0.0369)
500	LASSO	0.0626	0.6500	0.0209	0.0443	0.5940	1.0000
		(0.0205)				(0.1173)	(0.0000)
	MCP	0.1239	0.1221	0.0146	0.0161	0.9338	0.9991
		(0.0231)				(0.0640)	(0.0076)
2000	LASSO	0.0348	0.3270	0.0045	0.0107	0.6180	1.0000
		(0.0097)				(0.1102)	(0.0000)
	MCP	0.0874	0.0223	0.0027	0.0028	0.9935	1.0000
		(0.0178)				(0.0196)	(0.0000)
T	Penalty	λ_Θ	Precision Matrix, $\mathbf{\Theta}$				
			Bias	Variance	MSE	TNR	TPR
200	LASSO	0.6502	0.0045	0.0000	0.0000	0.2310	1.0000
		(0.3012)				(0.1324)	(0.0000)
	MCP	0.6633	0.0043	0.0000	0.0000	0.2365	1.0000
		(0.3049)				(0.1342)	(0.0000)
500	LASSO	0.7391	0.0042	0.0000	0.0000	0.3315	1.0000
		(0.2310)				(0.1479)	(0.0000)
	MCP	0.7564	0.0038	0.0000	0.0000	0.3385	1.0000
		(0.2178)				(0.1420)	(0.0000)
2000	LASSO	0.7439	0.0033	0.0000	0.0000	0.5405	1.0000
		(0.2004)				(0.1654)	(0.0000)
	MCP	0.7487	0.0032	0.0000	0.0000	0.5490	1.0000
		(0.2047)				(0.1692)	(0.0000)

of Model 5 and it would be expected that it is more difficult to estimate the AR coefficients in Model 6. Therefore, we expect the differences of bias, variance and MSE in the AR coefficient matrix are negative for most of the cases. Except for the difference of variance and MSE of MCP at 500 sample sizes, MSE of LASSO at 500 sample sizes and MSE of LASSO at 2000 samples in the AR coefficient matrix, all differences in the accuracy measures are negative or zero. The differences in variance and MSE in the AR coefficients and precision matrices are either a tiny positive value or zero for both the LASSO and the MCP methods. Differences in variance and MSE of the two methods in both matrices is very similar and differences in the bias of

MCP are smaller than that of LASSO. It indicates that MCP is slightly better in LASSO in accuracy performance.

Table 2.12: LASSO and MCP penalized results difference of Model 5 from Model 6

T	Penalty	Difference of AR Coefficient Matrix, $\mathbf{B}$				
		Bias	Variance	MSE	TNR	TPR
200	LASSO	-0.0574	-0.0032	-0.0078	0.0038	0.0000
	MCP	-0.0044	-0.0009	-0.0025	0.0133	-0.0019
500	LASSO	0.0248	-0.0011	0.0012	0.0205	-0.0003
	MCP	-0.0132	0.0008	0.0008	0.0050	-0.0003
2000	LASSO	0.0036	-0.0001	0.0002	0.0183	0.0000
	MCP	-0.0040	0.0000	0.0000	0.0015	0.0000
T	Penalty	Difference of Precision Matrix, $\mathbf{\Theta}$				
		Bias	Variance	MSE	TNR	TPR
200	LASSO	0.0444	0.0003	0.0004	0.3780	0.0000
	MCP	0.0429	0.0003	0.0004	0.3880	0.0000
500	LASSO	0.0393	0.0001	0.0002	0.3150	0.0000
	MCP	0.0377	0.0001	0.0002	0.3285	0.0000
2000	LASSO	0.0211	0.0000	0.0001	0.1515	0.0000
	MCP	0.0193	0.0000	0.0000	0.1580	0.0000

Table 2.12 also gives the difference in sparsity recognition performance between Model 5 and Model 6. The TPR differences of LASSO and MCP are close to zero, but the TNR differences of them are negative. When the sparsity measures are discussed, it would expect that the zero true values of the model with larger variances have higher chances to be estimated as non-zero, i.e. Model 6 is expected to have a lower TNR. That implies that the TNR difference between Model 5 and Model 6 would be positive. This pattern is observed in the TNR of the AR coefficients and precision matrices for the three selected sample sizes. On the contrary, the non-zero small values of the model with larger variances have higher chances of being estimated to be zero. It is expected that the TPR of that model is smaller. This pattern is also found in the TPR for Model 6 (model with larger variance) and hence their difference is negative. Therefore, the phenomena are expected. Most of the TNR differences of MCP are smaller than that of LASSO, while all TPR differences of LASSO and

MCP are close to zero. It indicates that MCP is superior in sparsity recognition.

When the minimum BIC criterion is used, the MCP method is always the best, because all final models are MCP models, as given in Tables 2.13 and 2.14. This example empirically confirms the robustness of the MCP method and is consistent with our theoretical discussion given in Section 2.2.1.

Patterns of accuracy measures and sparsity recognition rates of the MCP method across two variances are similar to that of the LASSO method. In addition, the MCP estimates are always chosen based on BIC. This indicates that the MCP method performs slightly better than that of the LASSO in the robustness test.

Table 2.13: Bayesian Information Criterion values for LASSO and MCP penalized results for Model 5 (VAR(1))

		BIC		% of model
T	Penalty	mean	sd	having min BIC
200	LASSO	7492.28	50.91	0.00
	MCP	7459.82	50.10	100.00
500	LASSO	18512.78	72.95	0.00
	MCP	18468.44	73.38	100.00
2000	LASSO	73467.11	159.46	0.00
	MCP	73405.30	158.69	100.00

Table 2.14: Bayesian Information Criterion values for LASSO and MCP penalized results for Model 6 (VAR(1))

		BIC		% of model
T	Penalty	mean	sd	having min BIC
200	LASSO	9438.36	49.12	0.00
	MCP	9406.75	49.67	100.00
500	LASSO	23354.49	73.81	0.00
	MCP	23309.50	72.65	100.00
2000	LASSO	92785.67	159.20	0.00
	MCP	92725.01	158.98	100.00

2.4 Application

In this section, we use Pearl River air pollution data to demonstrate the use of our proposed model and compare the result with existing sparse time series models.

2.4.1 Pearl River air pollution data

We applied the proposed penalized estimation method to a respirable suspended particles (RSP) time series in Pearl River Delta Region (PRDR) [1]. Seven data series for locations, Chengzhong, Donghu, Luhu, Tianhu, Tanija, Tap Mun and Xiapu, from January 2006 to December 2015, were transformed, detrended and deseasonalized in the same setting as in Yuen et al. (2018). LASSO, SCAD and MCP penalties were used and again, BIC was used to choose the final model. Finally, we compared our proposed models with existing sparse VAR models in the literature.

2.4.2 Results

We fit the data with VAR models up to order 4 by LASSO, SCAD and MCP penalized estimation and calculated their BIC values, tabulated in Table 2.15. The final LASSO penalized graphical VAR model estimated is a lag one model, which has the BIC value of 1365.7. The final SCAD penalized estimated model is also a lag one model with BIC value of 1341 and the final MCP penalized estimated model is a lag two model with BIC of 1301. Its BIC value is the minimum among the three final models. Therefore, the final MCP penalized estimated sparse graphical VAR(2) model is selected as the final model for analysis and interpretation.

The final MCP sparse graphical VAR(2) model is estimated with the regularization parameters $\lambda_{\mathbf{B}} = 0.21$ and $\mathbf{\Theta} = 0.09$. Its AR coefficients matrices and partial correlation matrix, determined by precision matrix, are visualized by the heatmaps

[1] http://www.epd.gov.hk/epd/english/resources/pub/publications/m_report.html.

Table 2.15: BIC values of penalized estimated sparse graphical VAR processes for RSP time series in Pearl River Delta Region.

	BIC for Penalties		
Graphical VAR model order	LASSO	SCAD	MCP
1	1365.7	1341.0	1301.7
2	1392.2	1354.5	1301.0
3	1416.8	1374.1	1312.9
4	1412.6	1404.0	1311.4

in Figure 2.4 and a mixed graph is plotted in Figure 2.5. The directed and undirected components are the temporal causal graph and the conditional independence graph; and they are separately plotted in Figure 2.6. The model consists of 14 and 7 significant AR lagged one and AR lagged two coefficients respectively and generates 21 pairs of cities pollutant causal lagged relationships. Four pairs of cities have both lags one and two relationships. Details refer to Figure 2.6 (a). In addition, it has 13 significant partial correlations. Details refer to Figure 2.6 (b).

2.4.3 Comparison with existing sparse time series models

We compare the MCP penalized sparse graphical VAR (sGVAR) model with existing sparse models in the literature. They are a structural VAR (SVAR) model, a 2-stage sparse VAR (2sVAR) model (Davis et al. (2016)) and two constrained graphical sparse VAR (CGsVAR) models in Yuen et al. (2018). The sparse information and BIC values are tabulated in Table 2.16. Despite the higher VAR order of the fitted sGVAR model, it gives the minimum BIC and would be selected as the final model for analysis. We compare the graphical representation of the models to find the difference between the models.

The most traditional sparse model is the SVAR model. It has almost the same model structure as the VAR model except for two differences. It assumes dependence between lag 0 components, but the VAR model does not. In addition, it assumes

independence between the innovations but the VAR model assumes a general covariance structure of innovations. That is, the SVAR model possesses one more sparse AR matrix with ones along diagonal at lag 0 and its innovation precision matrix has only 7 non-zero parameters. The SVAR model for RSP data gave altogether 26 temporal and lagged causal relationships and it needs 33 parameters in lags 0 and 1 autoregressive (AR) coefficient matrices. Its graphical representation is given in Figure 22 in Yuen et al. (2018). The total number of elements estimated as non-sparse is close to the sGVAR model. Since the SVAR model gives a directed acyclic graph (DAG) and the proposed sGVAR model gives a mixed graph, they could not be compared directly. The Moralization theorem was applied to the DAG graph of the SVAR model to form an undirected graph and it is similar to the undirected graph of the sGVAR model. It contains all edges in the moralized graph of the SVAR model, but the SVAR model does not contain lag one relationship from Tianhu to Tanjia and Tianhu to Chengzhong and lag 2 causal relationships in the sGVAR model.

The 2sVAR model captures 12 AR lagged coefficients respectively, as given in Figure 23(a) and (b) in Yuen et al. (2018). 11 of these lagged coefficients are common with the SVAR model and their values are quite similar. The main discrepancies are that the 2sVAR model does not have the directed edges from Tianhu to Donghu and to Tanjia; and from Tanjia to Xiapu and lagged two directed edges in the sGVAR model.

When the innovation precision matrix of the 2sVAR model is examined, 10 entries do not have significant values at the 5% level. These differences might cause a larger BIC value of the 2sVAR model. Thus, the 2sVAR model has not achieved the best sparsity pattern.

The last two models for comparison are the CGsVAR models (Yuen et al. (2018)) using frequency and time domain approaches for estimation and their AR coefficients and partial correlation matrices are shown in Figure 20(a) and (c) in Yuen et al.

(2018). The sparsity structure of AR coefficients is assumed same as the insignificant conditional partial correlations pattern.

Their estimated AR and precision matrices contain insignificant values and need further fine-tuning. But this phenomenon is not found in the sGVAR model. The frequency and time domain CGsVAR models consist of 13 and 11 statistically significant edges in the lagged one coefficients and have less directed edges, when compared with the sGVAR model. Nine and ten of them in these two models are common with the sGVAR model respectively. The sGVAR model has 5 to 6 more conditional dependencies in the partial correlation matrix than these two models (Figure 20(b) and (d) in Yuen et al. (2018)). The missing conditional dependence in the partial correlation matrix may account for the larger BIC values of these two CGsVAR models.

In general, the proposed sGVAR model can capture a mild higher lagged correlated relationship. It has not only the best sparsity pattern of AR coefficients and precision matrices, chosen by the BIC, but also the minimum BIC among the various sparse VAR models. From the above discussion, we notice that the penalization applied to both AR and precision matrices in estimation allows better flexibility in modelling sparsity structure and thus, improves the model performance.

2.5 Conclusion

We have considered a new sparse graphical time series model, which combines all possible sparse Gaussian graphical models and sparse vector autoregressive models and selects the combined one with the optimal sparsity combination of AR coefficients and precision matrices to be determined by minimum BIC. It avoids the sparse structure affected by any pre-estimates based on AR coefficients, partial correlations or spectral coherence. No sharing of sparsity assumption between AR coefficient and

Table 2.16: BIC values of different sparse models for Pearl River Delta Region.

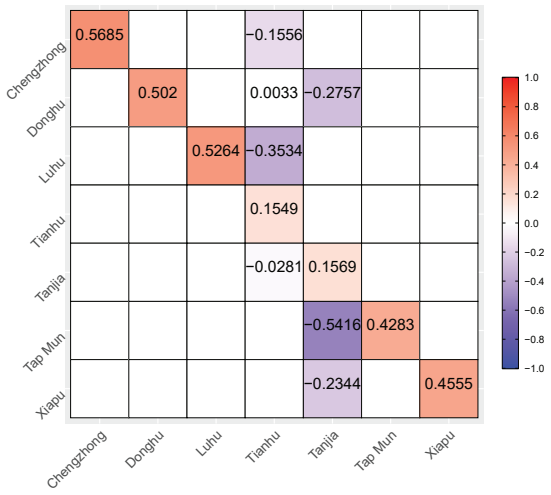
Sparse models	VAR Order	No. of non-sparse cells in matrices			BIC values
		AR Coefficients	Precision[1]	Total	
Structural VAR model	1	33*	7*	40	1380.9
2-stage sparse VAR model	1	12	28#	40	1334.1
CGsVAR model[2] (Freq Domain)	1	27	17	44	1365.8
CGsVAR model[2] (Time Domain)	1	25	16	41	1354.8
MCP penalized sGVAR model	2	21	20	41	1301.0

[1] Precision matrix is symmetric and therefore the maximum number of parameters required is 28.
[2] CGsVAR model refers to the constrained graphical sparse VAR model.
* The structural VAR model has autoregressive coefficients to lag 0 and its noise components are independent.
# The 2-stage model estimates covariance matrix. Ten cells values of the covariance are not significant at 5% level.

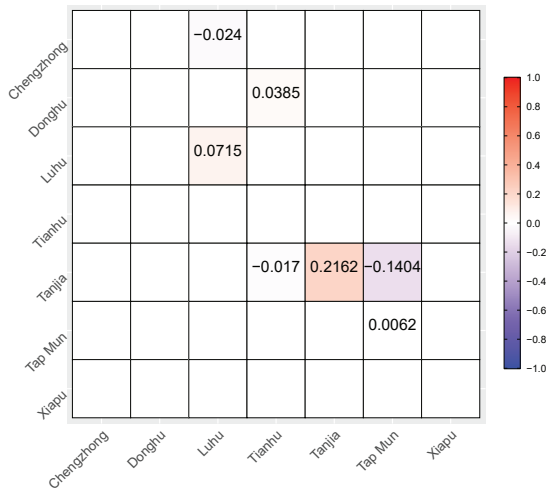
precision matrices is made and it would be more applicable in an exploratory stage in many areas. We have proved that the penalized maximum likelihood estimators of the model are consistent and converge to asymptotic normal distributions. We develop a new, effective and convergent iterative alternating algorithm for LASSO, SCAD and MCP penalized likelihood estimation for the sparse model. We overcome the challenge induced by some non-convex penalties in the penalized likelihood estimation, and allow flexibility to use the traditional LASSO method. Our algorithm does not require a Hessian matrix and enables us to obtain the iterative estimates using independent elementwise closed-form solutions, which allow parallel programming within the same iteration. This makes the complexity of the algorithm not increase much, as the dimension increases.

The simulation study shows that MCP provided the most satisfactory results. Its BIC is always the smallest and has always been selected among the three penalties. We have further illustrated the LASSO, SCAD and MCP sparse graphical VAR models on the Pearl River region RSP time series. Again, the MCP sparse model has minimum BIC and is the best among them. When we further compared our

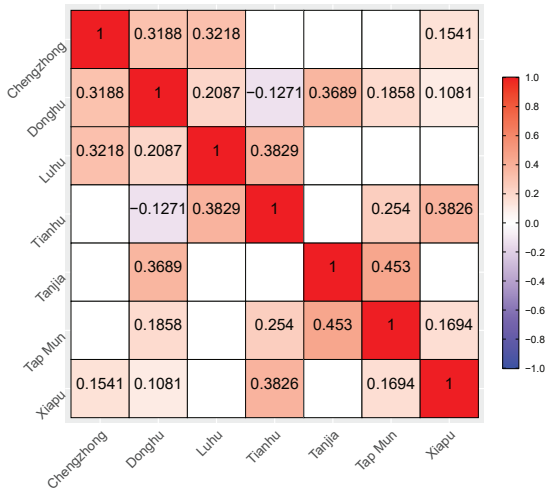
MCP sparse model results with the existing sparse time series models, our results are consistent with these models. In addition, our model avoids entangling more variables in the directed edges structures, as in structural VAR model or CGsVAR models; or detangling these directed edges, as in the 2-stage sparse VAR model. It has more balanced sparsity patterns over the AR coefficient and precision matrices and contains a few more mild correlated relationships in a higher lag order. It has the minimum BIC. Therefore, the proposed MCP sparse model is best.



(a) AR coefficients of lag order 1



(b) AR coefficients of lag order 2



(c) Partial correlations of innovations

Figure 2.4: The MCP penalized estimated AR coefficients and partial correlation of innovations for the RSP data.

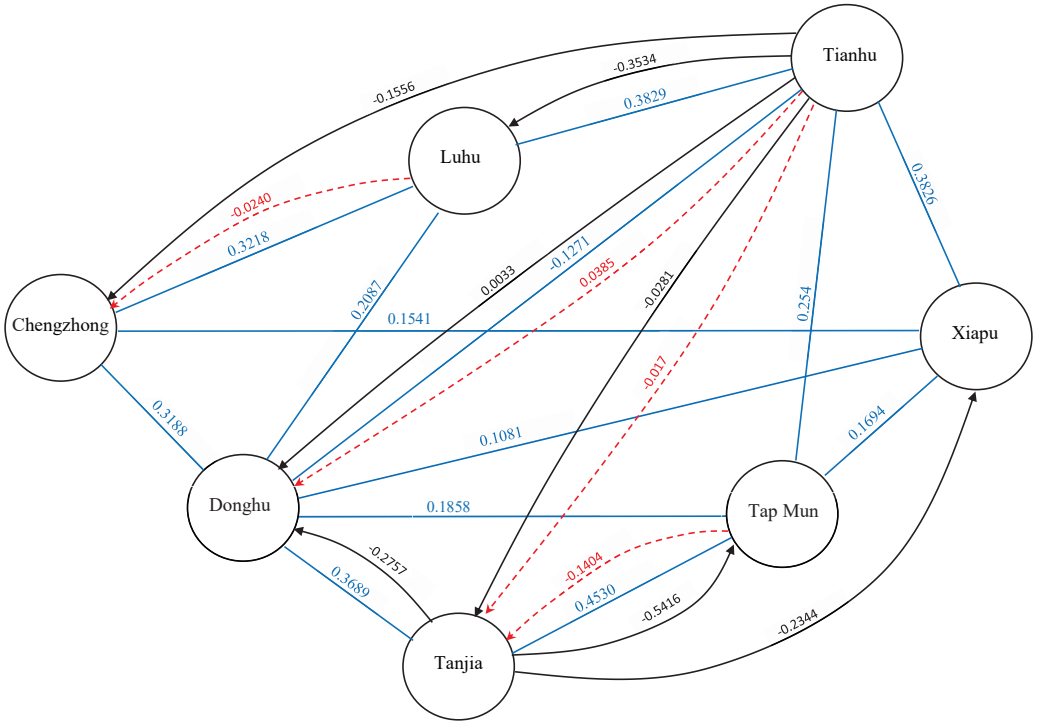
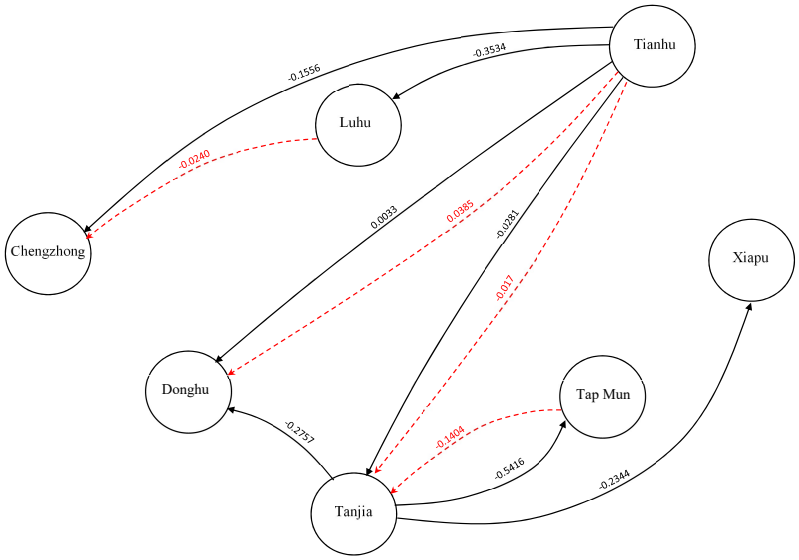
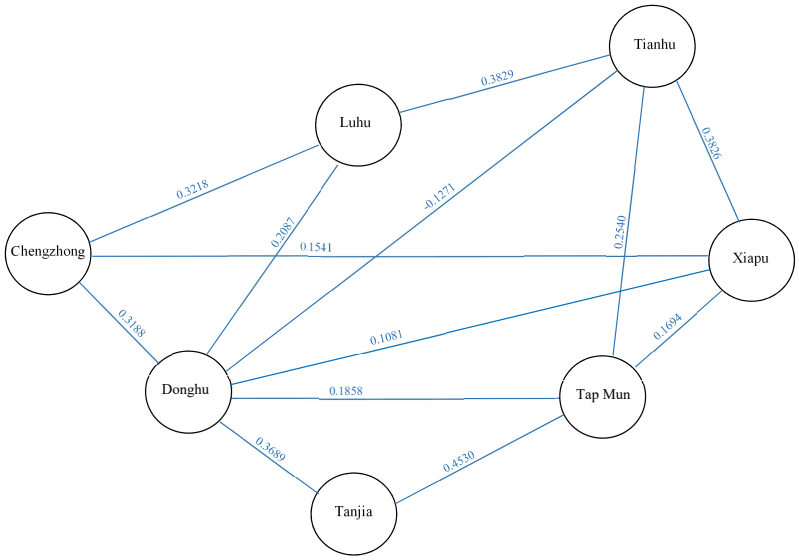


Figure 2.5: A mixed graph visualizing the MCP penalized estimated sparse graphical VAR(2) model for the RSP data. The black solid and red dashed arrows are directed edges representing AR order lag one and lag two coefficients respectively, while the blue solid blue lines are undirected edges representing partial correlations, which are determined by the precision matrix. The figure displays the approximate geographical location and is not drawn to scale.



(a)



(b)

Figure 2.6: (a) The temporal causal graph (directed component) and (b) the conditional independence graph (undirected component) of Figure 2.5. The black solid and red dashed arrows are directed edges representing AR order lag one and lag two coefficients respectively, while the blue solid blue lines are undirected edges representing partial correlations, which are determined by the precision matrix. The figure displays the approximate geographical location and is not drawn to scale.

Chapter 3

Graphical Matrix Time Series Models

Matrix-variate observations are commonly encountered in macroeconomics analysis and many other areas. For example, in a study of the influence of four economic indicators over five countries, it is natural to collect the quarterly data in a matrix form under the intersection of two categorical variables. The quarterly economic observations at each time point can naturally form a matrix $\mathbf{X}_t$ by taking each row as an economic indicator and each column as a country. The data exhibit more structural information, especially when the two categorical variables have a close relationship with each other. The collection of data matrix variates, $\mathbf{X}_t,\ t = 1, \ldots, T$, become a matrix-variate time series for analysis.

A traditional convention for analyzing matrix-variate observations is to treat these multiple observations as vectors and to model them by vector models. The traditional methods run two standard vector time series models for analysis and adopt dynamic factor analysis (Bai and Ng (2011), Forni et al. (2000); Lam et al. (2011)). To understand better the structural information, Tsai and Tsay (2010) added group constraints in a factor model for the time series. Hallin and Liška (2011) decomposed the time series into blocks and conducted factor analysis. All these methods treat all data using vector form.

Since the traditional modelling method cannot efficiently analyze the matrix structural relationship, matrix models have been first developed in regression and then extended to time series models. A bilinear matrix regression model originated from a growth curve model. Its model equation has a left and right design matrix, and an unknown matrix between design matrices for estimation. Theoretical details are given in von Rosen (2018). Chen et al. (2021) proposed to combine the autoregressive time series models with this bilinear regression model and called the proposed model a Matrix Autoregressive (MAR) model. Similar to the bilinear regression model, the MAR model has a bilinear form. It has a coefficient matrix multiplying a data matrix at time point $\mathbf{X}_t$ on the left side and another coefficient matrix multiplies the data matrix $\mathbf{X}_t$ on the right side. The left matrix of the bilinear form investigates the row-wise interactions and the column matrix examines the column-wise dependence. This bilinear form allows complete interpretability over the original matrix structure. This model further reduces the number of parameters by introducing the structured covariance tensor, which consists of existing row-wise covariance and column-wise covariance, and has a huge reduction in parameters as compared with a vector autoregressive model. However, the number of lag ready to use is one only and it is not adequate in many applications.

We aim to extend the existing matrix time series model to any general lag order and its inverse of the innovation covariance structure is free of structure. This proposed model enables exploring further the conditional dependence between variables and visualizing the relationship, based on a graphical model. The idea of the graphical model originated from the covariance selection problem in Dempster (1972) and a typical Gaussian graphical model can be found in Lauritzen (1996).

In addition, we would like to consider a sparse version of our 'graphical' MAR model by penalized estimation method. The penalized estimation on a likelihood with an innovation precision matrix builds a sparse graphical vector model for $\text{vec}(\mathbf{X}_t)$

and the sparsity of the precision matrix implies the sparsity of the partial correlation structure. We change the penalized estimation into a matrix setting and obtain a sparse graphical matrix time series model. It consists of a set of vertices and a set of edges, where each vertex represents a variable and an edge is absent from two vertices indicating the conditional independence between the corresponding variables. The absent edge is represented by a zero in the precision matrix.

The LASSO penalty will be adopted because of its popularity. Meinshausen and Bühlmann (2006) adopted the LASSO penalized method and Zhao and Leng (2014) studied the structured LASSO for regression with matrix covariates. It has been used as a benchmark for sparse penalized models.

In this chapter, we extend Chen et al. (2021)'s work on the MAR model as follows. We replace the structured covariance tensor of their MAR model with a freely structured precision matrix so that any imperfect relationship between the two categorical variables could be modelled. Besides this, the model lag order will be generalized to any general lag order, i.e. $p = 1, 2, 3, \ldots$. The detailed definition of the MAR(p) model will be given in Section 3.1. The data visualization concept from the graphical Gaussian model will be added to our MAR(p) model. LASSO penalized log-likelihood method is adopted. The sparse graphical MAR(p) model is defined in Section 3.2. Section 3.3 proposes two respective algorithms for these two models. The convergence of the algorithms is discussed. We conduct the simulation study in Section 3.4. Both algorithms work well for the MAR(p) model and the sparse graphical MAR(p) model. We revisit Chen et al. (2021)'s four economic indicators example in Section 3.5 and our general MAR(p) has a smaller in-sample residual sum of squares and an out-of-sample prediction error sum of squares than the existing MAR(p) model. A sparse graphical model is constructed and compared with the existing sparse graphical vector time series model. Again, our sparse graphical model is better because it has a smaller in-sample residual sum of squares and an out-of-

samples prediction error sum of squares. The corresponding conditional dependence graph is plotted and the relationship shown is intuitively correct.

3.1 General Matrix Autoregressive Models

Chen et al. (2021) adopt the bilinear regression in von Rosen (2018) into a matrix time series model and assume the covariance matrix of the model as a Kronecker product of column-wise covariance and row-wise covariance matrices. Assume data under the intersection of two classifications be an $m \times n$ valued matrix $\mathbf{X}_t$ at time t and the time series has a length T. Chen et al. (2021)'s bilinear form matrix autoregressive model is the following:

$$\mathbf{X}_t \;=\; \mathbf{A}\mathbf{X}_{t-1}\mathbf{B}^T + \mathbf{E}_t, \quad t = 1, \ldots, T, \tag{3.1}$$

where $\text{vec}(\mathbf{E}_t) \sim N(0, \mathbf{\Sigma}_c \otimes \mathbf{\Sigma}_r)$, $\mathbf{A} = (a_{ij})$ is an $m \times m$ left coefficient matrix, $\mathbf{B} = (b_{ij})$ is an $n \times n$ right coefficient matrix, $\mathbf{E}_t = (e_{ij,t})$ is an $m \times n$ white noise matrix and $\otimes$ is a Kronecker product.

Under this setting, the data under two classifications can be expressed as a matrix and a perfect independent relationship between the two classifications is assumed. The model takes advantage of the original matrix structure and reduced the dimension significantly. However, the model is restricted to lag order one and the structure of the covariance matrix is also very restrictive. These two features may not be adequate for real-life applications.

3.1.1 General MAR(p) models

We extend his matrix autoregressive (MAR) time series model to lag order p and combine it with a graphical model, originated from Dempster (1972). Lauritzen (1996) gives the theoretical details of the graphical model.

$$\mathbf{X}_t = \sum_{k=1}^{p} \mathbf{A}_k \mathbf{X}_{t-k} \mathbf{B}_k^T + \mathbf{E}_t, \qquad \text{for } t = 1, \ldots, T, \tag{3.2}$$

where $\text{vec}(\mathbf{E}_t) \sim N(\mathbf{0}, \mathbf{\Sigma})$, $\mathbf{A}_k = (a_{ij,k})$, is an $m \times m$ left autoregressive coefficient matrix, $\mathbf{B}_k = (b_{ij,k})$ is an $n \times n$ right autoregressive coefficient matrix and $\mathbf{E}_t = (e_{ij,t})$ is a $m \times n$ matrix white noise. Here, we assume $\text{Cov}(\text{vec}(\mathbf{E}_t)) = \mathbf{\Sigma}$, where $\mathbf{\Sigma} = \mathbf{\Theta}^{-1}$ is a $mn \times mn$ symmetric positive definite covariance matrix and $\mathbf{\Theta}$ is a $mn \times mn$ symmetric positive definite precision matrix.

The log-likelihood function of (3.2) is

$$\begin{aligned} l(\mathbf{A}, \mathbf{B}, \mathbf{\Sigma}) \;=\; & \frac{1}{2}\Big\{ - mn(T-p)\log(2\pi) + (T-p)\log(\det(\mathbf{\Sigma}^{-1})) \\ & - \sum_{t=p+1}^{T} \text{vec}(\mathbf{X}_t)^T \mathbf{\Sigma}^{-1} \text{vec}(\mathbf{X}_t) \\ & + \sum_{t=p+1}^{T} \sum_{i=1}^{p} \text{vec}(\mathbf{X}_{t-i})^T (\mathbf{B}_i^T \otimes \mathbf{A}_i^T) \mathbf{\Sigma}^{-1} \text{vec}(\mathbf{X}_t) \\ & + \sum_{t=p+1}^{T} \sum_{i=1}^{p} \text{vec}(\mathbf{X}_t)^T \mathbf{\Sigma}^{-1} (\mathbf{B}_i \otimes \mathbf{A}_i) \text{vec}(\mathbf{X}_{t-i}) \\ & - \sum_{t=p+1}^{T} \sum_{i=1}^{p} \sum_{j=1}^{p} \text{vec}(\mathbf{X}_{t-i})^T (\mathbf{B}_i^T \otimes \mathbf{A}_i^T) \mathbf{\Sigma}^{-1} (\mathbf{B}_j \otimes \mathbf{A}_j) \text{vec}(\mathbf{X}_{t-j}) \Big\} \end{aligned} \tag{3.3}$$

And this MAR(p) model can be represented by the following vector autoregressive model:

$$\text{vec}(\mathbf{X}_t) = (\mathbf{B}_1 \otimes \mathbf{A}_1)\text{vec}(\mathbf{X}_{t-1}) + \cdots + (\mathbf{B}_p \otimes \mathbf{A}_p)\text{vec}(\mathbf{X}_{t-p}) + \text{vec}(\mathbf{E}_t),$$

where $\otimes$ denotes the matrix Kronecker product. Lots of properties can be derived from its corresponding VAR(p) model. Lütkepohl (2005) gives theoretical details of the VAR(p) model.

3.1.2 Stability condition of the general MAR(p) model

We can adopt Proposition 1 of Chen et al. (2021) for the stability condition of our MAR(1) model, because the structure of covariance assumed in their proof of the stability is also valid in our case. As a result, the product of radius spectral $\rho(\mathbf{A})\cdot\rho(\mathbf{B})$ for a MAR(1) model is less than 1.

As p (≥ 2) gets larger, we define

$$\mathcal{A} = \begin{bmatrix} \mathbf{B}_1 \otimes \mathbf{A}_1 & \mathbf{B}_2 \otimes \mathbf{A}_2 & \cdots & \mathbf{B}_{p-1} \otimes \mathbf{A}_{p-1} & \mathbf{B}_p \otimes \mathbf{A}_p \\ \mathbf{I}_{mn} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{mn} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & & \ddots & & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I}_{mn} & \mathbf{0} \end{bmatrix}.$$

By Chapter 2 of Lütkepohl (2005), we can express the stability condition of a MAR(p) model is that the moduli of all eigenvalues of $\mathcal{A}$ are less than 1. We select $\mathbf{A}_i$'s and $\mathbf{B}_i$'s satisfying this condition and $\|\mathbf{A}_i\| = 1$ for $i = 1, \ldots, p$ is used to fix the identification problem.

3.2 Sparse Graphical MAR(p) Models

The purpose of using fewer parameters is not preserved in our general MAR(p) model, because its precision matrix $\mathbf{\Theta}$ has $mn \times mn$ number of parameters. But using a huge number of parameters with a very limited observation would cause inaccurate model estimation in high dimensional data. Therefore, Chen et al. (2021)'s MAR(1) model proposed a structured covariance tensor for a dimension reduction purpose. Then a huge number of parameters reduction will be found in the covariance matrix.

A modern approach to reducing the number of parameters is to make the model "sparse" by penalized log-likelihood estimation. Penalties are imposed in the log-likelihood function to make small values shrink in row-wise interactions coefficient $\mathbf{A}$ and column-wise dependence coefficient matrices as well as the precision matrix $\mathbf{\Theta}$.

This resulting sparse MAR(p) model gives another advantage of understanding the conditional dependence between variables. A mixed graph could be plotted to visualize the relationship between variables.

We propose a sparse graphical MAR(p) model, in a similar way, as our sparse graphical VAR(p) model Equation (2.5). i.e. The sparsity of the proposed model is selected from all possible sparse left and right coefficient matrices and all possible sparse precision matrices, based on minimum BIC. Then the model achieves an optimal sparsity.

LASSO is a popular penalty and is always used as a benchmark for all penalized estimation methods. Compared with SCAD and MCP, its computational time is the shortest. Therefore, it is chosen for our sparse model.

Let $\mathbf{A} = (\mathbf{A}_1, \ldots, \mathbf{A}_p)$ and $\mathbf{B} = (\mathbf{B}_1, \ldots, \mathbf{B}_p)$. The sparse graphical MAR(p) model has the LASSO penalized log-likelihood function as below.

$$\begin{aligned} \arg\min_{\mathbf{A},\mathbf{B},\mathbf{\Theta}} F(\mathbf{A},\mathbf{B},\mathbf{\Theta}) \;\; := \;\; & -l(\mathbf{A},\mathbf{B},\mathbf{\Theta}) + T\sum_{i,j,k}\lambda_{\mathbf{A}}|a_{ij,k}| + T\sum_{i,j,k}\lambda_{\mathbf{B}}|b_{ij,k}| \\ & + T\sum_{i\neq j}\lambda_{\mathbf{\Theta}}|\theta_{ij}|, \end{aligned} \tag{3.4}$$

where $\lambda_{\mathbf{A}}$, $\lambda_{\mathbf{B}}$ and $\lambda_{\mathbf{\Theta}}$ are regularization parameters for the $m \times m$ row-wise interaction matrix, $\mathbf{A}_k = (a_{ij,k})_{i,j=1,\ldots,m}$ with $\|\mathbf{A}_k\| = 1$, the $n \times n$ column-wise dependence matrix, $\mathbf{B}_k = (b_{ij,k})_{i,j=1,\ldots,n}$ (for $k = 1, \ldots, p$) and the $mn \times mn$ precision matrix, $\mathbf{\Theta} = \mathbf{\Sigma}^{-1} = (\theta_{ij})_{i,j=1,\ldots,mn}$ and T is the sample size. To solve the identifiability problem, we fix $\|\mathbf{A}_i\| = 1$ for $i = 1, \ldots, p$.

3.3 Estimation

To set up an algorithm for estimating the MAR(p) Model, we need to derive the first-order derivative of the log-likelihood function. This requires the following lemma:

Lemma 3.1. *Let* $\mathbf{a} = (a_{ij}), \mathbf{A}$ *be an* $m \times m$ *matrix,* $\mathbf{b} = (b_{ij}), \mathbf{B} = (B_{ij})$ *be an* $n \times n$ *matrix and* $\mathbf{R}$ *be an* $mn \times mn$ *matrix, the following expressions hold:*

1. $\mathrm{Tr}\left(\mathbf{R}(\mathbf{b} \otimes \mathbf{A})\right) = \mathrm{Tr}\left((\mathbf{I}_n \otimes \mathbf{1}_m^T)\left[\mathbf{R}(\mathbf{I}_n \otimes \mathbf{A}) \odot (\mathbf{J}_n \otimes \mathbf{I}_m)\right](\mathbf{I}_n \otimes \mathbf{1}_m)\mathbf{b}\right)$;
2. $\mathrm{Tr}\left(\mathbf{R}(\mathbf{b}^T \otimes \mathbf{A}^T)\right) = \mathrm{Tr}\left((\mathbf{I}_n \otimes \mathbf{1}_m^T)\left[(\mathbf{I}_n \otimes \mathbf{A})\mathbf{R}^T \odot (\mathbf{J}_n \otimes \mathbf{I}_m)\right](\mathbf{I}_n \otimes \mathbf{1}_m)\mathbf{b}\right)$;
3. $\mathrm{Tr}\left(\mathbf{R}(\mathbf{B}^T \otimes \mathbf{a}^T)\right) = \mathrm{Tr}\left((\mathbf{1}_n^T \otimes \mathbf{I}_m)\left[(\mathbf{B}^T \otimes \mathbf{J}_m) \odot \mathbf{R}^T\right](\mathbf{1}_n \otimes \mathbf{I}_m)\mathbf{a}\right)$; *and*
4. $\mathrm{Tr}\left(\mathbf{R}(\mathbf{B} \otimes \mathbf{a})\right) = \mathrm{Tr}\left((\mathbf{1}_n^T \otimes \mathbf{I}_m)\left[(\mathbf{B}^T \otimes \mathbf{J}_m) \odot \mathbf{R}\right](\mathbf{1}_n \otimes \mathbf{I}_m)\mathbf{a}\right)$,

where $\mathbf{1}_m$ *is a column vector with* m *entries being 1,* $\mathbf{J}_m$ *is an* $m \times m$ *matrix with all entries being 1,* $\mathbf{I}_m$ *is an* $m \times m$ *identity matrix,* $\otimes$ *is a Kronecker product and* $\odot$ *is a Hadamard product of matrices.*

Proof Let $\mathbf{R} = \begin{pmatrix} \mathbf{R}_{11} & \cdots & \mathbf{R}_{1n} \\ \vdots & & \vdots \\ \mathbf{R}_{n1} & \cdots & \mathbf{R}_{nn} \end{pmatrix}$ and $\mathbf{R}^T = \begin{pmatrix} \mathbf{R}_{11}^* & \cdots & \mathbf{R}_{1n}^* \\ \vdots & & \vdots \\ \mathbf{R}_{n1}^* & \cdots & \mathbf{R}_{nn}^* \end{pmatrix}$, where $\mathbf{R}_{ij}$ and $\mathbf{R}_{ij}^*$ are $m \times m$ matrices for $i, j = 1, \ldots, n$.

1.

$$\begin{aligned} \mathrm{Tr}\left(\mathbf{R}(\mathbf{b} \otimes \mathbf{A})\right) &= \mathrm{Tr}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}\mathbf{R}_{ij}\mathbf{A}\right) \\ &= \mathrm{Tr}\left(\begin{pmatrix} \mathrm{Tr}(\mathbf{R}_{11}\mathbf{A}) & & \mathrm{Tr}(\mathbf{R}_{1n}\mathbf{A}) \\ \vdots & & \\ \mathrm{Tr}(\mathbf{R}_{n1}\mathbf{A}) & & \mathrm{Tr}(\mathbf{R}_{nn}\mathbf{A}) \end{pmatrix}\begin{pmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & & \vdots \\ b_{n1} & \cdots & b_{nn} \end{pmatrix}\right) \end{aligned}$$

Note that $\mathrm{Tr}(\mathbf{R}_{ij}\mathbf{A}) = \mathrm{Tr}(\mathbf{R}_{ij}\mathbf{A} \odot \mathbf{I}_m) = \mathbf{1}_m^T \left(\mathbf{R}_{ij}\mathbf{A} \odot \mathbf{I}_m\right) \mathbf{1}_m$ for $i, j = 1, \ldots, n$.

$$
\begin{aligned}
& \mathrm{Tr}\left(\mathbf{R}(\mathbf{b} \otimes \mathbf{A})\right) \\
= \; & \mathrm{Tr}\left(\begin{pmatrix} \mathrm{Tr}(\mathbf{R}_{11}\mathbf{A} \odot \mathbf{I}_m) & \cdots & \mathrm{Tr}(\mathbf{R}_{1n}\mathbf{A} \odot \mathbf{I}_m) \\ \vdots & & \vdots \\ \mathrm{Tr}(\mathbf{R}_{n1}\mathbf{A} \odot \mathbf{I}_m) & \cdots & \mathrm{Tr}(\mathbf{R}_{nn}\mathbf{A} \odot \mathbf{I}_m) \end{pmatrix} \begin{pmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & & \vdots \\ b_{n1} & \cdots & b_{nn} \end{pmatrix} \right) \\
= \; & \mathrm{Tr}\Bigg(\underbrace{\begin{pmatrix} \mathbf{1}_m^T & 0 & \cdots & 0 \\ 0 & \mathbf{1}_m^T & & \\ \vdots & & \ddots & \\ 0 & 0 & \cdots & \mathbf{1}_m^T \end{pmatrix}}_{n \times n \text{ blocks}} \begin{pmatrix} \mathbf{R}_{11}\mathbf{A} \odot \mathbf{I}_m & \cdots & \mathbf{R}_{1n}\mathbf{A} \odot \mathbf{I}_m \\ \vdots & & \vdots \\ \mathbf{R}_{n1}\mathbf{A} \odot \mathbf{I}_m & \cdots & \mathbf{R}_{nn}\mathbf{A} \odot \mathbf{I}_m \end{pmatrix} \\
& \qquad \underbrace{\begin{pmatrix} \mathbf{1}_m & 0 & \cdots & 0 \\ 0 & \mathbf{1}_m & & \vdots \\ \vdots & & \ddots & \\ 0 & 0 & \cdots & \mathbf{1}_m \end{pmatrix}}_{n \times n \text{ blocks}} \mathbf{b} \Bigg)
\end{aligned}
\tag{3.5}
$$

Hence, $\mathrm{Tr}\left(\mathbf{R}(\mathbf{b} \otimes \mathbf{A})\right) = \mathrm{Tr}\left((\mathbf{I}_n \otimes \mathbf{1}_m^T)\left[\mathbf{R}(\mathbf{I}_n \otimes \mathbf{A}) \odot (\mathbf{J}_n \otimes \mathbf{I}_m)\right](\mathbf{I}_n \otimes \mathbf{1}_m)\mathbf{b}\right)$.

2.

$$
\begin{aligned}
\mathrm{Tr}\left(\mathbf{R}(\mathbf{b}^T \otimes \mathbf{A}^T)\right) &= \mathrm{Tr}\left((\mathbf{b} \otimes \mathbf{A})\mathbf{R}^T\right) \\
&= \mathrm{Tr}\left(\mathbf{R}^T(\mathbf{b} \otimes \mathbf{A})\right)
\end{aligned}
$$

By Expression 1, the result follows.

3.

$$\begin{aligned}
\mathrm{Tr}\left(\mathbf{R}(\mathbf{B}^T \otimes \mathbf{a}^T)\right) &= \mathrm{Tr}\left((\mathbf{B} \otimes \mathbf{a})\mathbf{R}^T\right) \\
&= \mathrm{Tr}\left(\mathbf{R}^T(\mathbf{B} \otimes \mathbf{a})\right) \\
&= \mathrm{Tr}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} B_{ji}\mathbf{R}_{ij}^{*}\mathbf{a}\right) \\
&= \mathrm{Tr}\left(\left(\sum_{i=1}^{n}\sum_{j=1}^{n} B_{ji}\mathbf{R}_{ij}^{*}\right)\mathbf{a}\right) \qquad (3.6)
\end{aligned}$$

Note that

$$\begin{aligned}
\sum_{i=1}^{n}\sum_{j=1}^{n} B_{ji}\mathbf{R}_{ij}^{*} &= \underbrace{\begin{pmatrix} \mathbf{I}_m & \cdots & \mathbf{I}_m \end{pmatrix}}_{n \text{ blocks}} \begin{pmatrix} B_{11}\mathbf{R}_{11}^{*} & \cdots & B_{n1}\mathbf{R}_{1n}^{*} \\ \vdots & & \vdots \\ B_{1n}\mathbf{R}_{n1}^{*} & \cdots & B_{nn}\mathbf{R}_{nn}^{*} \end{pmatrix} \underbrace{\begin{pmatrix} \mathbf{I}_m \\ \vdots \\ \mathbf{I}_m \end{pmatrix}}_{n \text{ vertical blocks}} \\
&= \left(\mathbf{1}_n^T \otimes \mathbf{I}_m\right)\left((\mathbf{B}^T \otimes \mathbf{J}_m) \odot \mathbf{R}^T\right)\left(\mathbf{1}_n^T \otimes \mathbf{I}_m\right) \qquad (3.7)
\end{aligned}$$

By combining (3.6) and (3.7), the result follows.

4.

$$\begin{aligned}
\mathrm{Tr}\left(\mathbf{R}(\mathbf{B} \otimes \mathbf{a})\right) &= \mathrm{Tr}\left((\mathbf{R}(\mathbf{B} \otimes \mathbf{a}))^T\right) \\
&= \mathrm{Tr}\left((\mathbf{B} \otimes \mathbf{a})^T\mathbf{R}^T\right) \\
&= \mathrm{Tr}\left(\mathbf{R}^T(\mathbf{B} \otimes \mathbf{a})^T\right) \\
&= \mathrm{Tr}\left(\mathbf{R}^T(\mathbf{B}^T \otimes \mathbf{a}^T)\right)
\end{aligned}$$

By Expression 3, the result follows.

□

Numerical verification was done with R programs.

We aim at setting up an algorithm of the MAR(p) and derive the gradient of the log-likelihood function with respect to $\mathbf{A}, \mathbf{B}$ and $\mathbf{\Sigma}^{-1}$.

Theorem 3.1. *Let* $\mathbf{Y}_1 = \sum_{t=p+1}^{T} vec(\mathbf{X}_t)vec(\mathbf{X}_t)^T$, $\mathbf{Y}_{2,i} = \sum_{t=p+1}^{T} vec(\mathbf{X}_t)vec(\mathbf{X}_{t-i})^T$ *and* $\mathbf{Y}_{3,i,j} = \sum_{t=p+1}^{T} vec(\mathbf{X}_{t-i})vec(\mathbf{X}_{t-j})^T$ *for* $i, j = 1, \ldots, p$. *Let* $\mathbf{A} = (\mathbf{A}_1, \ldots, \mathbf{A}_p)$ *and* $\mathbf{B} = (\mathbf{B}_1, \ldots, \mathbf{B}_p)$. *Then the gradient of the log-likelihood function of MAR(p) is*

$$\nabla l(\mathbf{A}, \mathbf{B}, \mathbf{\Sigma}^{-1}) = \left(D_{\mathbf{A}} l(\mathbf{A}, \mathbf{B}, \mathbf{\Sigma}^{-1}), D_{\mathbf{B}} l(\mathbf{A}, \mathbf{B}, \mathbf{\Sigma}^{-1}), D_{\mathbf{\Sigma}^{-1}} l(\mathbf{A}, \mathbf{B}, \mathbf{\Sigma}^{-1})\right),$$

where

$$D_{\mathbf{A}_i} l(\mathbf{A}, \mathbf{B}, \mathbf{\Sigma}^{-1}) = (\mathbf{1}_n^T \otimes \mathbf{I}_m)\left[(\mathbf{B}_i \otimes \mathbf{J}_m) \odot \mathbf{Y}_{4,i}\right](\mathbf{1}_n \otimes \mathbf{I}_m) \tag{3.8}$$

$$D_{\mathbf{B}_i} l(\mathbf{A}, \mathbf{B}, \mathbf{\Sigma}^{-1}) = \frac{1}{2}(\mathbf{I}_n \otimes \mathbf{1}_m^T)\left(\left[(\mathbf{I}_n \otimes \mathbf{A}_i^T)\mathbf{Y}_{4,i} + \mathbf{Y}_{4,i}(\mathbf{I}_n \otimes \mathbf{A}_i^T)\right] \odot (\mathbf{J}_n \otimes \mathbf{I}_m)\right) (\mathbf{I}_n \otimes \mathbf{1}_m) \tag{3.9}$$

$$\mathbf{Y}_{4,i} = \left[\mathbf{\Sigma}^{-1}\left(\mathbf{Y}_{2,i} - (\sum_{j=1}^{p}(\mathbf{B}_j \otimes \mathbf{A}_j)\mathbf{Y}_{3,i,j}^T\right)\right]$$

for $i = 1, \ldots, p$,

$$D_{\mathbf{\Sigma}^{-1}} l(\mathbf{A}, \mathbf{B}, \mathbf{\Sigma}^{-1}) = \frac{1}{2}\Big\{(T-1)\mathbf{\Sigma} - \mathbf{Y}_1 + \sum_{i=1}^{p}(\mathbf{B}_i \otimes \mathbf{A}_i)\mathbf{Y}_{2,i}^T + \sum_{i=1}^{p}\mathbf{Y}_{2,i}(\mathbf{B}_i^T \otimes \mathbf{A}_i^T) - \sum_{i=1}^{p}\sum_{j=1}^{p}(\mathbf{B}_i \otimes \mathbf{A}_i)\mathbf{Y}_{3,i,j}(\mathbf{B}_j^T \otimes \mathbf{A}_j^T)\Big\} \tag{3.10}$$

Proof Let $\mathbf{A} = (\mathbf{A}_1, \ldots, \mathbf{A}_p)$ and $\mathbf{B} = (\mathbf{B}_1, \ldots, \mathbf{B}_p)$, $\mathbf{a} = (\mathbf{a}_1, \ldots, \mathbf{a}_p)$ and $\mathbf{b} = (\mathbf{b}_1, \ldots, \mathbf{b}_p)$, where $\mathbf{a}_i (i = 1, \ldots, p), \mathbf{b}_i (i = 1, \ldots, p), \mathbf{s}$ are $m \times m$, $n \times n$ and $mn \times mn$ small matrices respectively. Consider the Taylor's expansion of the log-likelihood function (3.2),

$$
\begin{aligned}
& l(\mathbf{A}+\mathbf{a},\mathbf{B}+\mathbf{b},\boldsymbol{\Sigma}^{-\mathbf{1}}+\mathbf{s}) - \mathbf{l}(\mathbf{A},\mathbf{B},\boldsymbol{\Sigma}^{-\mathbf{1}}) \\
= \; & \frac{1}{2}\Big\{(T-p)\Big(\log(\det(\boldsymbol{\Sigma}^{-1}+\mathbf{s})) - \log(\det(\boldsymbol{\Sigma}^{-1}))\Big) \\
& - \sum_{t=p+1}^{T} \mathrm{vec}(\mathbf{X}_t)^T(\boldsymbol{\Sigma}^{-1}+\mathbf{s})\mathrm{vec}(\mathbf{X}_t) \\
& + \sum_{t=p+1}^{T} \mathrm{vec}(\mathbf{X}_t)^T\boldsymbol{\Sigma}^{-1}\mathrm{vec}(\mathbf{X}_t) \\
& + \sum_{t=p+1}^{T}\sum_{i=1}^{p} \mathrm{vec}(\mathbf{X}_{t-i})^T\left((\mathbf{B}_i+\mathbf{b}_i)^T\otimes(\mathbf{A}_i+\mathbf{a}_i)^T\right)(\boldsymbol{\Sigma}^{-1}+\mathbf{s}^{-1})\mathrm{vec}(\mathbf{X}_t) \\
& - \sum_{t=p+1}^{T}\sum_{i=1}^{p} \mathrm{vec}(\mathbf{X}_{t-i})^T(\mathbf{B}_i^T\otimes\mathbf{A}_i^T)\boldsymbol{\Sigma}^{-1}\mathrm{vec}(\mathbf{X}_t) \\
& + \sum_{t=p+1}^{T}\sum_{i=1}^{p} \mathrm{vec}(\mathbf{X}_t)^T(\boldsymbol{\Sigma}^{-1}+\mathbf{s}^{-1})\left((\mathbf{B}_i+\mathbf{b}_i)\otimes(\mathbf{A}_i+\mathbf{a}_i)\right)\mathrm{vec}(\mathbf{X}_{t-i}) \\
& - \sum_{t=p+1}^{T}\sum_{i=1}^{p} \mathrm{vec}(\mathbf{X}_t)^T\boldsymbol{\Sigma}^{-1}(\mathbf{B}_i\otimes\mathbf{A}_i)\mathrm{vec}(\mathbf{X}_{t-i}) \\
& - \sum_{t=p+1}^{T}\sum_{i=1}^{p}\sum_{j=1}^{p} \mathrm{vec}(\mathbf{X}_{t-i})^T\left((\mathbf{B}_i+\mathbf{b}_i)^T\otimes(\mathbf{A}_i+\mathbf{a}_i)^T\right)\cdot \\
& \qquad\qquad \left(\boldsymbol{\Sigma}^{-1}+\mathbf{s}^{-1}\right)\left((\mathbf{B}_j+\mathbf{b}_j)\otimes(\mathbf{A}_j+\mathbf{a}_j)\right)\mathrm{vec}(\mathbf{X}_{t-j}) \\
& + \sum_{t=p+1}^{T}\sum_{i=1}^{p}\sum_{j=1}^{p} \mathrm{vec}(\mathbf{X}_{t-i})^T(\mathbf{B}_i^T\otimes\mathbf{A}_i^T)\boldsymbol{\Sigma}^{-1}(\mathbf{B}_j\otimes\mathbf{A}_j)\mathrm{vec}(\mathbf{X}_{t-j})\Big\}
\end{aligned}
$$

$$
\begin{aligned}
= \quad & \frac{1}{2}\Bigg\{(T-1)\Big(\log(\det(\boldsymbol{\Sigma}^{-1}+\mathbf{s}))-\log(\det(\boldsymbol{\Sigma}^{-1}))\Big)-\sum_{t=p+1}^{T}\mathrm{vec}(\mathbf{X}_t)^T\mathbf{s}\,\mathrm{vec}(\mathbf{X}_t) \\
& +\sum_{t=p+1}^{T}\sum_{i=1}^{p}\Big[\mathrm{vec}(\mathbf{X}_{t-i})^T\Big((\mathbf{b}_i^T\otimes\mathbf{A}_i^T)\boldsymbol{\Sigma}^{-1}+(\mathbf{B}_i^T\otimes\mathbf{a}_i^T)\boldsymbol{\Sigma}^{-1} \\
& \qquad +(\mathbf{B}_i^T\otimes\mathbf{A}_i^T)\mathbf{s}\Big)\mathrm{vec}(\mathbf{X}_t)\Big]+\sum_{t=p+1}^{T}\sum_{i=1}^{p}\big[\mathrm{vec}(\mathbf{X}_t)^T\cdot \\
& \qquad \Big(\boldsymbol{\Sigma}^{-1}(\mathbf{b}_i\otimes\mathbf{A}_i)+\boldsymbol{\Sigma}^{-1}(\mathbf{B}_i\otimes\mathbf{a}_i)+\mathbf{s}(\mathbf{B}_i\otimes\mathbf{A}_i)\Big)\mathrm{vec}(\mathbf{X}_{t-i})\Big] \\
& -\sum_{t=p+1}^{T}\sum_{i=1}^{p}\sum_{j=1}^{p}\Big[\mathrm{vec}(\mathbf{X}_{t-i})^T\Big((\mathbf{b}_i^T\otimes\mathbf{A}_i^T)\boldsymbol{\Sigma}^{-1}(\mathbf{B}_j\otimes\mathbf{A}_j) \\
& \qquad +(\mathbf{B}_i^T\otimes\mathbf{a}_i^T)\boldsymbol{\Sigma}^{-1}(\mathbf{B}_j\otimes\mathbf{A}_j)+(\mathbf{B}_i^T\otimes\mathbf{A}^T)\mathbf{s}(\mathbf{B}_j\otimes\mathbf{A}_j) \\
& +(\mathbf{B}_i^T\otimes\mathbf{A}_i^T)\boldsymbol{\Sigma}^{-1}(\mathbf{b}_j\otimes\mathbf{A}_j)+(\mathbf{B}_i^T\otimes\mathbf{A}_i^T)\boldsymbol{\Sigma}^{-1}(\mathbf{B}_j\otimes\mathbf{a}_j)\Big)\mathrm{vec}(\mathbf{X}_{t-j})\Big]\Bigg\} \\
& +O(\|\mathbf{a}\|^2)+O(\|\mathbf{b}\|^2)+O(\|\mathbf{s}\|^2)
\end{aligned}
$$

$$
\begin{aligned}
= \quad & \frac{1}{2}\Bigg\{\mathrm{Tr}\Big((T-1)\boldsymbol{\Sigma}\mathbf{s}-\sum_{t=p+1}^{T}\mathrm{vec}(\mathbf{X}_t)\mathrm{vec}(\mathbf{X}_t)^T\mathbf{s} \\
& \qquad +\sum_{t=p+1}^{T}\sum_{i=1}^{p}(\mathbf{B}_i\otimes\mathbf{A}_i)\mathrm{vec}(\mathbf{X}_{t-i})\mathrm{vec}(\mathbf{X}_t)^T\mathbf{s} \\
& \qquad +\sum_{t=p+1}^{T}\sum_{i=1}^{p}\mathrm{vec}(\mathbf{X}_t)\mathrm{vec}(\mathbf{X}_{t-i})^T(\mathbf{B}_i^T\otimes\mathbf{A}_i^T)\mathbf{s} \\
& \qquad -\sum_{t=p+1}^{T}\sum_{i=1}^{p}\sum_{j=1}^{p}(\mathbf{B}_i\otimes\mathbf{A}_i)\Big(\mathrm{vec}(\mathbf{X}_{t-i})\mathrm{vec}(\mathbf{X}_{t-j})^T\Big)(\mathbf{B}_j^T\otimes\mathbf{A}_j^T)\mathbf{s}\Big)
\end{aligned}
$$

$$
\begin{aligned}
&+\operatorname{Tr}\Big(\sum_{t=p+1}^{T}\sum_{i=1}^{p}\boldsymbol{\Sigma}^{-1}\mathrm{vec}(\mathbf{X}_t)\mathrm{vec}(\mathbf{X}_{t-i})^T\big((\mathbf{b}_i^T\otimes\mathbf{A}_i^T)+(\mathbf{B}_i^T\otimes\mathbf{a}_i^T)\big)\\
&\quad+\sum_{t=p+1}^{T}\sum_{i=1}^{p}\mathrm{vec}(\mathbf{X}_{t-i})\mathrm{vec}(\mathbf{X}_t)^T\boldsymbol{\Sigma}^{-1}\big((\mathbf{b}_i\otimes\mathbf{A}_i)+(\mathbf{B}_i\otimes\mathbf{a}_i)\big)\\
&\quad-\sum_{t=p+1}^{T}\sum_{i=1}^{p}\sum_{j=1}^{p}\big[\boldsymbol{\Sigma}^{-1}(\mathbf{B}_i\otimes\mathbf{A}_i)\mathrm{vec}(\mathbf{X}_{t-i})\mathrm{vec}(\mathbf{X}_{t-j})^T\cdot\\
&\qquad\big((\mathbf{b}_j^T\otimes\mathbf{A}_j^T)+(\mathbf{B}_j^T\otimes\mathbf{a}_j^T)\big)\big]\\
&\quad-\sum_{t=p+1}^{T}\sum_{i=1}^{p}\sum_{j=1}^{p}\big[\mathrm{vec}(\mathbf{X}_{t-i})\mathrm{vec}(\mathbf{X}_{t-j})^T(\mathbf{B}_j^T\otimes\mathbf{A}_j^T)\boldsymbol{\Sigma}^{-1}\cdot\\
&\qquad\big((\mathbf{b}_i\otimes\mathbf{A}_i)+(\mathbf{B}_i\otimes\mathbf{a}_i)\big)\big]\Big)\Big\}\\
&\quad+O(\|\mathbf{a}\|^2)+O(\|\mathbf{b}\|^2)+O(\|\mathbf{s}\|^2)\\
=\;&\frac{1}{2}\Big\{\operatorname{Tr}\Big(\big[(T-1)\boldsymbol{\Sigma}-\mathbf{Y}_1+\sum_{i=1}^{p}(\mathbf{B}_i\otimes\mathbf{A}_i)\mathbf{Y}_{2,i}^T+\sum_{i=1}^{p}\mathbf{Y}_{2,i}(\mathbf{B}_i^T\otimes\mathbf{A}_i^T)\\
&\qquad-\sum_{i=1}^{p}\sum_{j=1}^{p}(\mathbf{B}_i\otimes\mathbf{A}_i)\mathbf{Y}_{3,i,j}(\mathbf{B}_j^T\otimes\mathbf{A}_j^T)\big]\mathbf{s}\Big)\\
&+\operatorname{Tr}\Big(\sum_{j=1}^{p}\big[\boldsymbol{\Sigma}^{-1}\mathbf{Y}_{2,j}-\sum_{i=1}^{p}\boldsymbol{\Sigma}^{-1}(\mathbf{B}_i\otimes\mathbf{A}_i)\mathbf{Y}_{3,i,j}\big](\mathbf{B}_j^T\otimes\mathbf{a}_j^T)\\
&\quad+\sum_{j=1}^{p}\big[\mathbf{Y}_{2,j}^T\boldsymbol{\Sigma}^{-1}-\sum_{i=1}^{p}\mathbf{Y}_{3,j,i}(\mathbf{B}_i^T\otimes\mathbf{A}_i^T)\boldsymbol{\Sigma}^{-1}\big](\mathbf{B}_j\otimes\mathbf{a}_j)\Big)
\end{aligned}
$$

$$+\operatorname{Tr}\Big(\sum_{j=1}^{p}\big[\boldsymbol{\Sigma}^{-1}\mathbf{Y}_{2,j}-\sum_{i=1}^{p}\boldsymbol{\Sigma}^{-1}(\mathbf{B}_i\otimes\mathbf{A}_i)\mathbf{Y}_{3,i,j}\big](\mathbf{b}_j^T\otimes\mathbf{A}_j^T)$$

$$+\sum_{j=1}^{p}\big[\mathbf{Y}_{2,j}^T\boldsymbol{\Sigma}^{-1}-\sum_{i=1}^{p}\mathbf{Y}_{3,j,i}(\mathbf{B}_i^T\otimes\mathbf{A}_i^T)\boldsymbol{\Sigma}^{-1}\big](\mathbf{b}_j\otimes\mathbf{A}_j)\Big)\Bigg\}$$

$$+O(\|\mathbf{a}\|^2)+O(\|\mathbf{b}\|^2)+O(\|\mathbf{s}\|^2)$$

$$=\quad\frac{1}{2}\Bigg\{\operatorname{Tr}\Big[\Big((T-1)\boldsymbol{\Sigma}-\mathbf{Y}_1+\sum_{i=1}^{p}(\mathbf{B}_i\otimes\mathbf{A})\mathbf{Y}_{2,i}^T+\sum_{i=1}^{p}\mathbf{Y}_{2,i}(\mathbf{B}_i^T\otimes\mathbf{A}_i^T)$$

$$-\sum_{i=1}^{p}\sum_{j=1}^{p}(\mathbf{B}_i\otimes\mathbf{A}_i)\mathbf{Y}_{3,i,j}(\mathbf{B}_j^T\otimes\mathbf{A}_j^T)\Big)^T\mathbf{s}\Big]$$

$$+\operatorname{Tr}\Big[\sum_{j=1}^{p}\Bigg(\Big((\mathbf{1}_n^T\otimes\mathbf{I}_m)\Big((\mathbf{B}_j\otimes\mathbf{J}_m)\odot\Big[2\boldsymbol{\Sigma}^{-1}\big(\mathbf{Y}_{2,j}-\sum_{i=1}^{p}((\mathbf{B}\otimes\mathbf{A})\mathbf{Y}_{3,i,j}\big)\Big]\Big)\cdot$$

$$(\mathbf{1}_n\otimes\mathbf{I}_m)\Big)^T\mathbf{a}_j\Bigg)\Big]$$

$$+\operatorname{Tr}\Big[\sum_{j=1}^{p}\Bigg((\mathbf{I}_n\otimes\mathbf{1}_m^T)\Big(\big[(\mathbf{I}_n\otimes\mathbf{A}_j^T)\big(\mathbf{Y}_{2,j}^T-\sum_{i=1}^{p}\mathbf{Y}_{3,i,j}(\mathbf{B}_i^T\otimes\mathbf{A}_i^T)\big)\boldsymbol{\Sigma}^{-1}$$

$$+\big(\mathbf{Y}_{2,j}^T-\sum_{i=1}^{p}\mathbf{Y}_{3,i,j}(\mathbf{B}_i^T\otimes\mathbf{A}_i^T)\big)\boldsymbol{\Sigma}^{-1}\cdot(\mathbf{I}_n\otimes\mathbf{A}_j^T)\big]\odot(\mathbf{J}_n\otimes\mathbf{I}_m)\Bigg)\cdot$$

$$(\mathbf{I}_n\otimes\mathbf{1}_m)\Big)^T\mathbf{b}_j\Bigg)\Big]\Bigg\}+O(\|\mathbf{a}\|^2)+O(\|\mathbf{b}\|^2)+O(\|\mathbf{s}\|^2)$$

Using Lemma 3.1, the last equality holds. It is obviously that $\mathbf{Y}_{3,i,j}=\mathbf{Y}_{3,j,i}^T$.

Define $\mathbf{Y}_{4,i}=\Big[\boldsymbol{\Sigma}^{-1}\big(\mathbf{Y}_{2,i}-(\sum_{j=1}^{p}(\mathbf{B}_j\otimes\mathbf{A}_j)\mathbf{Y}_{3,i,j}^T\big)\Big]$,

and using the fact that

$$\begin{aligned} & l(\mathbf{A}+\mathbf{a}, \mathbf{B}+\mathbf{b}, \boldsymbol{\Sigma}^{-1}+\mathbf{s}) - l(\mathbf{A}, \mathbf{B}, \boldsymbol{\Sigma}^{-1}) \\ = \; & \mathrm{Tr}(\sum_{i=1}^{p} D_{\mathbf{A}_i} l(\mathbf{A}, \mathbf{B}, \boldsymbol{\Sigma}^{-1})^T \mathbf{a}_i) + \mathrm{Tr}(\sum_{i=1}^{p} D_{\mathbf{B}_i} l(\mathbf{A}, \mathbf{B}, \boldsymbol{\Sigma}^{-1})^T \mathbf{b}_i) \\ & + \mathrm{Tr}(\sum_{i=1}^{p} D_{\boldsymbol{\Sigma}^{-1}} l(\mathbf{A}, \mathbf{B}, \boldsymbol{\Sigma}^{-1})^T \mathbf{s}) + O(\|\mathbf{a}\|^2) + O(\|\mathbf{b}\|^2) + O(\|\mathbf{s}\|^2), \end{aligned}$$

the results follow. □

3.3.1 Proposed algorithm for the general MAR(p) model

We aim to solve not just the MAR(p) model (3.2) with lag order $p = 1$ but also with lag order $p > 1$. From now onwards, we denote $\boldsymbol{\Sigma}^{-1}$ by $\boldsymbol{\Theta}$. Our algorithm has two parts. The former estimates the MAR(p) model with a general $\boldsymbol{\Theta}$ and the latter estimates the MAR(p) model under the structured covariance tensor, as stated in Chen, Xiao and Yang (2021).

Let $q = 1, .., Q$ be iteration number, $\mathbf{A} = (\mathbf{A}_1, \ldots, \mathbf{A}_p)$ and $\mathbf{B} = (\mathbf{B}_1, \ldots, \mathbf{B}_p)$. To fix the identifiability, we fix $\|\mathbf{A}_i\| = 1$ for $i = 1, \ldots, p$. We apply the concept of block coordinate gradient descent algorithm to our algorithm as follows.

1. General MAR(p) Estimation:

   (a) We solve $\mathbf{B} \otimes \mathbf{A}$ and $\boldsymbol{\Theta}$ based on the maximum likelihood estimates of the best fitted vector autoregressive model for the data. Then the projection method is extended to estimate $\mathbf{A}_i, \mathbf{B}_i, i = 1, \ldots, p$. These estimates $\mathbf{A}_i^{(0)}, \mathbf{B}_i^{(0)}, \boldsymbol{\Theta}^{(0)}$ are used as initial values.

   (b) Set iteration number $q = 1$.

   (c) Given $\mathbf{B}^{(q-1)}$ and $\boldsymbol{\Theta}^{(q-1)}$, solve

$$\mathbf{A}^{(q)} = \arg\min_{\mathbf{A}} -l(\mathbf{A}, \mathbf{B}^{(q-1)}, \boldsymbol{\Theta}^{(q-1)})$$

and renormalize $\mathbf{A}$ afterwards.

(d) Given $\mathbf{A}^{(q)}$ and $\mathbf{\Theta}^{(q-1)}$, solve

$$\mathbf{B}^{(q)} = \arg\min_{\mathbf{B}} -l(\mathbf{A}^{(q)}, \mathbf{B}, \mathbf{\Theta}^{(q-1)})$$

(e) Given $\mathbf{A}^{(q)}$ and $\mathbf{B}^{(q)}$, solve

$$\mathbf{\Theta}^{(q)} = \arg\min_{\mathbf{\Theta}} -l(\mathbf{A}^{(q)}, \mathbf{B}^{(q)}, \mathbf{\Theta})$$

(f) Set $q = q + 1$. Repeat Steps (c) and (e) until $\frac{\|\mathbf{A}^{(q)} - \mathbf{A}^{(q-1)}\|_F}{\max(1, \|\mathbf{A}^{(q)}\|_F)} < 1 \times 10^{-4}$, $\frac{\|\mathbf{B}^{(q)} - \mathbf{B}^{(q-1)}\|_F}{\max(1, \|\mathbf{B}^{(q)}\|_F)} < 1 \times 10^{-4}$ and $\frac{\|\mathbf{\Theta}^{(q)} - \mathbf{\Theta}^{(q-1)}\|_F}{\max(1, \|\mathbf{\Theta}^{(q)}\|_F)} < 1 \times 10^{-4}$.

2. Obtain initial values from VAR(p) MLE using the following steps:

(a) Perform the projection method on the i-th lag order coefficient matrices $\mathbf{C}_i$, $(i = 1, \ldots, p)$. i.e. $\min_{\mathbf{A}_i, \mathbf{B}_i} \|\mathbf{C}_i - \mathbf{B}_i \otimes \mathbf{A}_i\|_F^2$

(b) Normalize $\hat{\mathbf{A}}_i$: i.e. $\widehat{\mathbf{A}}_i \leftarrow \frac{\widehat{\mathbf{A}}_i}{\|\widehat{\mathbf{A}}_i\|}$ and $\widehat{\mathbf{B}}_i \leftarrow \widehat{\mathbf{B}}_i \cdot \|\widehat{\mathbf{A}}_i\|$

(c) Direct apply the precision matrix of the best fitted VAR(p) model as the initial value of MAR(p).

In the $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{\Theta}$ steps, we use the algorithm in Section 2.2.3 with the following change in Step 3:

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \nabla l(\mathbf{w}^{(k)})/t^{(k)}.$$

And the whole big algorithm forms a block coordinate gradient descent algorithm and its convergence will be discussed in Section 3.3.3

3.3.2 Proposed algorithm for the sparse graphical MAR(p) model

The proposed algorithm for solving $(\mathbf{A}, \mathbf{B}, \mathbf{\Theta})$ is the following:

1: Set $i_{\mathbf{A}} = i_{\mathbf{B}} = i_{\mathbf{\Theta}} = 1$ and the regularization parameter triplet $(\lambda_{\mathbf{A}}, \lambda_{\mathbf{B}}, \lambda_{\mathbf{\Theta}})$ to (0.01, 0.01, 0.01). Set up minimum and maximum for these regularization parameters and form a grid for $(\lambda_{\mathbf{A}}, \lambda_{\mathbf{B}}, \lambda_{\mathbf{\Theta}})$.

2: *(Initialization of parameters $\mathbf{A}, \mathbf{B}$ and $\mathbf{\Theta}$)* For each $(\lambda_{\mathbf{A}}, \lambda_{\mathbf{B}}, \lambda_{\mathbf{\Theta}})$ set of given values, set the outer iteration counter, q, to 1. When $i_{\mathbf{B}} = i_{\mathbf{\Theta}} = 1$, set the initial values of $\mathbf{A}$ $\mathbf{B}$ and $\mathbf{\Theta}$ as $\mathbf{A}^{(0)}, \mathbf{B}^{(0)}$ and $\mathbf{\Theta}^{(0)}$, which are the maximum likelihood estimates of (2.3), otherwise use a warm start in the following way:

(a) When $i_{\mathbf{A}} = 1, i_{\mathbf{B}} = 1$ and $i_{\mathbf{\Theta}} > 1$, set the initial value as $\mathbf{A}_{(1,1,1)}, \mathbf{B}_{(1,1,1)}$ and $\mathbf{\Theta}_{(1,1,1)}$.

(b) When $i_{\mathbf{A}} = 1$, for any $i_{\mathbf{B}} > 1$, and any $i_{\mathbf{\Theta}}$, set the initial value from $\mathbf{A}_{(1,i_{\mathbf{B}}-1,i_{\mathbf{\Theta}})}$, $\mathbf{B}_{(1,i_{\mathbf{B}}-1,i_{\mathbf{\Theta}})}$ and $\mathbf{\Theta}_{(1,i_{\mathbf{B}}-1,i_{\mathbf{\Theta}})}$.

(c) For any $i_{\mathbf{A}} > 1, i_{\mathbf{B}} > 1$, $i_{\mathbf{\Theta}}$, set the initial value from $\mathbf{A}_{(i_{\mathbf{A}}-1,i_{\mathbf{B}},i_{\mathbf{\Theta}})}$, $\mathbf{B}_{(i_{\mathbf{A}}-1,i_{\mathbf{B}},i_{\mathbf{\Theta}})}$ and $\mathbf{\Theta}_{(i_{\mathbf{A}}-1,i_{\mathbf{B}},i_{\mathbf{\Theta}})}$.

3: *(Block Coordinate Gradient Descent Algorithm)* Given $\mathbf{B}^{(q-1)}$ and $\mathbf{\Theta}^{(q-1)}$, solve $\mathbf{A}^{(q)}$ from the following:

$$\mathbf{A}^{(q)} = \arg\min_{\mathbf{A}} -l(\mathbf{A}, \mathbf{B}, \mathbf{\Theta}^{(q-1)}) + T \sum_{i,j,k} \lambda_{\mathbf{A}} |a_{ij,k}| \tag{3.11}$$

$$\mathbf{A}_k^{(q)} \leftarrow \mathbf{A}_k^{(q)} / \|\mathbf{A}_k^{(q)}\| \text{ for } k = 1, \ldots, p.$$

4: Given $\mathbf{A}^{(q-1)}$ and $\mathbf{\Theta}^{(q-1)}$, solve $\mathbf{B}^{(q)}$ from the following.

$$\mathbf{B}^{(q)} = \arg\min_{\mathbf{B}} -l(\mathbf{A}, \mathbf{B}, \mathbf{\Theta}^{(q-1)}) + T \sum_{i,j,k} \lambda_{\mathbf{B}} |b_{ij,k}| \tag{3.12}$$

5: Given $\mathbf{A}^{(q-1)}, \mathbf{B}^{(q-1)}$, solve $\mathbf{\Theta}^{(q)}$ from the following.

$$\mathbf{\Theta}^{(q)} = \arg\min_{\mathbf{\Theta}} -l(\mathbf{A}^{(q)}, \mathbf{B}^{(q)}, \mathbf{\Theta}) + T \sum_{i \neq j} \lambda_{\mathbf{\Theta}} |\theta_{ij}| \tag{3.13}$$

6: Set $q = q + 1$ and repeat Steps 3 to 5 until the following stopping criterion is fulfilled:
$\frac{\|\mathbf{A}^{(q)} - \mathbf{A}^{(q-1)}\|}{\max(1, \|\mathbf{A}^{(q)}\|)} \leq 1 \times 10^{-4}, \frac{\|\mathbf{B}^{(q)} - \mathbf{B}^{(q-1)}\|}{\max(1, \|\mathbf{B}^{(q)}\|)} \leq 1 \times 10^{-4}$ and
$\frac{\|\mathbf{\Theta}^{(q)} - \mathbf{\Theta}^{(q-1)}\|}{\max(1, \|\mathbf{\Theta}^{(q)}\|)} \leq 1 \times 10^{-4}$.

7: Set the solutions $\mathbf{A}_{(i_{\mathbf{A}}, i_{\mathbf{B}}, i_{\mathbf{\Theta}})} = \mathbf{A}^{(q)}$, $\mathbf{B}_{(i_{\mathbf{A}}, i_{\mathbf{B}}, i_{\mathbf{\Theta}})} = \mathbf{B}^{(q)}$, $\mathbf{\Theta}_{(i_{\mathbf{A}}, i_{\mathbf{B}}, i_{\mathbf{\Theta}})} = \mathbf{\Theta}^{(q)}$ and set $(i_{\mathbf{A}}, i_{\mathbf{B}}, i_{\mathbf{\Theta}})$ to next grid value by $i_{\mathbf{A}} = i_{\mathbf{A}} + 1$, $i_{\mathbf{B}} = i_{\mathbf{B}} + 1$ and/ or $i_{\mathbf{\Theta}} = i_{\mathbf{\Theta}} + 1$ and go to Step 2. Repeat Steps 2 to 6 until all grid points of $(i_{\mathbf{A}}, i_{\mathbf{B}}, i_{\mathbf{\Theta}})$ are used.

8: The final model is selected based on minimum BIC among the all grid estimates.

In the $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{\Theta}$ steps, we use the algorithm in Section 2.2.3. Running the above sparse graphical MAR(p) model algorithm is computationally costly because one may run Steps 1 to 8 of the above algorithm for $100 \times 100 \times 100$ grid points. Based on the experience of fitting a sparse graphical VAR(p) model for the simulated samples in Chapter 2, the regularization parameters of coefficient and precision matrices have very small mean values and small deviations. Refer to Tables 2.3 and 2.5 for details. Therefore, it is not necessary to run all grid points for obtaining the optimal sparse model.

We consider a strategy to reduce the number of grid points for running the sparse MAR(p) model. We first consider the step size for a grid is 0.05, instead of 0.01.

Then the number of grid points is reduced to $20 \times 20 \times 20 = 8000$. Conduct the above algorithm from Steps 1 to 7 for these 8000 grid points. We extract the estimated triplets with total positive rates (TPRs) and total negative rates (TNRs) of $\mathbf{A}, \mathbf{B}, \mathbf{\Theta}$) and plot a three-dimensional graph of $(\lambda_{\mathbf{A}}, \lambda_{\mathbf{B}}, \lambda_{\mathbf{\Theta}})$ from the extracted estimated triplets. Then the plotted ranges of $(\lambda_{\mathbf{A}}, \lambda_{\mathbf{B}}, \lambda_{\mathbf{\Theta}})$ suggest smaller ranges from 0.01 to 1 for the grid of $(\lambda_{\mathbf{A}}, \lambda_{\mathbf{B}}, \lambda_{\mathbf{\Theta}})$ for sparse graphical MAR(p) model estimation. For simplicity, we consider plotting the diagram where all TPRs and TNRs are greater than 75% for the three-dimensional plot and determining the ranges of the regularization parameter triplet for using a 0.01 step size afterwards. Normally, we take a bigger range than the range observed from the range observed in a three-dimensional plot.

3.3.3 Convergence of the algorithms

In this section, we discuss the convergence of the algorithm for MAR(p) model estimation and sparse graphical MAR(p) model estimation.

Theorem 3.2. *The MAR(p) model estimation algorithm is a block coordinate gradient descent algorithm and is convergent.*

Proof: We modify the proof of Theorem 2.1 with the following:

1. Change the objective function to $F(x_1, x_2, x_3)$ instead of $F(x_1, x_2)$.

2. Set the penalty function of $P_1(x_1) = P_2(x_2) = 0$.

Results follow. □

Theorem 3.3. *The sparse graphical MAR(p) model estimation algorithm is a block coordinate gradient descent algorithm and is convergent.*

Proof: We formulate the minimization problem as

$$\arg \min_{\mathbf{x}=(x_1,x_2,x_3)} F(x_1, x_2, x_3) := f(x_1, , x_2, x_3) + P_1(x) + P_2(x) + P_3(x).$$

Using the proof of Theorem 2.1, results follow. □

3.4 Simulation Study

3.4.1 Evaluation measures

We conducted 100 samples to evaluate the performance of our algorithm. For a MAR(p) model, let $\mathbf{A} = [\mathbf{A}_1, \ldots, \mathbf{A}_p]$, $\mathbf{B} = [\mathbf{B}_1, \ldots, \mathbf{B}_p]$ and $\mathbf{B} \otimes \mathbf{A} = [\mathbf{B}_1 \otimes \mathbf{A}_1, \ldots, \mathbf{B}_p \otimes \mathbf{A}_p]$. The following are performance evaluation metrics:

1. $RMSE_{\mathbf{A}} = \dfrac{E(\|\hat{\mathbf{A}} - \mathbf{A}\|)}{m\sqrt{p}}$

2. $RMSE_{\mathbf{B}} = \dfrac{E(\|\hat{\mathbf{B}} - \mathbf{B}\|)}{n\sqrt{p}}$

3. $RMSE_{\mathbf{B}\otimes\mathbf{A}} = \dfrac{E(\|\hat{\mathbf{B}} \otimes \hat{\mathbf{A}} - \mathbf{B} \otimes \mathbf{A}\|)}{mn\sqrt{p}}$

4. $RMSE_{\mathbf{\Theta}} = \dfrac{E(\|\widehat{\mathbf{\Theta}} - \mathbf{\Theta}\|)}{mn}$

They are root mean squares (RMSE) per cell of the coefficient matrices and precision matrices.

3.4.2 The general MAR(p) model

We conducted a simulation study nine MAR(p) ($p = 1, 2, 3$) stable models with three combinations of dimensions $(m, n) = (3, 2), (6, 4), (9, 6)$ at length (T) 200, 500 and 2000 and the models were the following:

1. Models p802, p813 and p816: MAR(1) model, i.e. $\mathbf{X}_t = \mathbf{A}_1^{(j)} \mathbf{X}_{t-1} (\mathbf{B}_1^{(j)})^T + \mathbf{E}_t$, where $\mathbf{E}_t \sim N(0, (\mathbf{\Theta}_1^{(j)})^{-1})$, for $j =$ p802, p813 and p816;

2. Models p822, p818 and p824: MAR(2) model, i.e. $\mathbf{X}_t = \mathbf{A}_1^{(j)}\mathbf{X}_{t-1}(\mathbf{B}_1^{(j)})^T + \mathbf{A}_2^{(j)}\mathbf{X}_{t-2}(\mathbf{B}_2^{(j)})^T + \mathbf{E}_t$, where $\mathbf{E}_t \sim N(0, (\mathbf{\Theta}_2^{(j)})^{-1})$ for j =p822, p818 and p824; and

3. Models p16, p819 and p825: MAR(3) model, i.e. $\mathbf{X}_t = \mathbf{A}_1^{(j)}\mathbf{X}_{t-1}(\mathbf{B}_1^{(j)})^T + \mathbf{A}_2^{(j)}\mathbf{X}_{t-2}(\mathbf{B}_2^{(j)})^T + \mathbf{A}_3^{(j)}\mathbf{X}_{t-3}(\mathbf{B}_3^{(j)})^T + \mathbf{E}_t$, where $\mathbf{E}_t \sim N(0, (\mathbf{\Theta}_3^{(j)})^{-1})$, for j =p16, p819 and p825.

We selected the spectral radii of the left and right coefficient matrices ($\mathbf{A}_i^{(j)}$ and $\mathbf{B}_i^{(j)}$) between 0 and 1. Entries of the left and right coefficient matrices ($\mathbf{A}_i^{(j)}$ and $\mathbf{B}_i^{(j)}$) are randomly generated from normal distributions with mean zero and standard deviation randomly chosen from a value between 0 and 5 and the randomly generated matrices are then divided by their largest absolute eigenvalues times the spectral radius. Note that the selection of the spectral radii were chosen in the way that the generated time series satisfies the following stability conditions:

1. When $p = 1$, $\rho(\mathbf{A}) \cdot \rho(\mathbf{B}) < 1$, where $\rho(\mathbf{A})$ is the spectral radius of $\rho(\mathbf{A})$.

2. When $p = 2$, the largest modulus of eigenvalues of

$$\mathcal{A}_2 = \begin{bmatrix} \mathbf{B}_1 \otimes \mathbf{A}_1 & \mathbf{B}_2 \otimes \mathbf{A}_2 \\ \mathbf{I}_{mn} & \mathbf{0} \end{bmatrix}$$

   is less than 1.

3. When $p = 3$, the largest modulus of eigenvalues of

$$\mathcal{A}_3 = \begin{bmatrix} \mathbf{B}_1 \otimes \mathbf{A}_1 & \mathbf{B}_2 \otimes \mathbf{A}_2 & \mathbf{B}_3 \otimes \mathbf{A}_3 \\ \mathbf{I}_{mn} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{mn} & \mathbf{0} \end{bmatrix}$$

   is less than 1.

Note that $\|\mathbf{A}_i^{(j)}\| = 1$ for $i = 1, 2, 3$ and all j to fix the identification problem.

Models p802, p813 and p816 have dimensions of $(m, n) = (3, 2), (6, 4), (9, 6)$ respectively. Since they are lag order 1 models, we considered the product of their spectral radii of $\mathbf{A}_1$ and $\mathbf{B}_1$ for checking the stability of the time series. They are 0.29, 0.35 and 0.33 respectively. All are less than 1. Therefore, the model is stable. Their respective determinants of $\mathbf{\Theta}$ are 0.0005, 0.6996 and 1.2454.

The lag order 2 models p822, p818 and p824 have dimensions of $(m, n) = (3, 2), (6, 4), (9, 6)$ respectively. Since their lag orders are greater than 1, we examined matrix $\mathcal{A}_2$ for checking the stability of the time series. The respective moduli of eigenvalues of $\mathcal{A}_2$ range from 0.4924 to 0.5704, 0.0054 to 0.7556 and 0.1336 to 0.5867. All are less than 1. Therefore, the model is stable. Their respective determinants of $\mathbf{\Theta}$ are 0.0097, 0.0020 and 0.0168.

The lag order 3 models p16, p823 and p825 have dimensions of $(m, n) = (3, 2)$, $(6, 4), (9, 6)$ respectively. Since their lag orders are greater than 1, we examined the eigenvalues of matrix $\mathcal{A}_3$ for checking the stability of the time series. The respective moduli of eigenvalues of $\mathcal{A}_3$ range from 0.0825 to 0.7057, 0.1050 to 0.8427 and 0.0404 to 0.5500. All are less than 1. Therefore, the model is stable. Their respective determinants of $\mathbf{\Theta}$ are 6.38, 69.02 and 42.67.

Table 3.1 gave the root mean squares of errors of the left row-wise interactions and right column-wise interactions coefficient matrices, Kronecker products of the autoregressive coefficient matrices and precision matrices. The majority of the RMSEs of the coefficients $\mathbf{A}$ and $\mathbf{B}$ are less than 0.1. Only a few RMSEs of these coefficients are slightly greater than 0.1. This indicates the estimation over $\mathbf{A}$ and $\mathbf{B}$ are good.

The mean RMSE of the coefficients $\mathbf{B} \otimes \mathbf{A}$ for MAR(1) models ranged from 0.044 to 0.0437, while the mean RMSE of the precision matrix was from 0.0081 to 0.1454.

In the cases of MAR(2) models, the mean RMSE($\mathbf{B} \otimes \mathbf{A}$), ranging from 0.0044 to 0.0389, were in similar magnitudes as the selected MAR(1) models but the mean RMSE($\mathbf{\Theta}$), ranging from 0.0140 to 0.2792, were slightly larger than the MAR(1) models. In the cases of MAR(3) models, the mean RMSE($\mathbf{B} \otimes \mathbf{A}$), ranging from 0.0044 to 0.0399, were in similar magnitudes as the selected MAR(1) and MAR(2) models but the mean RMSE($\mathbf{\Theta}$) , ranging from 0.0486 to 0.4015, were slightly larger in higher lag MAR(3) models. As the length of the time series increased, all RMSEs of the same model were getting smaller and the corresponding standard deviations were smaller. The longer the time series fitted, the smaller the mean RMSEs of the model were. When the dimensions of $\mathbf{A}$ and $\mathbf{B}$ in the model increased, the RMSEs among the models had similar values. The small values of RMSE of coefficients and covariance matrices indicated that the proposed algorithm worked satisfactorily.

3.4.3 Comparison of the general MAR model and the existing MAR model

We simulated 100 random samples from our general MAR model and the existing MAR model, proposed by Chen et al. (2021), and fitted the data with these two models for comparison. The existing MAR model has a structured covariance tensor product and lag order one. For simplicity, it is named as SCT MAR(1) model. The covariance has the form, $\mathbf{\Sigma}_c^{(j)} \otimes \mathbf{\Sigma}_r^{(j)}$. So the precision matrix can be written as $\mathbf{\Theta} = \mathbf{\Sigma}^{-1} = \mathbf{\Sigma}_c^{-1} \otimes \mathbf{\Sigma}_r^{-1}$, i.e. $\mathbf{\Theta} = \mathbf{\Theta}_c \otimes \mathbf{\Theta}_r$ with $\|\mathbf{\Theta}_r^{-1}\| = 1$.

As the existing MAR model has lag order one. Therefore, we selected a general MAR(1) model for comparison. Both our MAR(p) algorithm and the maximum likelihood estimation under the structured covariance tensor (MLESCT) algorithm in Chen et al. (2021) were used to estimate the models. The following are the selected general MAR(1) stable model and three stable SCT MAR(1) models:

1. Model p807: $\mathbf{X}_t = \mathbf{A}^{(j)}\mathbf{X}_{t-1}(\mathbf{B}^{(j)})^T + \mathbf{E}_t$ with $\mathbf{E}_t \sim N(0, (\mathbf{\Theta}^{(j)})^{-1})$, for $j =$

Table 3.1: Root mean squares of errors of estimated autoregressive coefficients and precision matrices of MAR(p) models. (*All models used VAR(p) MLE as initial values, except the case of Model p825 and T=200. Its initial values were two arbitrary coefficient matrices and an identity precision matrix.)

Model	VAR				RMSE($\mathbf{A}$)		RMSE($\mathbf{B}$)		RMSE($\mathbf{B} \otimes \mathbf{A}$)		RMSE($\mathbf{\Theta}$)	
	order	m	n	Length	mean	sd	mean	sd	mean	sd	mean	sd
p802	1	3	2	200	0.1013	0.0301	0.0797	0.0315	0.0437	0.0107	0.0278	0.0070
				500	0.0617	0.0177	0.0508	0.0197	0.0272	0.0064	0.0157	0.0031
				2000	0.0299	0.0082	0.0227	0.0094	0.0127	0.0029	0.0081	0.0018
p813	1	6	4	200	0.0563	0.0062	0.0865	0.0186	0.0221	0.0027	0.1047	0.0082
				500	0.0342	0.0045	0.0485	0.0100	0.0129	0.0014	0.0551	0.0030
				2000	0.0172	0.0020	0.0236	0.0044	0.0063	0.0007	0.0249	0.0013
p816	1	9	6	200	0.0483	0.0041	0.1027	0.0134	0.0176	0.0013	0.1454	0.6408
				500	0.0276	0.0021	0.0523	0.0069	0.0094	0.0007	0.0605	0.0023
				2000	0.0132	0.0011	0.0235	0.0029	0.0044	0.0003	0.0249	0.0006
p822	2	3	2	200	0.0762	0.0162	0.0617	0.0182	0.0389	0.0071	0.0536	0.0154
				500	0.0471	0.0099	0.0360	0.0118	0.0239	0.0042	0.0303	0.0066
				2000	0.0228	0.0040	0.0180	0.0053	0.0116	0.0018	0.0140	0.0031
p818	2	6	4	200	0.0320	0.0046	0.0613	0.0100	0.0192	0.0021	0.1188	0.0135
				500	0.0197	0.0028	0.0352	0.0054	0.0114	0.0012	0.0579	0.0054
				2000	0.0095	0.0013	0.0170	0.0025	0.0056	0.0005	0.0254	0.0018
p824	2	9	6	200	0.0487	0.0043	0.0754	0.0101	0.0172	0.0014	0.2792	0.0320
				500	0.0291	0.0024	0.0357	0.0039	0.0095	0.0007	0.1017	0.0098
				2000	0.0139	0.0012	0.0161	0.0018	0.0044	0.0003	0.0381	0.0023
p16	3	3	2	200	0.1037	0.0214	0.0718	0.0226	0.0399	0.0071	0.2264	0.0777
				500	0.0661	0.0131	0.0435	0.0124	0.0246	0.0040	0.1246	0.0384
				2000	0.0324	0.0068	0.0210	0.0066	0.0119	0.0022	0.0576	0.0142
p819	3	6	4	200	0.0398	0.0036	0.0507	0.0070	0.0173	0.0014	0.2654	0.0488
				500	0.0234	0.0022	0.0290	0.0037	0.0101	0.0008	0.1190	0.0212
				2000	0.0116	0.0012	0.0139	0.0017	0.0049	0.0004	0.0484	0.0061
p825	3	9	6	200*	0.0590	0.0041	0.0822	0.0082	0.0173	0.0012	0.4015	0.0447
				500	0.0351	0.0025	0.0384	0.0033	0.0094	0.0006	0.1326	0.0116
				2000	0.0172	0.0014	0.0164	0.0014	0.0044	0.0003	0.0486	0.0034

p807.

2. Models p802, p813 and p816: $\mathbf{X}_t = \mathbf{A}^{(j)}\mathbf{X}_{t-1}(\mathbf{B}^{(j)})^T + \mathbf{E}_t$ with $\mathbf{E}_t \sim N(0, \mathbf{\Sigma}_c^{(j)} \otimes \mathbf{\Sigma}_r^{(j)})$, for j = p802, p813 and p816.

The (j) index indicates the model number. For simplicity, we drop this index for discussion. The general MAR(1) model, p807, has dimensions of $(m, n) = (9, 6)$ and its product of the spectral radii $\rho(\mathbf{A}) \cdot \rho(\mathbf{B})$ is 0.2276. The determinant of $\mathbf{\Theta}$ is 0.7903. The Models, p802, p813 and p816 are SCT MAR(1) models with respective dimensions (m, n) as $(3, 2), (6, 4)$ and $(9, 6)$. Their products of the spectral radii $\rho(\mathbf{A}) \cdot \rho(\mathbf{B})$ are 0.912, 0.2214 and 0.2276 respectively. Their covariances $\mathbf{\Sigma}_r$ have

Frobenius norm, one. The determinants of covariances $\mathbf{\Sigma}_r$ are 0.1231, 0.0046 and 5.0707×10^{-5} respectively, while the determinants of covariances $\mathbf{\Sigma}_c$ are 50.3155, 38.329 and 712.3589. Since these models have the products of the spectral radii $\rho(\mathbf{A}) \cdot \rho(\mathbf{B})$ less than 1. They are all stable. To fix the identification problem, their left row-wise interactions coefficient matrices have unit norms, i.e. $\|A\| = 1$.

The RMSEs on $\mathbf{B} \otimes \mathbf{A}$ and $\mathbf{\Theta}$ of the above models are tabulated in Table 3.2 for comparison. Our algorithm performed better on the general MAR(1) model, p807, and the MLESCT algorithm performed better on SCT MAR(1) models, p802, p813 and p816. For Model p807, the RMSEs of $\mathbf{B} \otimes \mathbf{A}$ estimated by the MLESCT algorithm are 3 to 4 times bigger than that estimated by our algorithm. In addition, the RMSEs of $\mathbf{\Theta}$ estimated by the MLESCT algorithm were at least 1.1 for these three series with different lengths, while the RMSEs estimated by our algorithm shrunk as the length of the time series increased. For the cases of three SCT MAR(1) models, p802, p813 and p816, the RMSEs of $\mathbf{B} \otimes \mathbf{A}$ estimated by our algorithm were slightly larger than that of RMSEs estimated by MLESCT algorithm. The differences were less than 0.04. When $\mathbf{\Theta}$ RMSEs were examined, our algorithm only generated at most about 0.08 difference. This indicated that fitting a general MAR(p) model was better in general.

3.4.4 Sparse graphical MAR(p) model

We chose the following four sparse graphical MAR(p) stable models with dimensions of $\mathbf{A}$ and $\mathbf{B}$, $(m, n) = (6, 4)$ and lengths (T) 200, 500, 2000:

1. Model p3004: MAR(1) model with sparsity in $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{\Theta}$ are 0.47, 0.43 and 0.84 respectively. The graph corresponding to the $\mathbf{\Theta}$ is the first variable connecting all the other variables.

2. Model p3005: MAR(1) model with sparsity in $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{\Theta}$ are 0.56, 0.75 and

Table 3.2: Comparison of MAR(1) estimates between our proposed and MLESCT algorithms

Model			Length	RMSE($\mathbf{B} \otimes \mathbf{A}$)				RMSE($\mathbf{\Theta}$)			
				Our algorithm		MLESCT algorithm		Our algorithm		MLESCT algorithm	
	m	n		mean	sd	mean	sd	mean	sd	mean	sd
A MAR(1) model with general $\mathbf{\Theta}$											
p807	9	6	200	0.0043	0.0003	0.0149	0.0011	0.8460	0.0692	1.1084	0.0007
			500	0.0024	0.0002	0.0092	0.0007	0.3231	0.0255	1.1087	0.0004
			2000	0.0011	0.0001	0.0045	0.0004	0.1257	0.0083	1.1089	0.0002
Three SCT MAR(1) models with $\mathbf{\Theta}_c^{-1} \otimes \mathbf{\Theta}_r^{-1}$											
p802	3	2	200	0.0437	0.0107	0.0432	0.0106	0.0278	0.0070	0.0168	0.0055
			500	0.0272	0.0064	0.0271	0.0064	0.0157	0.0031	0.0100	0.0027
			2000	0.0127	0.0029	0.0127	0.0029	0.0081	0.0018	0.0052	0.0016
p813	6	4	200	0.0221	0.0027	0.0209	0.0025	0.1047	0.0082	0.0268	0.0045
			500	0.0129	0.0014	0.0125	0.0014	0.0551	0.0030	0.0165	0.0024
			2000	0.0063	0.0007	0.0063	0.0007	0.0249	0.0013	0.0079	0.0011
p816	9	6	200	0.0176	0.0013	0.0145	0.0010	0.1454	0.0080	0.0172	0.0020
			500	0.0094	0.0007	0.0089	0.0006	0.0605	0.0023	0.0104	0.0011
			2000	0.0044	0.0003	0.0043	0.0003	0.0249	0.0006	0.0050	0.0005

0.84 respectively. $\mathbf{B} = \mathbf{I}$ and $\mathbf{A}$ and $\mathbf{\Theta}$ are tridiagonal matrices.

3. Model p3006: MAR(2) model with sparsity in $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{\Theta}$ are 0.47, 0.38 and 0.71 respectively. The sparseness of $\mathbf{A}_i$, $\mathbf{B}_i$, $(i = 1, 2)$ and $\mathbf{\Theta}$ was randomly chosen.

4. Model p3007: MAR(3) model with sparsity in $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{\Theta}$ are 0.45, 0.40 and 0.71 respectively. Values of $\mathbf{A}_i$, $\mathbf{B}_i$, $(i = 1, 2, 3)$ of $\mathbf{\Theta}$ are randomly generated and the sparseness of $\mathbf{A}_i$, $\mathbf{B}_i$, $(i = 1, 2, 3)$ of $\mathbf{\Theta}$ is the same as that of Model p3006.

We developed a strategy to obtain the sparse graphical MAR(p) model at a minimal computation cost. i.e. Not all models with $\lambda_{\mathbf{A}}$, $\lambda_{\mathbf{B}}$ and $\lambda_{\mathbf{\Theta}}$ ranging from 0.01 to 1 were run. Based on the experience of the sparse graphical VAR(p) models in 2, we found that the optimal regularization parameters were quite small for our examples in the simulation study. We then set up an experiment for getting ranges of regularization parameters in our sparse graphical MAR(p) models. The models with regularization parameters ranging from a larger grid size from 0.01 to 1 were

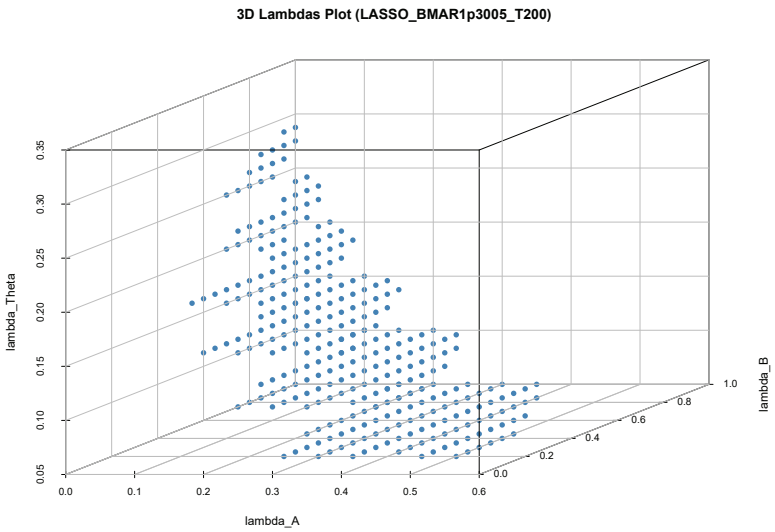


Figure 3.1: Plots of regularization parameters $\lambda_{\mathbf{A}}$, $\lambda_{\mathbf{B}}$ and $\lambda_{\mathbf{\Theta}}$ for Model p3005 with length $T = 200$ having $TPR_{\mathbf{A}}$, $TNR_{\mathbf{A}}$, $TPR_{\mathbf{B}}$, $TNR_{\mathbf{B}}$, $TPR_{\mathbf{\Theta}}$ and $TNR_{\mathbf{\Theta}}$ greater than or equal to 0.75

Table 3.3: Maximum values of regularization parameters for sparse graphical MAR(p) models runs

Model	$\max \lambda_{\mathbf{A}}$	$\max \lambda_{\mathbf{B}}$	$\max \lambda_{\mathbf{\Theta}}$
p3004	0.4	0.5	1
p3005	0.65	1	0.4
p3006	0.3	0.35	0.1
p3007	0.25	0.35	1

run. We found that the regularization parameters for models with high total positive rates (TPR) and total negative rates (TNR) in $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{\Theta}$ formed clusters. Plots of $\lambda_{\mathbf{A}}$, $\lambda_{\mathbf{B}}$ and $\lambda_{\mathbf{\Theta}}$ for Model p3005 were shown in Figures 3.1, 3.2 and 3.3. We used these diagrams to set up the maximum values of regularization parameters for sparse MAR(p) models running. The ranges of $\lambda_{\mathbf{A}}$, $\lambda_{\mathbf{B}}$ and $\lambda_{\mathbf{\Theta}}$ were set as in Table 3.3.

Table 3.4 gave the mean optimal size of regularization parameters, $\lambda_{\mathbf{A}}$, $\lambda_{\mathbf{B}}$ and $\lambda_{\mathbf{\Theta}}$, and their standard deviations, as well as the TPRs and TNRs for $\mathbf{A}$, $\mathbf{B}$ and

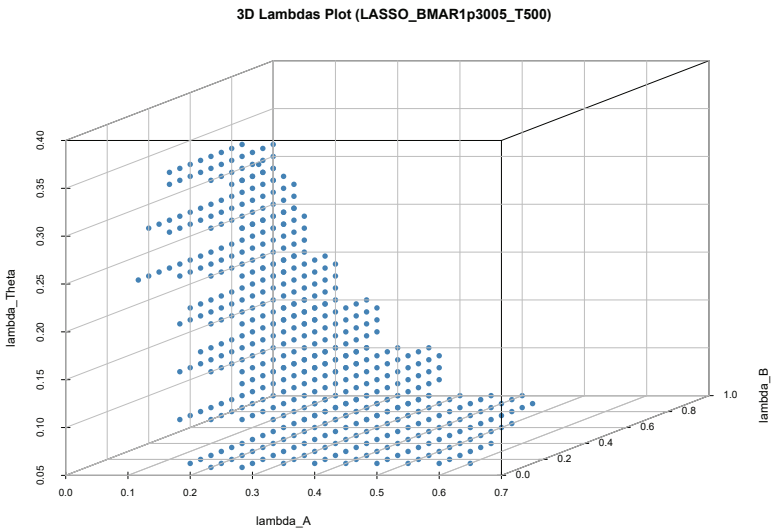


Figure 3.2: Plots of regularization parameters $\lambda_{\mathbf{A}}$, $\lambda_{\mathbf{B}}$ and $\lambda_{\mathbf{\Theta}}$ for Model p3005 with length $T = 500$ having $TPR_{\mathbf{A}}$, $TNR_{\mathbf{A}}$, $TPR_{\mathbf{B}}$, $TNR_{\mathbf{B}}$, $TPR_{\mathbf{\Theta}}$ and $TNR_{\mathbf{\Theta}}$ greater than or equal to 0.75

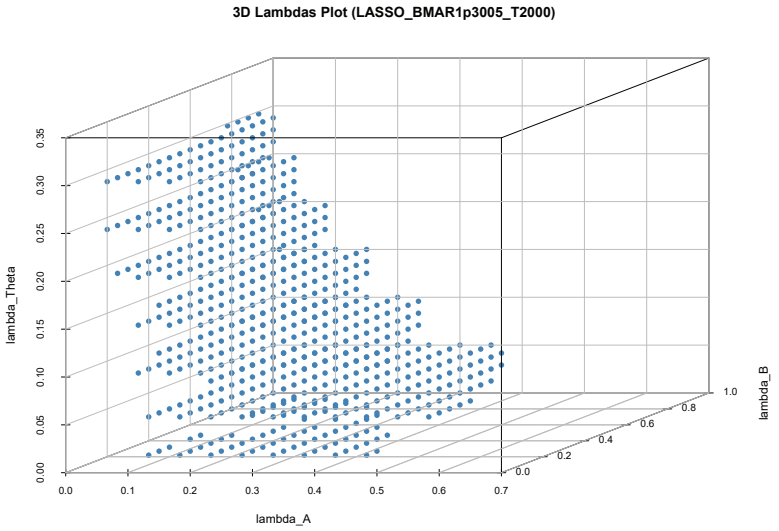


Figure 3.3: Plots of regularization parameters $\lambda_{\mathbf{A}}$, $\lambda_{\mathbf{B}}$ and $\lambda_{\mathbf{\Theta}}$ for Model p3005 with length $T = 2000$ having $TPR_{\mathbf{A}}$, $TNR_{\mathbf{A}}$, $TPR_{\mathbf{B}}$, $TNR_{\mathbf{B}}$, $TPR_{\mathbf{\Theta}}$ and $TNR_{\mathbf{\Theta}}$ greater than or equal to 0.75

Table 3.4: Mean regularized parameters values, total positive and negative rates for coefficients and precision matrices of LASSO MAR(p) Models

Model	T	λ_A		λ_B		$\lambda_{\mathbf{\Theta}}$	
		mean	sd	mean	sd	mean	sd
p3004	200	0.227	0.052	0.067	0.025	0.122	0.012
	500	0.153	0.039	0.051	0.016	0.081	0.009
	2000	0.083	0.018	0.03	0.009	0.043	0.005
p3005	200	0.161	0.051	0.165	0.051	0.034	0.005
	500	0.122	0.035	0.118	0.029	0.021	0.003
	2000	0.079	0.019	0.071	0.018	0.01	0
p3006	200	0.099	0.019	0.057	0.022	0.03	0.006
	500	0.065	0.013	0.035	0.014	0.02	0.001
	2000	0.036	0.006	0.024	0.008	0.01	0
p3007	200	0.093	0.022	0.069	0.024	0.03	0.007
	500	0.051	0.014	0.041	0.014	0.019	0.003
	2000	0.025	0.005	0.024	0.007	0.01	0

Model	T	$TPR_{\mathbf{A}}$	$TNR_{\mathbf{A}}$	$TPR_{\mathbf{B}}$	$TNR_{\mathbf{B}}$	$TPR_{\mathbf{\Theta}}$	$TNR_{\mathbf{\Theta}}$
p3004	200	0.753	0.889	0.93	0.677	1	0.987
	500	0.852	0.885	0.918	0.744	1	0.989
	2000	0.956	0.91	0.91	0.779	1	0.991
p3005	200	0.998	0.763	1	0.88	1	0.806
	500	1	0.772	1	0.908	1	0.798
	2000	1	0.799	1	0.937	1	0.787
p3006	200	0.726	0.91	0.936	0.643	0.934	0.824
	500	0.925	0.898	0.997	0.627	0.999	0.82
	2000	1	0.9	1	0.712	1	0.812
p3007	200	0.497	0.928	0.72	0.776	0.932	0.818
	500	0.736	0.861	0.934	0.667	0.999	0.81
	2000	0.954	0.818	1	0.684	1	0.813

$\mathbf{\Theta}$. The mean $\lambda_{\mathbf{A}}$, $\lambda_{\mathbf{B}}$ and $\lambda_{\mathbf{\Theta}}$ were less than 0.1 with standard deviations being less than 0.1. Their values were far from the maximum values set for the sparse graphical MAR(p) model runs. This indicated that our strategy was adequate for sparse graphical MAR(p) models running. Most of their TPRs and TNRs for $\mathbf{A}$, $\mathbf{B}$ and $\lambda_{\mathbf{\Theta}}$ varied from 0.7 to 1.0. The TNRs for $\mathbf{B}$ for Models p3006 and p3007 were larger than 0.6. These might be due to the low dimension of $\mathbf{B}$, 4, and higher lag order, i.e. higher complexity of the model. As the length of the time series increased, the optimal size of regularization parameters decreased and the TPRs and TNRs for $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{\Theta}$ were similar in magnitude.

Table 3.5: RMSEs of coefficients and precision matrices of LASSO MAR(p) Models

Model	Lag	T	$RMSE_{\mathbf{A}}$		$RMSE_{\mathbf{B}}$		$RMSE_{\mathbf{B}\otimes\mathbf{A}}$		$RMSE_{\mathbf{\Theta}}$	
			mean	sd	mean	sd	mean	sd	mean	sd
p3004	1	200	0.0274	0.0056	0.0660	0.0153	0.0174	0.0029	0.0275	0.0032
		500	0.0186	0.0035	0.0420	0.0097	0.0116	0.0018	0.0192	0.0022
		2000	0.0096	0.0016	0.0221	0.0055	0.0061	0.0009	0.0105	0.0013
p3005	1	200	0.0278	0.0059	0.0521	0.0133	0.0159	0.0028	0.1738	0.0213
		500	0.0172	0.0034	0.0342	0.0084	0.0102	0.0017	0.1203	0.0125
		2000	0.0086	0.0020	0.0193	0.0048	0.0054	0.0011	0.0651	0.0051
p3006	2	200	0.0873	0.0158	0.0794	0.0188	0.0222	0.0029	0.1694	0.0195
		500	0.0488	0.0082	0.0427	0.0070	0.0127	0.0016	0.1229	0.0079
		2000	0.0220	0.0035	0.0218	0.0039	0.0061	0.0007	0.0701	0.0035
p3007	3	200	0.1203	0.0103	0.0920	0.0143	0.0215	0.0019	0.1747	0.0202
		500	0.0870	0.0157	0.0522	0.0135	0.0137	0.0016	0.1216	0.0131
		2000	0.0386	0.0075	0.0220	0.0036	0.0065	0.0008	0.0700	0.0032

Table 3.5 gave the accuracy performance evaluation of the sparse graphical MAR(p) models. Model p3004 and p3005 are lag order 1 models and had very similar performance in $RMSE_{\mathbf{A}}$ and $RMSE_{\mathbf{B}}$: Their $RMSE_{\mathbf{A}}$'s varied from 0.01 to 0.02 and their $RMSE_{\mathbf{B}}$'s varied from 0.02 and 0.07. The $RMSE_{\mathbf{\Theta}}$'s for Model p3004 were from 0.01 to 0.02 and for Model p3005 were from 0.06 to 0.2. Their $RMSE_{\mathbf{B}\otimes\mathbf{A}}$ were about 0.005 to 0.02. This indicated that their vector form models were quite accurate. Keeping the same length of the time series, the higher the lag order of the model, the larger the RMSE. The Model p3006 had $RMSE_{\mathbf{A}}$'s from 0.02 to 0.09, $RMSE_{\mathbf{B}}$'s from 0.02 to 0.07 and $RMSE_{\mathbf{\Theta}}$'s from 0.07 to 0.17. The Model p3007 had $RMSE_{\mathbf{A}}$'s from 0.04 to 0.12, $RMSE_{\mathbf{B}}$'s from 0.02 to 0.1 and $RMSE_{\mathbf{\Theta}}$'s from 0.07 to 0.18. Both had $RMSE_{\mathbf{B}\otimes\mathbf{A}}$ of about 0.006 to 0.025. This indicated that their vector form models were quite accurate. All these models gave satisfactory accuracy.

3.5 Application

We revisited the economic indicator time series example in Chen et al. (2021), with an extension of the time frame from 1991 to 2019. The data were quarterly observations

from four indicators, first differenced 3-month interbank interest rate, GDP growth (log difference), total manufacturing production growth (log difference), and CPI core inflation growth (log difference) from five countries, USA, Germany (DEU), France (FRA), United Kingdom (GBR) and Canada (CAN), obtained from Organisation for Economic Co-operation and Development (OECD) at https://data.oecd.org/. Before fitting any autoregressive models, the seasonality of CPI was adjusted by subtracting the sample quarterly means. All series are normalized and the combined variance of each indicator is 1. All these 20 time series formed a 4×5 matrix time series with time t at quarter with the length of $T = 115$.

70% of the data were used as training for fitting autoregressive models and sparse autoregressive models and the remaining 30% were used to get the out-of-sample forecast for model comparisons.

3.5.1 Fitting VAR and MAR models

We fitted the OECD training data with the SCT MAR(1), i.e. MAR(1) under a structured covariance tensor and general MAR(p) models, up to $p = 3$, and also a traditional general VAR(p) up to $p = 2$ model for comparison. Their BIC values are tabulated in Table 3.6. The SCT MAR(1) model had the smallest number of parameters and minimum BIC value (135.4).

When the in-sample and out-of-sample performance were examined, VAR(1) model has the minimum residual sum of squares, while MAR(2) model had the smallest out-of-sample prediction sum of error squares. Although the number of parameters was bigger than the SCT MAR(1) model, no overfitting phenomenon was observed. Therefore, the MAR(2) model was chosen as the final model.

Figure 3.4 shows the estimated left coefficient matrices, $\mathbf{A}_1$ and $\mathbf{A}_2$. The first columns of $\mathbf{A}_1$ and $\mathbf{A}_2$ show the influence on the current economic indicators from the last quarter's interest rate and the previous two quarter's interest rate respectively.

The coefficients of interest rate to GDP, production and CPI growth were negative in lags 1 and 2. This indicated that higher interest rates would slow down the GDP, production and CPI growth in the last quarter and the negative effect would be to a larger extent in the previous two quarters. The second columns in $\mathbf{A}_1$ and $\mathbf{A}_2$ show the influence on the current economic indicators from the last quarter's and previous two quarters' GDP growth. An increase in GDP growth would result in a small growth in interest rate in the last two quarters, but such GDP growth would first decrease the production growth slightly and would increase the production growth in the previous two quarters. In the case of CPI growth, it is the opposite of the influence of production growth. The third column of $\mathbf{A}_1$ is all negative. It indicated the impact of the production growth would make the interest rate, GDP and CPI growth slower in the past quarter and make the interest rate, GDP and CPI growth faster in the previous two quarters. The fourth column of $\mathbf{A}_1$ and $\mathbf{A}_2$ have small values. This indicated that CPI had a small positive impact on the interest rate and the production growth in the last two quarters.

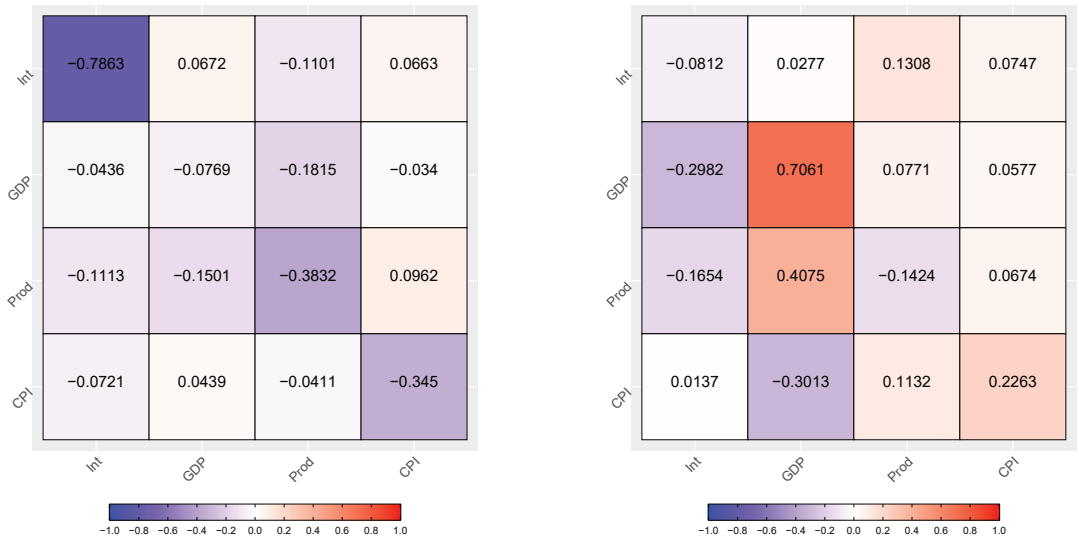
Figure 3.5 shows the estimated right coefficient matrices, $\mathbf{B}_1$ and $\mathbf{B}_2$. Their effect should be considered as $\mathbf{B}_1\mathbf{X}_t^T$ and $\mathbf{B}_2\mathbf{X}_{t-1}^T$. It could be interpreted similarly as matrix $\mathbf{A}_i, i = 1, 2$.

One way to select the sign of $\mathbf{A}_i, \mathbf{B}_i, i = 1, 2$ is to use the shock-first impulse and shock-second impulse functions so that the sign of the coefficients is consistent with the sign of the changes from the shock impulse function.

Figure 3.6 shows the partial correlation matrix of the MAR(2) Model. It can be observed that a lot of pairs have estimated partial correlation values smaller than 0.02. It is worth conducting the sparse MAR(p) model, which would give meaningful conditional dependent economic indicator pairs.

Table 3.6: BIC values, in-sample residual sum of squares (RSS), out-of-sample prediction error sum of squares (PSS) for models for the OECD data from 1991 to 2019

Model	Number of parameters	BIC values	RSS	PSS
1 SCT MAR(1)	66	135.4	103.05	45.5
2 General MAR(1)	251	446.8	113.15	38.45
3 General MAR(2)	292	470.2	126.44	34.55
4 General MAR(3)	333	542.1	114.23	38.87
5 VAR(1)	610	1265.2	70.4	48.99
6 VAR(2)	1010	1976.1	42.05	77.70

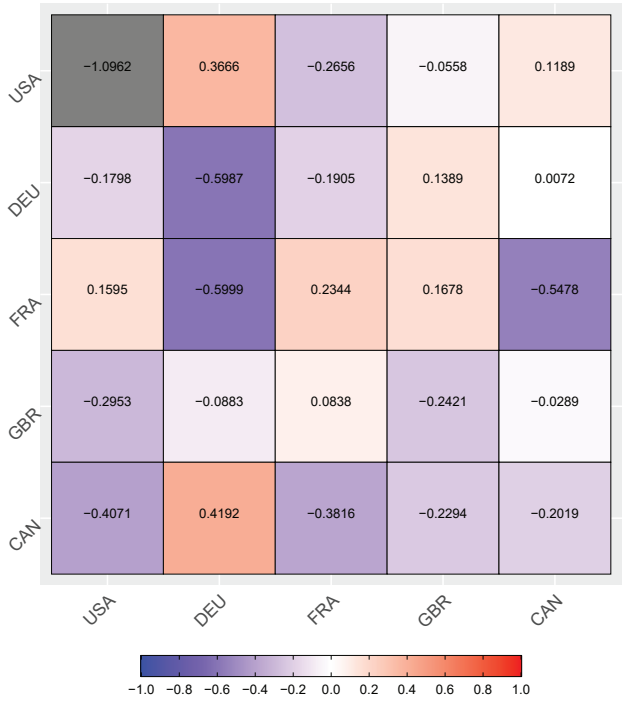


(a) A_1 (b) A_2

Figure 3.4: Row-wise interactions coefficients matrices of the MAR(2) Model for the OECD data

3.5.2 Fitting sparse graphical VAR and MAR models

We considered our sparse graphical MAR(p) model, up to $p = 3$ and compared it with the constrained graphical sparse vector autoregressive (CGsVAR) model in Yuen et al. (2018). The sparsity pattern of this model was pre-determined by the partial correlations of the data and the sparsity entries of the coefficient matrix were assumed the same as that of the precision matrix. The final selected constrained



(a) B_1

(b) B_2

Figure 3.5: Column-wise dependence coefficients matrices, B_1 and B_2 of the MAR(2) Model for the OECD data

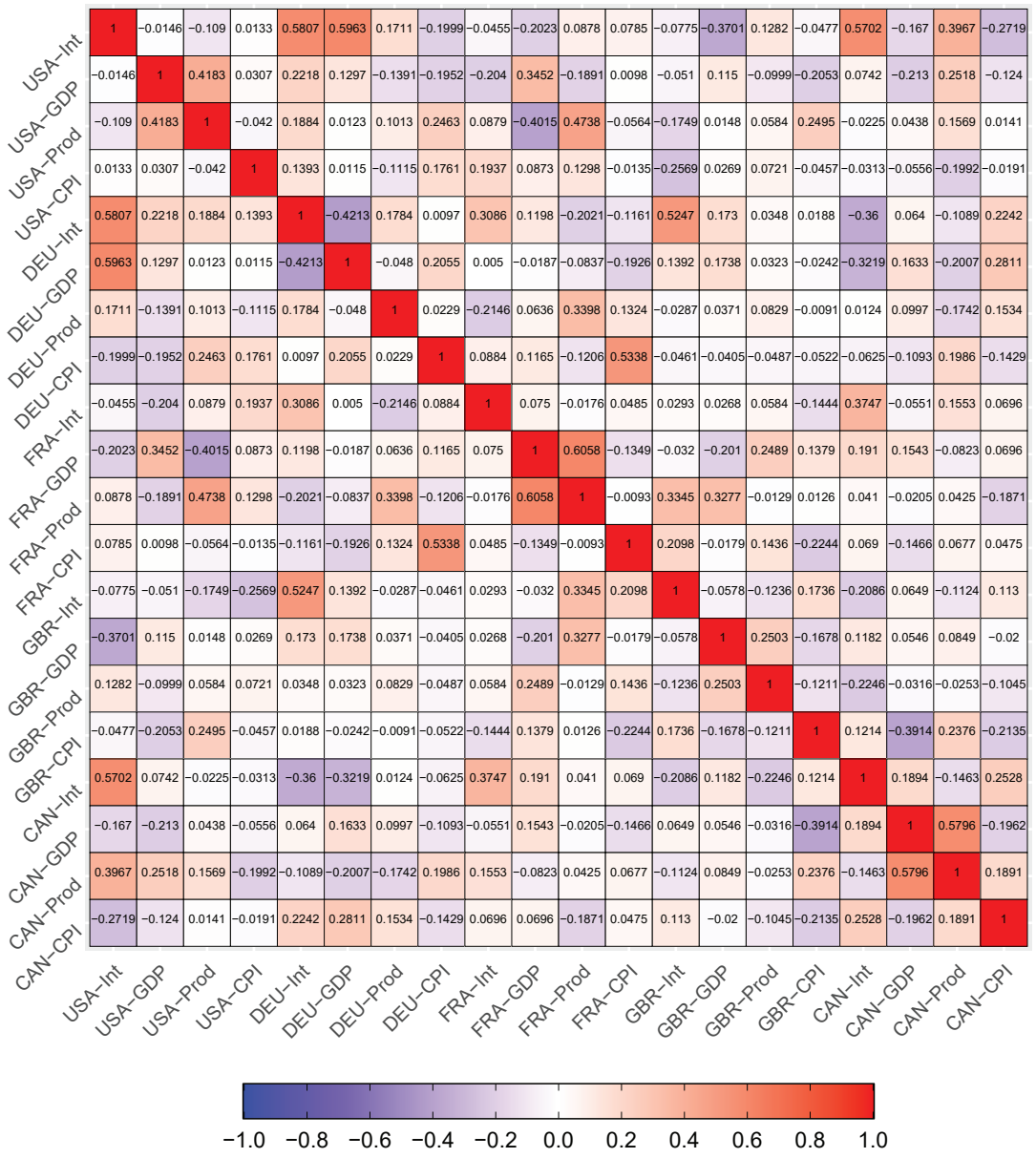


Figure 3.6: Partial correlation matrix of the MAR(2) Model for the OECD data

Table 3.7: BIC values, in-sample residual sum of squares (RSS), out-of-sample prediction sum of squares (PSS) for OECD sparse graphical VAR and MAR models

Model	Number of parameters	BIC	RSS	PSS
Constrained graphical sparse VAR(1) model	166	425.4	101.2	39.7
Sparse graphical MAR(1)	80	237.4	103.7	42.3
Sparse graphical MAR(2)	81	196.1	101.3	39.1
Sparse graphical MAR(3)*	86	160.6	94.4	37.9

sparse graphical model had lag order 1 and 166 parameters. It had a slightly larger in-sample residual sum of squares and out-of-sample prediction error sum of squares than the sparse graphical MAR(p) model ($p = 1, 2, 3$). The three MAR models had a similar number of parameters. Since the MAR(3) model had the smallest residual sum of squares and prediction error sum of squares, the sparse graphical MAR(3) model was selected as the final model.

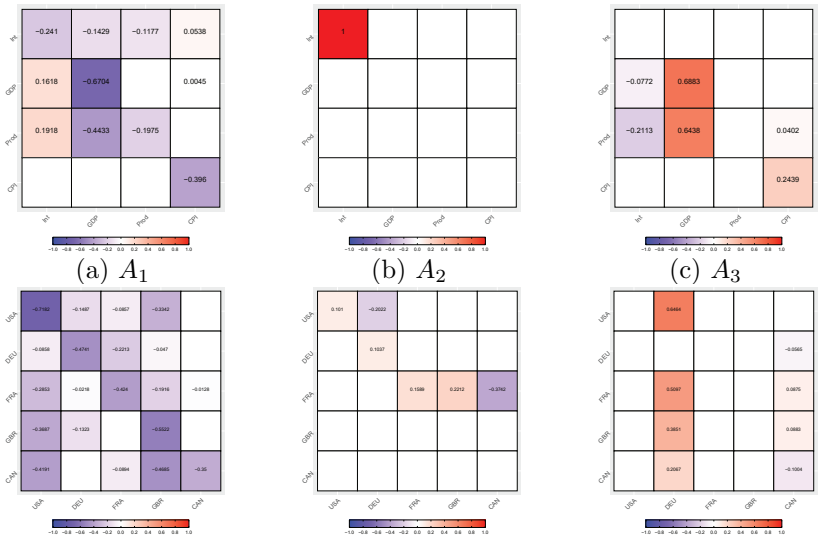
Figure 3.7 gives a table This MAR(3) model can be interpreted in a similar way as the MAR(2) model.

The sparse precision matrix is converted to a partial correlation matrix. Only those significant values are tabulated in Table 3.8. A conditional dependence graph is plotted in Figure 3.9. Interestingly, the US interest rate is connected with Canada's and Germany's interest rates. Apart from this relationship, we cannot see any connection between the USA and the other three European countries. And lots of edges are found between Germany and France and therefore, their closest relationship is closest. Canada has no edges connected with the three European Countries. This is intuitively correct.

3.6 Conclusion

We studied the matrix time series model, which is modified from a bilinear regression model into a time series with a matrix variate with a structured covariance tensor.

We extend the model to a higher lag order and introduce the general covariance structure so that any data with an imperfect independent relationship over two classifications could be modelled. In addition, a graphical model is merged with our matrix time series model to form a graphical matrix time series model. We adopt the optimal sparsity concept as in Chapter 2 for our sparse model and LASSO penalized estimation is used. The economic indicator example demonstrated that our MAR model had a higher lag and had a lower residual sum of squares value and a prediction error sum of squares value. The sparse model of the example exhibited an intuitively correct economic relationship between the five countries.

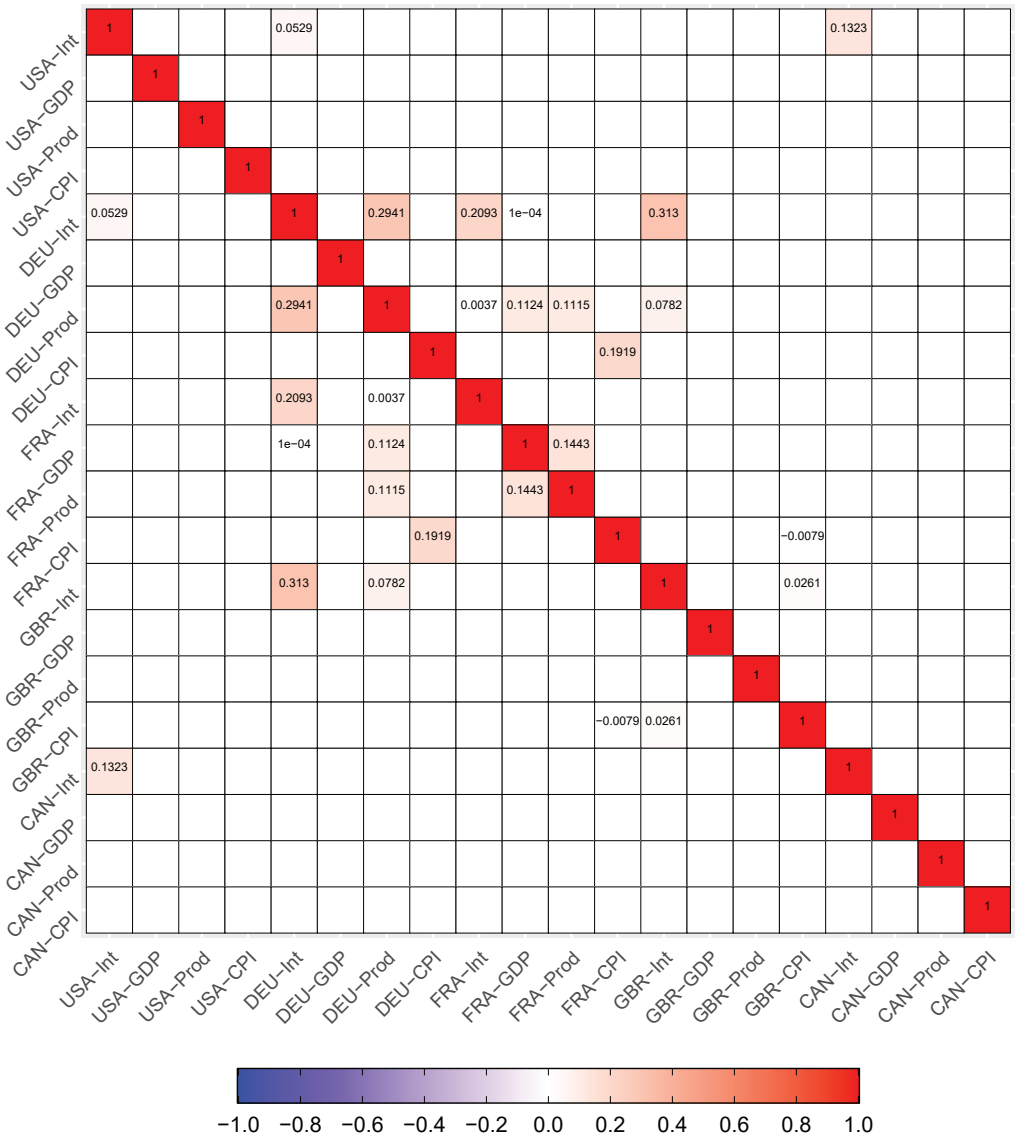


(a) A_1 (b) A_2 (c) A_3

(d) B_1 (e) B_2 (f) B_3

$\mathbf{A}_1$	Int	GDP	Prod	CPI	$\mathbf{B}_1$	USA	DEU	FRA	GBR	CAN
Int	-0.2410	-0.1429	-0.1177	0.0538	USA	-0.7182	-0.1487	-0.0857	-0.3342	0
GDP	0.1618	-0.6704	0	0.0045	DEU	-0.0858	-0.4741	-0.2213	-0.0470	0
Prod	0.1918	-0.4433	-0.1975	0	FRA	-0.2853	-0.0218	-0.4240	-0.1916	-0.0128
CPI	0	0	0	-0.3960	GBR	-0.3687	-0.1323	0	-0.5522	0
					CAN	-0.4191	0	-0.0894	-0.4685	-0.3500
$\mathbf{A}_2$	Int	GDP	Prod	CPI	$\mathbf{B}_2$	USA	DEU	FRA	GBR	CAN
Int	1.0000	0	0	0	USA	0.1010	-0.2022	0	0	0
GDP	0	0	0	0	DEU	0	0.1037	0	0	0
Prod	0	0	0	0	FRA	0	0	0.1589	0.2212	-0.3742
CPI	0	0	0	0	GBR	0	0	0	0	0
					CAN	0	0	0	0	0
$\mathbf{A}_3$	Int	GDP	Prod	CPI	$\mathbf{B}_3$	USA	DEU	FRA	GBR	CAN
Int	0	0	0	0	USA	0	0.6464	0	0	0
GDP	-0.0772	0.6883	0	0	DEU	0	0	0	0	-0.0565
Prod	-0.2113	0.6438	0	0.0402	FRA	0	0.5097	0	0	0.0875
CPI	0	0	0	0.2439	GBR	0	0.3851	0	0	0.0883
					CAN	0	0.2067	0	0	-0.1004

Figure 3.7: Row-wise interactions and column dependence coefficients matrices and the heatmap diagrams of the sparse graphical MAR(3) model for the OECD data



Significant partial correlation of sparse graphical MAR(3) model for the OECD data

Item1 ($I1$)	Item2 ($I2$)	$\rho_{I1,I2}$	Item3 ($I3$)	$\rho_{I1,I3}$	Item4 ($I4$)	$\rho_{I1,I4}$	Item5 ($I5$)	$\rho_{I1,I5}$
USA-Int	DEU-Int	-0.0529	CAN-Int	-0.1324				
DEU-Int	DEU-Prod	-0.2941	FRA-Int	-0.2093	FRA-GDP	-0.0001	GBR-Int	-0.313
DEU-Prod	FRA-Int	-0.0037	FRA-GDP	-0.1124	FRA-Prod	-0.1115	GBR-Int	-0.078
DEU-CPI	FRA-CPI	-0.1919						
FRA-GDP	FRA-Prod	-0.1444						
FRA-CPI	GBR-CPI	0.0079						
GBR-INT	GBR-CPI	-0.0261						

Figure 3.8: Partial correlation matrix of the sparse graphical MAR(3) model for the OECD data

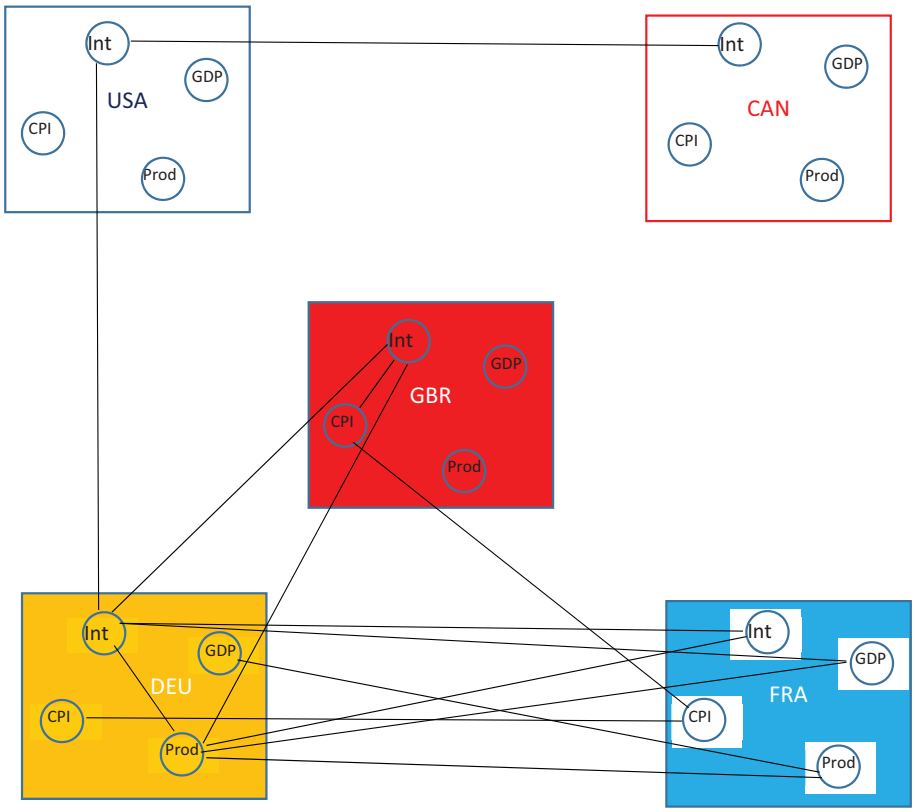


Figure 3.9: Conditional dependence graph for the OECD data

Chapter 4

Precision Matrix Estimation of High Dimensional Time Series

4.1 Introduction

Estimating high dimensional covariance and correlation matrices are important in portfolio selection, risk management, and asset pricing but their estimates may not be positive definite in some cases. For example, there are numerous stocks considered in a study, but the available stock time series data are short. The time elapses for several stock time series data may not be coincident. Then the number of variables for stock data might be greater than the number of observations and the estimation of the covariance matrix even becomes problematic, especially when the matrix is of high dimension. As a result, the estimated correlation and covariance matrices may not be positive definite and may have low ranks. Thus, the inverse of the covariance matrix estimates might not exist.

To handle the non-positive definite covariance problem in estimation, researchers have studied a number of methods.

Bai and Shi (2011) reviewed various methods of reducing the number of parameters for covariance matrix estimation. In the context of asset return, the shrinkage method gives a covariance estimator based on the linear combination of a single index

model and the sample covariance. In a factor model, the estimated factor vector and the estimated factor covariance matrix are used to construct a quadratic form and this quadratic form added with a diagonal matrix of the variance of the noise is used to estimate the sample covariance matrix. It can be observed that the above two estimation methods give the sample covariance matrix as a sum of a lower dimension full rank symmetric positive definite matrix and a diagonal positive matrix. Then the inverse of the covariance matrix estimate could be calculated.

Another approach is to find the nearest correlation matrix problem, which is one of the matrix nearness problems. A survey on this problem can be found in Higham (1999). Higham (1998) first studied a positive approximant for any arbitrary matrix and this matrix is the nearest symmetric positive semi-definite (psd) matrix. It is used to modify an indefinite Hessian matrix in the Newton method. Higham (1998) showed that the closest psd matrix was unique, based on the Frobenius norm, while the uniqueness of the positive approximant does not hold based on the shortest 2-norm distance. Higham (2002) examined a particular type of psd matrix, namely the correlation matrix. He used a modified alternating projection method to compute a symmetric psd matrix for a correlation matrix having zero or negative eigenvalues. An algorithm is developed and is linearly convergent.

Finding the nearest covariance matrix has been also a focus of nearness matrix problems. Boyd and Lin (2005) studied the least-squares covariance adjustment problem (LSCAP) and it was solved via its dual problem using standard optimization methods. The LSCAP aims to find the nearest symmetric psd matrix using the least squares sense in the Frobenius norm. In addition, linear equalities and inequalities can be imposed on the problem. Same as the correlation matrix approach, the resultant matrix is a projection on the positive semidefinite cone and is the optimal adjustment. The rank of the optimal adjustment was studied. Qi and Sun (2006) developed a quadratically convergent Newton method to find the nearest correlation

matrix and the algorithm is extended to find the nearest covariance matrix.

The positive definiteness of covariance matrices in many financial models is an obviously unavoidable constraint in estimation. This positive definiteness matrix constraint is also unavoidable for the precision matrix.

Traditional estimation methods need the vectorization of the covariance or the precision matrix, but this method destructs its positive definite property. Even if the matrix optimization is used, it is difficult to express the positive definite property of these matrices as linear equality or inequality constraints in the optimization problems. Computation of $\log(\det(\mathbf{\Sigma}))$ gives a numerical error when the k-th step non-positive definite covariance iterate, $\mathbf{\Sigma}^{(k)}$, for some $k \in \mathbb{Z}$, is generated in the algorithm. The algorithm stops and the estimation is not successful.

Instead of imposing constraints on the positive definiteness of covariance matrices, we develop convergent methods for precision matrix estimation in constrained vector time series model and matrix time series model estimation. Both methods are developed based on existing optimization methods. They skip the non-positive definite covariance/precision matrix iterates and replace them with the closest symmetric psd matrices, based on Qi and Sun (2006)'s nearest correlation matrix method. We proved that these algorithms are descent methods.

This chapter is organized as follows. Section 4.2 outlines precision matrix estimation as an optimization problem. Section 4.3 discusses the proposed algorithms to keep the precision matrix iterates in positive definite cones for vector and matrix approaches. Section 4.4 illustrates the convergence of our three proposed algorithms based on examples which were failed to be estimated by some standard optimization methods. Section 4.5 gives a conclusion.

4.2 The General Problem

Let K_1 and K_2 be positive integers and $\mathbf{\Theta}$ be a $K_2 \times K_2$ covariance/precision matrix. Let $K = K_1 + K_2$ be the dimension of the problem in the vector/ square matrix form and $w \in \mathbb{R}^{K_1+K_2}$, $l(w)$ be the log-likelihood function of a probabilistic model. Define $\mathcal{S}_{++} = \{u = \text{vech}(\mathbf{\Theta}) \mid \mathbf{\Theta} \succ 0 \text{ and } \mathbf{\Theta} \in \mathbb{R}^{K_2 \times K_2}\}$ for some K_2. Here, the statement $\mathbf{\Theta} \succ 0$ represents $\mathbf{\Theta}$ being positive definite.

We assume that $l(w)$ is bounded above and continuously differentiable with Lipschitz continuous gradient, that is, there exists $\beta > 0$ so that

$$\|\nabla l(w_1, w_2) - \nabla l(u_1, u_2)\| \leq \beta \|(w_1 - u_1, w_2 - u_2)\| \quad \text{for all} \quad w_1, u_1 \in \mathbb{R}^{K_1}, w_2, u_2 \in \mathcal{S}_{++} \tag{4.1}$$

The boundedness and continuous differentiability assumptions of $l(.)$ can be established similarly in the matrix form.

We consider the following general problems:

(a) in the vector form,

$$\begin{aligned} \min_{y_1, y_2} f(y_1, y_2) &:= -l(y_1, y_2) \\ \text{subject to } y_2 &= \text{vech}(\mathbf{\Theta}), \\ \mathbf{\Theta} &\succ 0, \\ g_i(y_1, y_2) &\leq 0 \text{ for } i = 1, \ldots, v, \text{ and} \\ h_j(y_1, y_2) &= 0 \text{ for } j = 1, \ldots, w. \end{aligned} \tag{4.2}$$

where l is the log-likelihood function, y_1 and y_2 are model parameter vectors, $\mathbf{\Theta}$ is a positive definite matrix, g_i and h_j are inequality and equality constraints and $v, w \in \mathbb{Z}$. The g_i's and h_j's constraints are not required in some estimation problems.

(b) in the matrix form,

$$\begin{aligned} \min_{\mathbf{x}_1,\ldots,\mathbf{x}_q,\boldsymbol{\Theta}} f(\mathbf{x}_1,\ldots,\mathbf{x}_q,\boldsymbol{\Theta}) &:= -l(\mathbf{x}_1,\ldots,\mathbf{x}_q,\boldsymbol{\Theta}) \\ \text{subject to } \boldsymbol{\Theta} &\succ 0 \\ g_i(\mathbf{x}_1,\ldots,\mathbf{x}_q,\boldsymbol{\Theta}) &\leq 0 \text{ for } i=1,\ldots,v, \text{ and} \\ h_j(\mathbf{x}_1,\ldots,\mathbf{x}_q) &= 0 \text{ for } j=1,\ldots,w. \end{aligned} \tag{4.3}$$

where l is the log-likelihood function, $\mathbf{x}_1,\ldots,\mathbf{x}_q$ and $\boldsymbol{\Theta}$ are model parameter matrices, $\boldsymbol{\Theta}$ is a positive definite matrix, and g_i and h_j are inequality and equality constraints and $v, w \in \mathbb{Z}$. The g_i's and h_j's constraints are not required in some estimation problems.

In optimization problems, the superscript is always added to a symbol to represent an iterate of an algorithm. i.e. $y^{(k)}$ is the k-th iterate of y in an algorithm.

4.3 The Proposed Algorithms

The positive definiteness of $\boldsymbol{\Theta}$ is not easily handled as equality or inequality constraints in most of the existing optimization algorithms. To avoid numerical errors in the computation of $\log(\det(\boldsymbol{\Theta}))$, we propose to replace the non-positive definite $\boldsymbol{\Theta}^{(k)}$ iterates with the closest symmetric positive definite matrix and amend some existing standard optimization algorithms. The matrix replacement incorporates a calibration procedure of a covariance matrix, based on the Frobenius norm, in Qi and Sun (2006) and this matrix calibration procedure can be written as follows.

$$\begin{aligned} \min_{\mathbf{Z}} \quad & \frac{1}{2}\|\mathbf{Z}-\boldsymbol{\Theta}^{(k)}\|^2 \\ s.t. \quad & \mathbf{Z} \succeq \tau\mathbf{I}, \\ & \langle \mathbf{I}, \mathbf{Z}\rangle = \operatorname{tr}(\boldsymbol{\Theta}^{(k)}), \\ & z_{ii} = \theta_{ii}^{(k)}, i=1,\ldots,K_2, \end{aligned} \tag{4.4}$$

where $\mathbf{Z} = (z_{ij})$, $\mathbf{\Theta}^{(k)} = (\theta_{ij}^{(k)})$, $\tau > 0$.

This calibration step is applied in the estimation of two time series models. Algorithm 2, COVLS, is designed for vector time series model estimation while Algorithms LSNCM and LSNCM_IV aim for matrix time series models estimation.

4.3.1 Vector optimization algorithm COVLS

We study the constrained vector time series model problem (4.2). A typically constrained optimization without the positive definiteness constraint of a precision matrix can be formulated as follows.

$$\begin{aligned} y &= \min_{y} f(y) \\ \text{subject to } g_i(y) &\leq 0 \text{ for } i = 1, \ldots, v, \text{ and} \\ h_j(y) &= 0 \text{ for } j = 1, \ldots, w. \end{aligned}$$

This is equivalent to minimizing an approximate problem:

$$f_\mu(y, s) = f(y) - \mu \sum_{i=1}^{r} \log(s_i) \tag{4.5}$$

where $s = (s_1, \ldots, s_r)$ are slack variables, $r = \dim(y)$ and $\mu > 0$. Define the auxiliary Lagrangian function as

$$L(y, \lambda) = f(y) + \sum_{i=1}^{\nu} \lambda_{g_i} g_i(y) + \sum_{j=1}^{\omega} \lambda_{h_j} h_j(y),$$

where $\lambda = (\lambda_{g_1}, \ldots, \lambda_{g_\nu}, \lambda_{h_1}, \ldots, \lambda_{h_w})$, $g = (g_1, \ldots, g_\nu)$ and $h = (h_1, \ldots, h_w)$. Then the Karush-Kuhn-Tucker (KKT) conditions are $\nabla L(y, \lambda) = 0$, $\lambda_{g_i} g_i(y) = 0$ for all i and $g(y) \leq 0$, $h(y) = 0$ and $\lambda_{g_i} \geq 0$.

Let $y = (y_1, y_2)$ and $\mathbf{H}(y_1, y_2)$ be the Hessian of the Lagrangian of function of

$f_\mu(y_1, y_2, s)$. Then

$$\mathbf{H}(y_1, y_2) = \nabla^2 f(y_1, y_2) + \sum_{i=1}^{\nu} \lambda_{g_i} \nabla^2 g_i(y_1, y_2) + \sum_{j=1}^{\omega} \lambda_{h_j} \nabla^2 h_j(y_1, y_2).$$

Let $\mathbf{J}_g(y_1, y_2)$ and $\mathbf{J}_h(y_1, y_2)$ be the Jacobian of constraint functions g and h respectively.

Suppose the iterate $(y_1^{(k-1)}, y_2^{(k-1)})$ is close to the solutions. The solution can be obtained via the approximate problem using the Newton method. i.e. $y^{(k)} = y^{(k-1)} + \Delta y^{(k-1)}$ and $\Delta y^{(k-1)}$ is a solution of Δy of

$$\begin{pmatrix} \mathbf{H} & 0 & \mathbf{J}_h^t & \mathbf{J}_g^T \\ 0 & \mathbf{\Lambda}_g & 0 & \mathbf{S} \\ \mathbf{J}_h & 0 & 0 & 0 \\ \mathbf{J}_g & \mathbf{I} & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta y \\ \Delta s \\ \Delta \lambda_h \\ \Delta \lambda_g \end{pmatrix} = \begin{pmatrix} \nabla f + \mathbf{J}_h^T \lambda_h + \mathbf{J}_g^T \lambda_g \\ \mathbf{S}\lambda_g - \mu e \\ h \\ g + s \end{pmatrix}, \tag{4.6}$$

where $\mathbf{H} = \mathbf{H}(y_1^{(k-1)}, y_2^{(k-1)})$, $\mathbf{J}_g = \mathbf{J}_g(y_1^{(k-1)}, y_2^{(k-1)})$ and $\mathbf{J}_h = \mathbf{J}_h(y_1^{(k-1)}, y_2^{(k-1)})$, $\mathbf{S} = \text{diag}(s)$, $\mathbf{\Lambda}_g = \text{diag}(\lambda_g)$, λ_g is the Lagrange multiplier vector associated with constraint g, λ_h is the Lagrange multiplier vector associated with constraint h and $e = (1, \ldots, 1)^T$ with length ν.

This algorithm converges to infeasible points or does not converge for some examples. Possible reasons are that the initial point is not close to the solutions or $\mathbf{\Theta}$ obtained from y_2 is not positive definite and causes a numerical error in the $\log(\det(\mathbf{\Theta}))$ term in the function $f(y)$. Therefore, there is a need to handle the positive definiteness constraint of the precision matrix. This Newton step on Lagrange optimization can be written as Algorithm 1, CMY.

We make use of the ideas of the line search for the descent algorithm, the nearest correlation matrix and the above Newton step on Lagrange optimization for a new algorithm on vector time series models estimation and propose Algorithm 2, COVLS.

Algorithm 1 CMY - Constrained minimization of y

Require: Initial values of $k > 0$ and $y^{(k-1)} = (y_1^{(k-1)}, y_2^{(k-1)})$, vech($\mathbf{\Theta}^{(k-1)}$)=$y_2^{(k-1)}$.

1: **repeat**

2: Compute

$$\begin{aligned} \mathbf{H} &= \mathbf{H}(y_1^{(k-1)}, y_2^{(k-1)}), \\ \mathbf{J}_g &= \mathbf{J}_g(y_1^{(k-1)}, y_2^{(k-1)}) \text{ and} \\ \mathbf{J}_h &= \mathbf{J}_h(y_1^{(k-1)}, y_2^{(k-1)}). \end{aligned}$$

3: Solve Δy from Equation 4.6.

4: Compute $y^{(k)} = y^{(k-1)} + \Delta y$ and $\max |\frac{\partial}{\partial y_i} f(y^{(k)})|$ (for $i = 1, \ldots, K$).

5: $k = k + 1$

6: **until** one of the following criteria is fulfilled:

- $y^{(k)} = (y_1^{(k)}, y_2^{(k)})$ converges with a positive definite precision matrix $\mathbf{\Theta}^{(k)}$ (vech($\mathbf{\Theta}^{(k)}$)=$y_2^{(k)}$) or
- a non-positive definite precision matrix $\mathbf{\Theta}^{(k)}$; or
- $\begin{pmatrix} \mathbf{H} & 0 & \mathbf{J}_h^t & \mathbf{J}_g^T \\ 0 & \mathbf{\Lambda}_g & 0 & \mathbf{S} \\ \mathbf{J}_h & 0 & 0 & 0 \\ \mathbf{J}_g & \mathbf{I} & 0 & 0 \end{pmatrix}$ does not have an inverse.

7: **if** $y^{(k)}$ converges with a positive definite precision matrix $\mathbf{\Theta}^{(k)}$ **then**

8: Denote the solution by $y^* = (y_1^*, y_2^*)$.

9: **end if**

Three main ideas are applied to establish the convergence of the proposed Algorithm 2, COVLS. The first idea is to obtain a descent f via the line search method in Step 7. Secondly, the line search iterate is used as an initial value for the above constrained optimization algorithm. Thirdly, whenever a non-positive definite precision matrix is obtained in the line search, it is replaced by the closest symmetric positive definite matrix generated from Steps 9 to 17. The positive τ value entails the positive definiteness of the precision matrix.

Some steps to speed up the new algorithm are considered. A non-positive definite matrix in the line search will only be replaced by a positive definite matrix when $f(y_1^{(k')}, y_{2,\tau}^{(k',j)})$ is less than $2f(y_1^{(0)}, y_2^{(0)})$ in Step 13. This avoids obtaining a $\mathbf{\Theta}$ with

Algorithm 2 COVLS

Set the initial value $y^{(0)} = (y_1^{(0)}, y_2^{(0)})$ and $k = 1$.
Conduct **Algorithm 1**, **CMY** for obtaining a list of y's and their first-order optimality, $\max|\frac{\partial}{\partial y_i} f(y^{(k')})|$, $i = 1, \ldots, K$.
From the iterates generated from the previous step, choose the iterate, $y^{(k')}$ which has the smallest first-order optimality, $\max|\frac{\partial}{\partial y_i} f(y^{(k')})|$ (for $i = 1, \ldots, K$).
Obtain the norm of the step $s^{(k')} = \|y^{(k')} - y^{(k'-1)}\|$.
Obtain $y_2^{(k')}$ from $y^{(k')}$, set $j = 1$ and $t^{(k',j)} = cs^{(k')}$. In practice, $c = 1/10$.
repeat
    Compute $y^{(k',j)} = y^{(k')} - t^{(k',j)} \nabla f(y_1^{(k')}, y_2^{(k')})$ and obtain $\mathbf{\Theta}^{(k',j)}$ from $y_2^{(k',j)}$.
    if $\mathbf{\Theta}^{(k',j)}$ is not positive definite **then**
        set $\tau = 0.01$.
        repeat
            Compute $\mathbf{\Theta}_\tau^{(k')} = \min_{\mathbf{Q}} \frac{1}{2}\|\mathbf{Q} - \mathbf{\Theta}^{(k')}\|$, such that

$$\begin{aligned} \mathbf{Q} &\succeq \tau\mathbf{I}, \\ \langle \mathbf{I}, \mathbf{Q} \rangle &= tr(\mathbf{\Theta}^{(k',j)}) \text{ and} \\ (\mathbf{Q})_{ii} &= (\mathbf{\Theta}^{(k',j)})_{ii}. \end{aligned}$$

            Compute $y_{2,\tau}^{(k',j)} = vech(\mathbf{\Theta}_\tau^{(k')})$.
            if $f(y_1^{(k')}, y_{2,\tau}^{(k',j)}) < 2f(y_1^{(0)}, y_2^{(0)})$ **then**
                Conduct **Algorithm 1**, **CMY**, with
                - $k = 1$
                - $(y_1^{(k')}, y_{2,\tau}^{(k',j)})$ as an initial value.
            end if
            $\tau = \tau + 0.01$.
        until (y_1, y_2) iterate sequence converges or $\tau > \min(0.1, \text{diag}(\mathbf{\Theta}^{(k',j)}))$.
    else
        Conduct **Algorithm 1**, **CMY** for getting (y_1, y_2), with
        - $k = 1$.
        - $(y_1^{(k')}, y_2^{(k',j)})$ as an initial value,
    end if
    if (y_1, y_2) iterate sequence converges with first-order optimality ≈ 0 **then**
        $y^* = (y_1^*, y_2^*)$ is the solution.
        Stop
    else
        $t^{(k',j)} = t^{(k',j)}/2$ and $j = j + 1$.
    end if
until $t^{(k',j)} < 1e - 10$.

an unnecessarily extremely large f value in the line search because the iterate $\boldsymbol{\Theta}$ will be replaced by another iterate with a smaller f value. Another action is to start a line search from an iterate with the smallest first-order derivative of f selected from Algorithm 1, CMY of its unconstrained VAR(p) model so that an initial value is closer to the solution for the above Newton step is obtained. In theory, the line search can be started directly from the maximum likelihood estimate.

Algorithm 1, CMY, and Algorithm 2, COVLS, are coded with the *fmincon* procedure in MATLAB in the simulation study. Similar equivalent functions, gekko() function in python and the nloptr function in R, can also be used.

4.3.2 Matrix optimization algorithm LSNCM

In this section, we assume that the coordinate gradient descent algorithm can be used to estimate a matrix time series model in Problem 4.3. The coordinate gradient descent algorithm is given in Algorithm 3.

Algorithm 3 Coordinate gradient descent algorithm for matrix time series model

1: Let $\mathbf{x}_0 = (\mathbf{x}_1^{(0)}, \ldots, \mathbf{x}_q^{(0)}, \boldsymbol{\Theta}^{(0)})$ be the initial value. Set $i = 1$.
2: **repeat**
3: **for** $j \leftarrow 1, q$ **do**
4: Compute $\mathbf{x}_j^{(i)} = \min_{\mathbf{x}_j} f(\mathbf{x}_1, \mathbf{x}_2^{(i-1)}, \ldots, \mathbf{x}_q^{(i-1)}, \boldsymbol{\Theta}^{(i-1)})$.
5: **end for**
6: Compute $\boldsymbol{\Theta}^{(i)} = \min_{\boldsymbol{\Theta}} f(\mathbf{x}_1^{(i-1)}, \ldots, \mathbf{x}_q^{(i-1)}, \boldsymbol{\Theta})$
7: $i = i + 1$.
8: **until** some convergence criterion is satisfied.

The $\boldsymbol{\Theta}$ estimation step may generate a non-positive definite matrix in a line search step. To guarantee the positive definiteness of all precision matrix iterates, we approximate any non-positive definite matrix, $\boldsymbol{\Theta}$, with its closest positive definite matrix. As the new algorithm makes use of the two concepts: the line search and the nearest correlation matrix, all the first letter of the name is used to form the name of the algorithm, i.e. LSNCM.

Algorithm 4 LSNCM

1: Let $\mathbf{\Theta}^{(i)}$ be the i-th MAR(p) estimation iterate. Set step size, $s = 1$. Set $j = 1$.
2: **repeat**
3: Compute
$$\mathbf{\Theta}^{(i,j)} = \mathbf{\Theta}^{(i-1)} - s \cdot \nabla f(\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_q^{(i)}, \mathbf{\Theta}^{(i-1)}).$$
4: Compute the eigenvalues $\lambda_1, \ldots, \lambda_K$ of the j-th iterate of $\mathbf{\Theta}^{(i)}$, $\mathbf{\Theta}^{(i,j)}$.
5: **if** $\min\{\lambda_j : j = 1, \ldots, K\} <= 0$ **then**
6: Set $\tau = 0.1$.
7: **repeat**
8: $\mathbf{\Theta}_\tau^{(i,j)} = \min \frac{1}{2}\|\mathbf{Q} - \mathbf{\Theta}^{(i)}\|$ such that
$$\begin{aligned} \mathbf{Q} &\succeq \tau\mathbf{I}, \\ \langle \mathbf{I}, \mathbf{Q} \rangle &= \operatorname{tr}(\mathbf{\Theta}^{(i)}) \text{ and} \\ \operatorname{diag}(\mathbf{Q}) &= \operatorname{diag}(\mathbf{\Theta}^{(i)}). \end{aligned}$$
9: $\tau = \tau + 0.1$.
10: Compute $f_{new} = f(\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_q^{(i)}, \mathbf{\Theta}_\tau^{(i,j)})$.
11: Set $j = j + 1$.
12: **until** one of the following criterion is fulfilled:
- $\tau \geq \min(1, \theta_{ii})$, where $\operatorname{diag}(\theta_{11}, \ldots, \theta_{mn,mn})$; or
- $f_{new} >= f(\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_q^{(i)}, \mathbf{\Theta}^{(i-1)})$.

13: **end if**
14: $s = s/2$.
15: **until** some convergence criterion is satisfied.

The main idea of Algorithm 4, LSNCM, is to replace a precision matrix iterate with its closest positive definite symmetric matrix, when it is not positive definite in the line search. To ensure the f being descent in the algorithm, we impose the condition $f_{new} >= f(\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_q^{(i)}, \mathbf{\Theta}^{(i-1)})$ in Step 12. Again, the positive τ value is to ensure that the precision matrix is estimated to be a positive definite matrix instead of a positive semi-definite matrix.

4.3.3 Matrix optimization algorithm for initial value

Initial values are important in a nonlinear optimization because they are the starting point of the algorithm and most of the time they are taken as maximum likelihood

estimates. When the sample size is small, the maximum likelihood estimate may not be close to the true value. In this case, the initial value of the precision matrix may not be stable and may have a zero determinant.

The technique of closest positive definite matrix replacement can also be applied in the initial value of $\mathbf{\Theta}_0$, when a numerical error occurs in the $\log(\det(\mathbf{\Theta}_0))$ term in the log-likelihood function calculation. For simplicity, we apply the following IV algorithm when the $\det(\mathbf{\Theta}_0)$ term is less than 1×10^{-8}:

Algorithm 5 IV Algorithm for fixing non-positive definite initial precision matrix

1: $\tau = 0.01$
2: **repeat**
3: $\quad \mathbf{\Theta}_{0,\tau} = \min \frac{1}{2}\|\mathbf{Q} - \widehat{\mathbf{\Theta}}_0\|$ such that

$$\begin{aligned} \mathbf{Q} &\succeq \tau\mathbf{I}, \\ \langle \mathbf{I}, \mathbf{Q} \rangle &= \operatorname{tr}(\widehat{\mathbf{\Theta}}_0) \text{ and} \\ \operatorname{diag}(\mathbf{Q}) &= \operatorname{diag}(\widehat{\mathbf{\Theta}}_0). \end{aligned}$$

4: $\quad \tau = \tau + 0.01$
5: **until** $\det(\mathbf{\Theta}_{0,\tau}) > 1 \times 10^{-8}$ or $\tau > \min\{\theta_{ii} : \operatorname{diag}(\widehat{\mathbf{\Theta}}_0) = (\theta_{11}, \ldots, \theta_{mn,mn})\}$.

When the **Algorithm 5, IV** is stacked up with **Algorithm 4, LSNCM**, we call it as **Algorithm LSNCM_IV** for simplicity.

4.3.4 Convergence of the algorithms

In this section, we discuss the convergence of the two algorithms. The algorithms require descent line search methods. A line search method determines a decent direction and a step size for the next iterate so that the next iterate moves along the descent direction of the objective function f.

Let the step size is $1/s^{(k)}$. We would like to prove the algorithm satisfying the following criterion:

$$f(y_1^{(k)}, y_2^{(k)}) \leq f(y_1^{(k-1)}, y_2^{(k-1)}) - \frac{\eta}{2} s^{(k)} \|(y_1^{(k)}, y_2^{(k)}) - (y_1^{(k-1)}, y_2^{(k-1)})\|^2 \qquad (4.7)$$

for $\eta \in (0,1)$ and $(y_1^{(k)}, y_2^{(k)}), (y_1^{(k-1)}, y_2^{(k-1)})$ are the k-th and $(k-1)$-th iterate and are in the domain of f.

Assume $y_2^{(k-1)} = \text{vech}(\mathbf{\Theta}^{(k-1)})$ and $\mathbf{\Theta}^{(k-1)} \succ 0$. We conduct the line search, i.e.

$$(y_1^{(k)}, y_2^{(k)}) = (y_1^{(k-1)}, y_2^{(k-1)}) - \frac{1}{s^{(k)}} \nabla f(y_1^{(k-1)}, y_2^{(k-1)})$$

We actually search for a minimizer of

$$\begin{aligned} (y_1^{(k)}, y_2^{(k)}) &= \arg\min_{(y_1,y_2)} \frac{1}{2} \|(y_1, y_2) - (u_1^{(k)}, u_2^{(k)})\| \\ \text{where } (u_1^{(k)}, u_2^{(k)}) &= (y_1^{(k-1)}, y_2^{(k-1)}) - \frac{1}{s^{(k)}} \nabla f(y_1^{(k-1)}, y_2^{(k-1)}) \end{aligned}$$

This is equivalent to

$$\begin{aligned} (y_1^{(k)}, y_2^{(k)}) = \arg\min_{(y_1^{(k-1)}, y_2^{(k-1)})} & f(y_1^{(k-1)}, y_2^{(k-1)}) \\ & + \left\langle \nabla f(y_1^{(k-1)}, y_2^{(k-1)}), (y_1 - y_1^{(k-1)}, y_2 - y_2^{(k-1)}) \right\rangle \\ & + \frac{s^{(k-1)}}{2} \left(\|y_1 - y_1^{(k-1)}\|^2 + (\|y_2 - y_2^{(k-1)}\|^2 \right) \end{aligned} \tag{4.8}$$

Since $(y_1^{(k)}, y_2^{(k)})$ is a minimizer, we have

$$\begin{aligned} & \left\langle \nabla f(y_1^{(k-1)}, y_2^{(k-1)}), (y_1^{(k)} - y_1^{(k-1)}, y_2^{(k)} - y_2^{(k-1)}) \right\rangle \\ & \qquad + \frac{s^{(k-1)}}{2} \left(\|y_1^{(k)} - y_1^{(k-1)}\|^2 + \|y_2^{(k)} - y_2^{(k-1)}\|^2 \right) \leq 0 \end{aligned} \tag{4.9}$$

From Qi and Sun (2006), we have for each $y_2^{(k)}$, there exists a $\tau \geq 0$ and $y_{2,\tau}^{(k)} \succ 0$ such that $\|y_{2,\tau}^{(k)} - y_2^{(k)}\| = \gamma_{k,\tau}$, for $\gamma_{k,\tau} \geq 0$. Using the inequality (4.1), we have

$$\begin{aligned} f(y_1^{(k)}, y_{2,\tau}^{(k)}) \leq & f(y_1^{(k-1)}, y_2^{(k-1)}) \\ & + \left\langle \nabla f(y_1^{(k-1)}, y_2^{(k-1)}), (y_1^{(k)}, y_{2,\tau}^{(k)}) - (y_1^{(k-1)}, y_2^{(k-1)}) \right\rangle \\ & + \frac{\beta}{2} \|(y_1^{(k)}, y_{2,\tau}^{(k)}) - (y_1^{(k-1)}, y_2^{(k-1)})\|^2 \end{aligned} \tag{4.10}$$

Combining (4.9) and (4.10), we have

$$\begin{aligned}
f(y_1^{(k)}, y_{2,\tau}^{(k)}) &\leq f(y_1^{(k-1)}, y_2^{(k-1)}) \\
&\quad + \left\langle \nabla_{y_2} f(y_1^{(k-1)}, y_2^{(k-1)}), y_{2,\tau}^{(k)} - y_2^{(k)}) \right\rangle \\
&\quad + \left\langle \nabla f(y_1^{(k-1)}, y_2^{(k-1)}), (y_1^{(k)}, y_2^{(k)}) - (y_1^{(k-1)}, y_2^{(k-1)}) \right\rangle \\
&\quad + \frac{\beta}{2} \|(y_1^{(k)}, y_{2,\tau}^{(k)}) - (y_1^{(k-1)}, y_2^{(k-1)})\|^2 \\
&\leq f(y_1^{(k-1)}, y_2^{(k-1)}) \\
&\quad + \left\langle \nabla_{y_2} f(y_1^{(k-1)}, y_2^{(k-1)}), y_{2,\tau}^{(k)} - y_2^{(k)}) \right\rangle \\
&\quad - \frac{s^{(k-1)}}{2} \left(\|y_1^{(k)} - y_1^{(k-1)}\|^2 + \|y_2^{(k)} - y_2^{(k-1)}\|^2 \right) \\
&\quad + \frac{\beta}{2} \|(y_1^{(k)}, y_{2,\tau}^{(k)}) - (y_1^{(k-1)}, y_2^{(k-1)})\|^2 \\
&\leq f(y_1^{(k-1)}, y_2^{(k-1)}) \\
&\quad + \left\langle \nabla_{y_2} f(y_1^{(k-1)}, y_2^{(k-1)}), y_{2,\tau}^{(k)} - y_2^{(k)}) \right\rangle \\
&\quad - \frac{s^{(k-1)}}{2} \left(\|y_1^{(k)} - y_1^{(k-1)}\|^2 + \|y_2^{(k)} - y_2^{(k-1)}\|^2 \right) \\
&\quad + \frac{\beta}{2} \|(y_1^{(k)}, y_{2,\tau}^{(k)}) - (y_1^{(k-1)}, y_2^{(k-1)})\|^2 \qquad (4.11)
\end{aligned}$$

By the inequality (4.1) and boundedness of f, there exists a $L_f > 0$ such that

$$\begin{aligned}
\left\langle \nabla_{y_2} f(y_1^{(k-1)}, y_2^{(k-1)}), y_{2,\tau}^{(k)} - y_2^{(k)} \right\rangle &\leq \|\nabla_{y_2} f(y_1^{(k-1)}, y_2^{(k-1)})\| \|y_{2,\tau}^{(k)} - y_2^{(k)}\| \\
&\leq L_f \|y_{2,\tau}^{(k)} - y_2^{(k)}\|^2 \\
&= L_f \cdot \gamma_{k,\tau}^2
\end{aligned}$$

Assume $\|y_2^{(k)} - y_2^{(k-1)}\| = \|y_{2,\tau}^{(k)} - y_2^{(k-1)}\| + \delta_{k,\tau}$ for some $\delta_{k,\tau}$. The inequality (4.11)

becomes

$$\begin{aligned} f(y_1^{(k)}, y_{2,\tau}^{(k)}) \;\leq\; & f(y_1^{(k-1)}, y_2^{(k-1)}) \\ & + L_f \cdot \gamma_{k,\tau}^2 - \left(\frac{s^{(k-1)}}{2} - \frac{\beta}{2}\right)\left(\|(y_1^{(k)} - y_1^{(k-1)})\|^2 + \|(y_{2,\tau}^{(k)} - y_2^{(k-1)})\|^2\right) \\ & - \frac{s^{(k-1)}}{2}\delta_{k,\tau} \end{aligned} \tag{4.12}$$

In order to make the line search criterion well-defined, the following condition

$$L_f \cdot \gamma_{k,\tau}^2 - (\frac{s^{(k-1)}}{2} - \frac{\beta}{2})(\|(y_1^{(k)} - y_1^{(k-1)})\|^2 + \|(y_{2,\tau}^{(k)} - y_2^{(k-1)})\|^2) - \frac{s^{(k-1)}}{2}\delta_{k,\tau} \leq 0 \tag{4.13}$$

is fulfilled. Under this condition, $f(y_1^{(k)}, y_2^{(k)})$ is descent.

Algorithm 2, COVLS, starts with a line search method followed by a constrained minimization using the Newton step. Under Condition (4.13), f is descent under the line search. And the Newton step is a standard optimization method and it is proved descent and convergent. Therefore, Algorithm 2, COVLS, is descent and convergent under the condition (4.13) holds.

The condition (4.13) can easily be extended to a matrix form. And when the condition is true, f is descent under the line search in the $\mathbf{\Theta}$ step. Let $P_{\mathbf{B},1}(\mathbf{B}) = P_{\mathbf{B},2}(\mathbf{B}) = 0$ and $P_{\mathbf{\Theta},1}(\mathbf{\Theta}) = P_{\mathbf{\Theta},2}(\mathbf{\Theta}) = 0$ and use Lemmas 2.1, 2.2 and 2.3, we can conclude that LSNCM is a block coordinate descent algorithm and it is convergent.

4.4 Numerical Experiments

In this section, we evaluate the convergence performance of Algorithm 2, COVLS, Algorithm 4, LSNCM, and Algorithm LSNCM_IV. For the former algorithm, we considered a constrained Graphical sparse VAR(p) (CGsVAR(p)) models in Yuen et al. (2018) and for the latter two algorithms, MAR(p) models in Chapter 3 are consid-

ered. We would like to demonstrate the convergence of the iterates for these models using our algorithms and to make a comparison of existing estimation methods.

Samples of the models considered were those which could not be estimated by algorithms without imposing the positive definiteness constraints of the precision matrices. In the minimization of the objective function (f), i.e. the negative log-likelihood function values, the calculation of the $\log(\det(\boldsymbol{\Theta}))$ was required. Thus the precision matrices must be positive definite. When an iterate of the precision matrix is non-positive definite, the calculation of the $\log(\det(\boldsymbol{\Theta}))$ would give numerical errors. And the algorithm stopped at this point.

In addition, we considered some cases, which had MLE of the precision matrices having very close to zero determinants. This caused a termination of the estimation without solutions.

All these examples were re-estimated using our algorithms, COVLS, LSNCM and LSNCM_IV. The iterates convergence plots and root mean squares errors on estimates were produced.

4.4.1 Algorithm COVLS

We chose the constrained graphical sparse VAR(p) model in Yuen et al. (2018) for the demonstration of Algorithm 2, COVLS. The model is a vector time series model and it can be estimated via the following optimization problem:

$$\begin{aligned} \min_{\mathbf{B},\boldsymbol{\Sigma}_u^{-1}} \quad & -\left\{ -\frac{KT}{2}\log 2\pi + \frac{T}{2}\log\det(\boldsymbol{\Sigma}_u^{-1}) \right. \\ & \left. -\frac{1}{2}\mathrm{tr}\left((\mathbf{Y}-\mathbf{BZ})^T\boldsymbol{\Sigma}_u^{-1}(\mathbf{Y}-\mathbf{BZ})\right)\right\} \qquad (4.14) \\ \text{subject to} \quad & \begin{cases} \mathbf{C}\beta & = & \mathbf{0}, \\ (\boldsymbol{\Sigma}_u^{-1})_{ij} & = & 0, \quad (i,j)\in\Omega, \\ \boldsymbol{\Sigma}_u^{-1} & \succ & 0, \end{cases} \end{aligned}$$

where $\beta = \text{vec}(\mathbf{B})$, $\mathbf{C}$ is a matrix of known constants with full row rank, and $\mathbf{0}$ is a vector or zeros. The above problem adopts the traditional estimation, which minimizes the minus of the log-likelihood function in the vectorized form of the coefficient and precision matrices. The positive definiteness of the $vech(\mathbf{\Theta})$ was not easy to be written as either inequality or equality constraints in the optimization and therefore, the positive definiteness of the precision matrix criterion was not included as constraints in every step of estimation.

A partial correlation graph was investigated to obtain the zero constraints in coefficient matrix $\mathbf{B}$ and the precision matrix $\mathbf{\Theta}$. A MATLAB function for constrained minimization of the negative log-likelihood function, the *fmincon* procedure, was selected for the estimation of the parameters in vector form and the constraints were the zero constraints determined by the sample partial correlation graph. Initial values were taken as their maximum likelihood estimates.

The following constrained graphical VAR(p) models were selected:

1. Model 1: $\mathbf{y}_t = \mathbf{A}_1^{(1)}\mathbf{y}_{t-1} + \mathbf{u}_t, \;\; \mathbf{u}_t \sim N(\mathbf{0}, \mathbf{\Sigma}_1)$; and

2. Model 2: $\mathbf{y}_t = \mathbf{A}_1^{(2)}\mathbf{y}_{t-1} + \mathbf{A}_2^{(2)}\mathbf{y}_{t-2} + \mathbf{u}_t, \;\; \mathbf{u}_t \sim N(\mathbf{0}, \mathbf{\Sigma}_2)$;

where

$$\mathbf{A}_1^{(1)} = \begin{pmatrix} 0.2177 & 0.3066 & 0 & 0 & 0 & 0.3775 \\ -0.6324 & -0.665 & 0.0214 & 0 & 0 & 0 \\ 0 & -0.2749 & -0.7509 & 0.4482 & 0 & 0 \\ 0 & 0 & -0.3046 & -0.8066 & 0.994 & 0 \\ 0 & 0 & 0 & -0.7313 & 0.5054 & 0.7959 \\ -0.0587 & 0 & 0 & 0 & -0.514 & -0.947 \end{pmatrix},$$

$$\mathbf{\Sigma}_1^{-1} = \begin{pmatrix} 1 & 0.4 & 0 & 0 & 0 & 0.4 \\ 0.4 & 1 & 0.4 & 0 & 0 & 0 \\ 0 & 0.4 & 1 & 0.4 & 0 & 0 \\ 0 & 0 & 0.4 & 1 & 0.4 & 0 \\ 0 & 0 & 0 & 0.4 & 1 & 0.4 \\ 0.4 & 0 & 0 & 0 & 0.4 & 1 \end{pmatrix},$$

$$\mathbf{A}_1^{(2)} = \begin{pmatrix} -0.6 & 0.4 & 0 & 0 & 0 & 0.4 \\ 0.4 & -0.6 & 0.4 & 0 & 0 & 0 \\ 0 & 0.4 & -0.6 & 0.4 & 0 & 0 \\ 0 & 0 & 0.4 & -0.6 & 0.4 & 0 \\ 0 & 0 & 0 & 0.4 & -0.6 & 0.4 \\ 0.4 & 0 & 0 & 0 & 0.4 & -0.6 \end{pmatrix},$$

$$\mathbf{A}_2^{(2)} = \begin{pmatrix} -0.3 & 0.2 & 0 & 0 & 0 & 0.2 \\ 0.2 & -0.3 & 0.2 & 0 & 0 & 0 \\ 0 & 0.2 & -0.3 & 0.2 & 0 & 0 \\ 0 & 0 & 0.2 & -0.3 & 0.2 & 0 \\ 0 & 0 & 0 & 0.2 & -0.3 & 0.2 \\ 0.2 & 0 & 0 & 0 & 0.2 & -0.3 \end{pmatrix}, \text{ and}$$

$$\mathbf{\Sigma}_2^{-1} = \begin{pmatrix} 1 & -0.3 & 0 & 0 & 0 & -0.3 \\ -0.3 & 1 & -0.3 & 0 & 0 & 0 \\ 0 & -0.3 & 1 & -0.3 & 0 & 0 \\ 0 & 0 & -0.3 & 1 & -0.3 & 0 \\ 0 & 0 & 0 & -0.3 & 1 & -0.3 \\ -0.3 & 0 & 0 & 0 & -0.3 & 1 \end{pmatrix}.$$

Model 1 is a six-dimensional VAR(1) model and its coefficient and precision matrices are Toeplitz. The MATLAB *fmincon* procedure fails to estimate the model for Samples, d4 with length 200, d10 with length 200, d11 with length 500 and d1 with length 1000 and therefore, they were selected for investigation.

The general phenomena observed in the constrained Newton step in the *fmincon* procedure in MATLAB that the norm of the first iteration steps was greater than 1, relatively large compared with the magnitude of individual cell parameter matrices value in general. Besides this, the maximum of the absolute first-order derivatives was far away from zeros. In every sample, the objective function $f(= -l)$ descended dramatically and the precision matrix of the last iterate had negative eigenvalues. This indicated that the iterated precision matrices were not positive definite and were outside the feasible region. In particular, Sample d4 gave the last iterate gave a complex value of the negative log-likelihood function f value, which should not exist. As a result, the *fmincon* procedure produced no solutions. A summary of the *fmincon* failure results was tabulated in Table 4.1.

Figures 4.1, 4.2, 4.3 and 4.4 gave the convergence plots in grey-green coloured

Table 4.1: Summary of CGsVAR model failure cases in the *fmincon* procedure

Model	Sample	T	Initial f	f after 1st iteration	Last Iteration		
					iteration no.	f value	minimum eigenvalue of $\mathbf{\Theta}$
1	d1	100	905.3	-19671.1	1000	-1.73E+08	-43.6
1	d4	200	1780.8	-5474.0	13	-1.28E+09	-76.8
1	d10	200	1761.4	1761.4	1000	-6.26E+10	-283.9
1	d11	500	4545.0	4545.0	49	-1.39E+14	-2625.6
2	d3	100	833.3	-24427.7	407	-3.95E+13	-4639.7
2	d5	200	1705.3	-800.4	37	-2.90E+18	-189400.0
2	d6	200	1742.1	-27256.8	1000	-9.76E+18	-243270.0
2	d7	200	1725.1	-13422.0	1000	-1.27E+12	-1204.9
2	d8	500	4376.8	2348.9	1000	-7.48E+18	-163500.0
2	d9	500	4456.2	-7561.5	722	-7.34E+30	-1737200000.0

Remarks: Models 1 and 2 are CGsVAR(1) and CGsVAR(2) models respectively.

lines, run by the *fmincon* procedure. It would be observed in Figure 4.1 for Sample d4 that the estimated negative log-likelihood function, f descended extremely fast to -1×10^9 in less than 10 steps and could not converge. For Sample d10, it would be observed in Figure 4.2 that the estimated negative log-likelihood function, f descended extremely fast to -1×10^9 in 6 steps and was convergent to -6×10^{10} but the iterate did not fall into the feasible region. It would be observed in Figure 4.3 for Sample d11 that the estimated negative log-likelihood function, f descended extremely fast to a negative value and kept descending to -1.4×10^{14}. For the last case of Sample d1, Figure 4.4 shows that the estimated negative log-likelihood function, f descended extremely fast to -1×10^8 in less than 10 steps. Although it was convergent, the iterate did not fall into the feasible region.

Model 2 is a six-dimensional VAR(2) model and has two Toeplitz coefficient matrices and a Toeplitz precision matrix. The MATLAB *fmincon* procedure failed to estimate the model for Samples, d3 with length 100, d5, d6, d7 with length 200, d8 and d9 with length 500 of Model 2 and therefore, they were selected for investigation.

Similar *fmincon* procedure failure in convergence patterns of the four samples

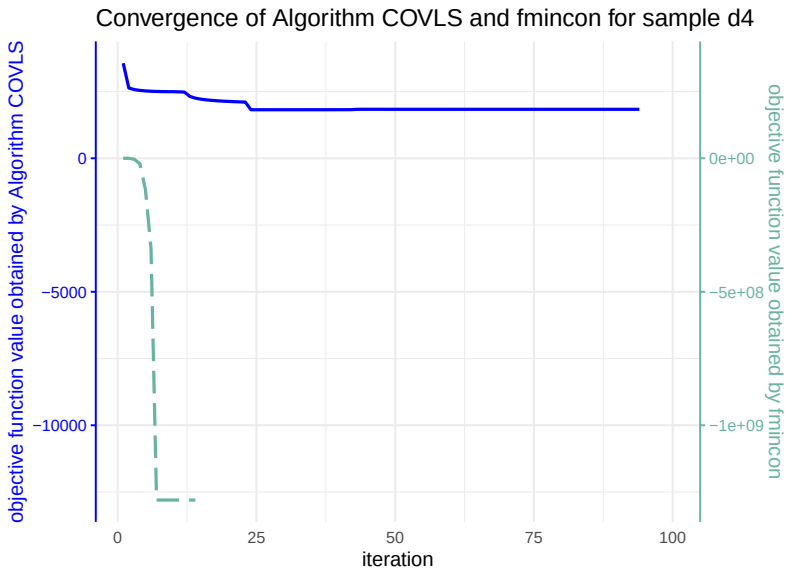


Figure 4.1: The COVLS algorithm and *fmincon* procedure convergence plot for Sample 4 of Model 1 (*fmincon* in grey green line converged to an infeasible point)

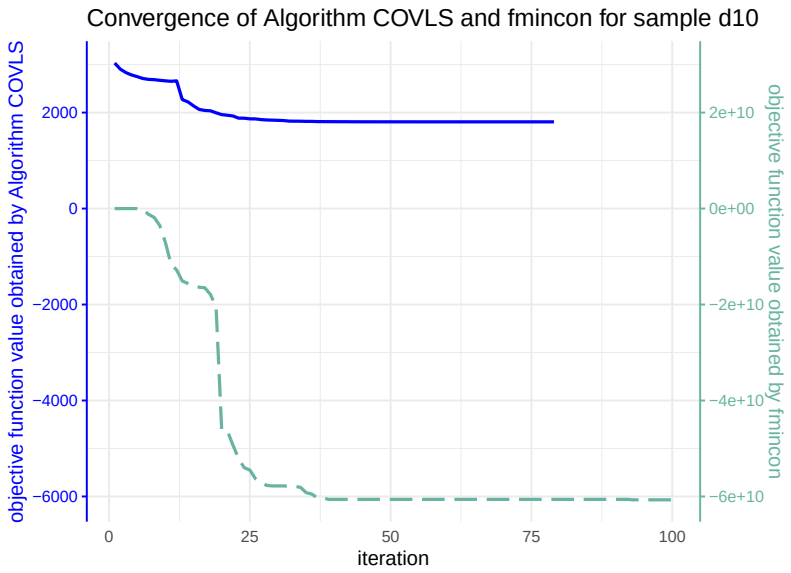


Figure 4.2: The COVLS algorithm and *fmincon* procedure convergence plot for Sample 10 of Model 1 (*fmincon* in grey green line converged to an infeasible point)

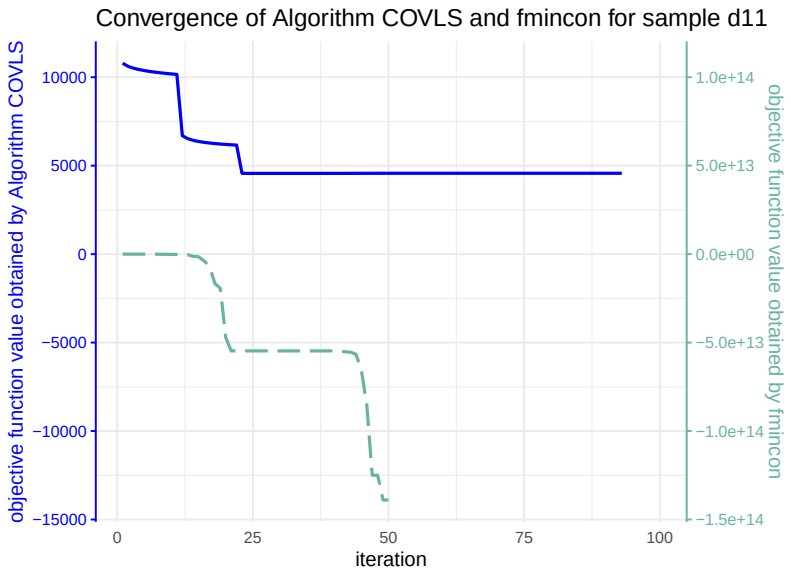


Figure 4.3: The COVLS algorithm and *fmincon* procedure convergence plot for Sample 11 of Model 1 (*fmincon* in grey green line kept descending)

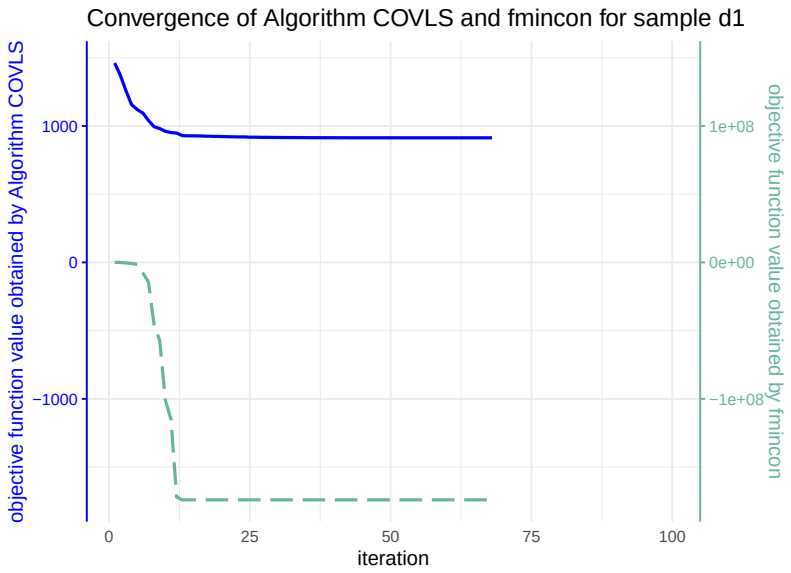


Figure 4.4: The COVLS algorithm and *fmincon* procedure convergence plot for Sample 1 of Model 1 (*fmincon* in grey green line converged to an infeasible point)

of Model 1 were summarized in Table 4.1. The *fmincon* procedure would not give convergent iterates or convergent iterates in an infeasible region. For samples d7 and d11, the direct step in the *fmincon* procedure gave complex log-likelihood function values, which are outside the feasible region. After some iterations in each run, the $vech(\mathbf{\Theta})$ ended up with a non-positive definite $\mathbf{\Theta}$ with negative eigenvalues.

Convergence plots of these six samples were given in Figures 4.5, 4.6, 4.7, 4.8, 4.9 and 4.10. The grey-green coloured lines gave the trace of *fmincon* procedure iterates.

It would be observed in Figure 4.5 for Sample d3 that the estimated negative log-likelihood function, f kept descending to -7×10^{11} and could not converge. For Sample d5, it would be observed in Figure 4.6 that the estimated negative log-likelihood function, f, descended to -2.9×10^{18} and failed to converge. It would be observed in Figure 4.7 for Sample d6 that the estimated negative log-likelihood function, f descended to negative values and converged to -8×10^{18}, but the iterate did not fall into the feasible region. For the case of Sample d7, it would be observed in Figure 4.8 that the estimated negative log-likelihood function, f descended and was convergent to -1.2×10^{12} but the iterate did not fall into the feasible region. It would be observed in Figure 4.9 that the estimated negative log-likelihood function, f descended quickly to negative values and was convergent to -5.5×10^{13} but the iterate did not fall into the feasible region. For the last case of Sample d1, it would be observed in Figure 4.10 that the estimated negative log-likelihood function, f descended to -4.0×10^{20} but it was not convergent.

We repeated to estimate constrained graphical VAR(p) models for these ten samples using our algorithm COVLS.

Our algorithm first applied the line search to find a new vector to replace the MLE initial value vector using a step size of one-tenth of the norm size in the first iteration in the *fmincon* procedure. We continued to apply the line search and examine whether the precision matrix was positive definite in each step. When the

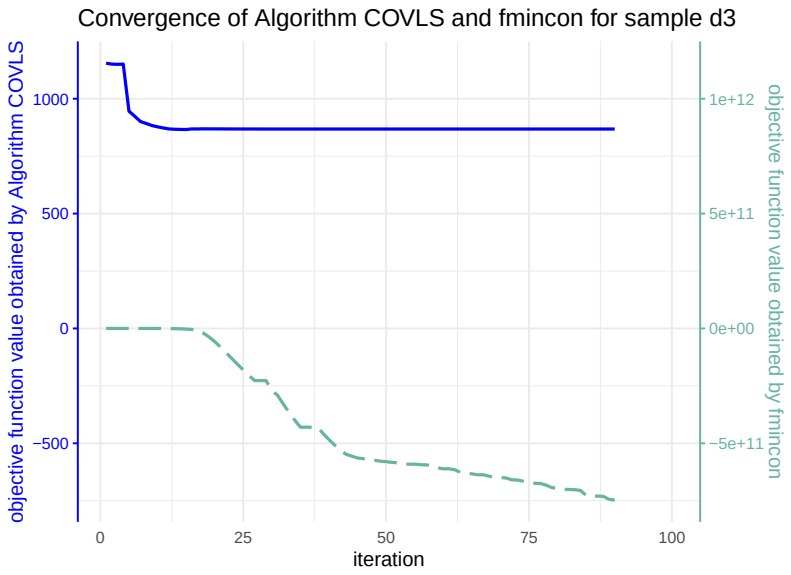


Figure 4.5: The COVLS algorithm and *fmincon* procedure convergence plot for Sample 3 of Model 2 (*fmincon* in grey green line kept descending)

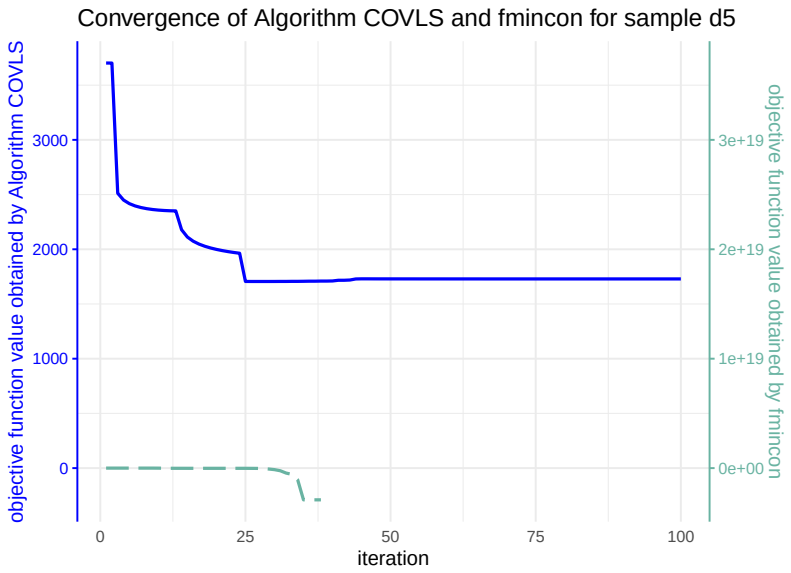


Figure 4.6: The COVLS algorithm and *fmincon* procedure convergence plot for Sample 5 of Model 2 (*fmincon* in grey green line kept descending)

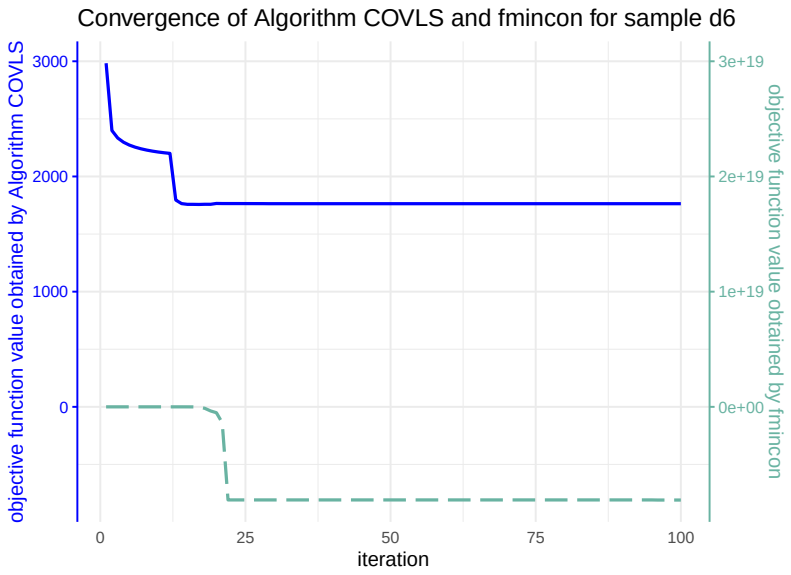


Figure 4.7: The COVLS algorithm and *fmincon* procedure convergence plot for Sample 6 of Model 2 (*fmincon* in grey green line kept descending)

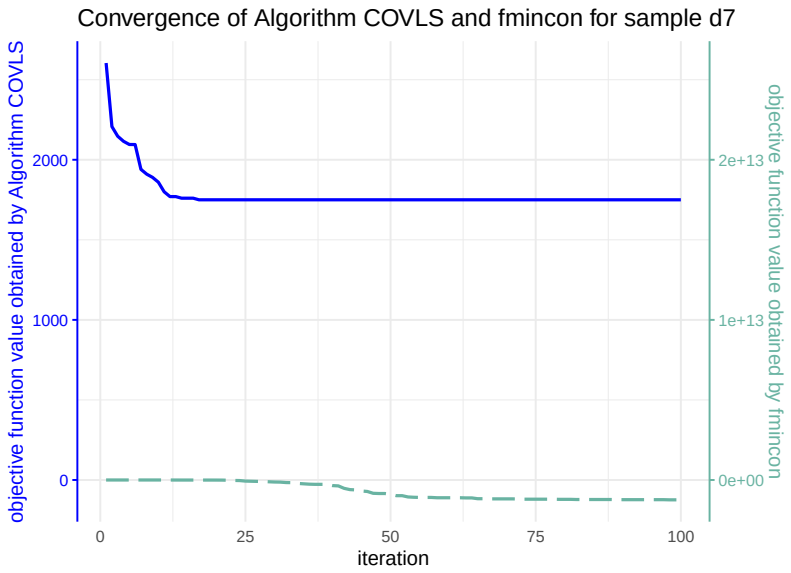


Figure 4.8: The COVLS algorithm and *fmincon* procedure convergence plot for Sample 7 of Model 2 (*fmincon* in grey green line kept descending)

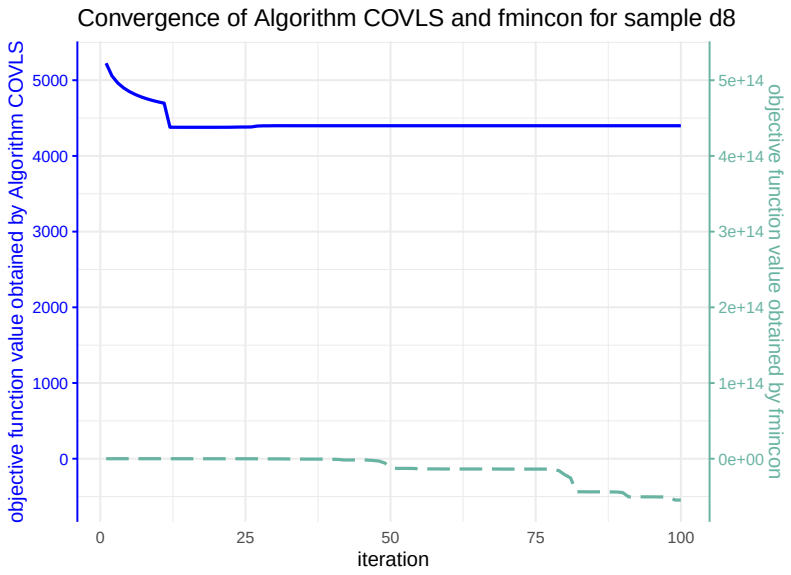


Figure 4.9: The COVLS algorithm and *fmincon* procedure convergence plot for Sample 8 of Model 2 (*fmincon* in grey green line kept descending)

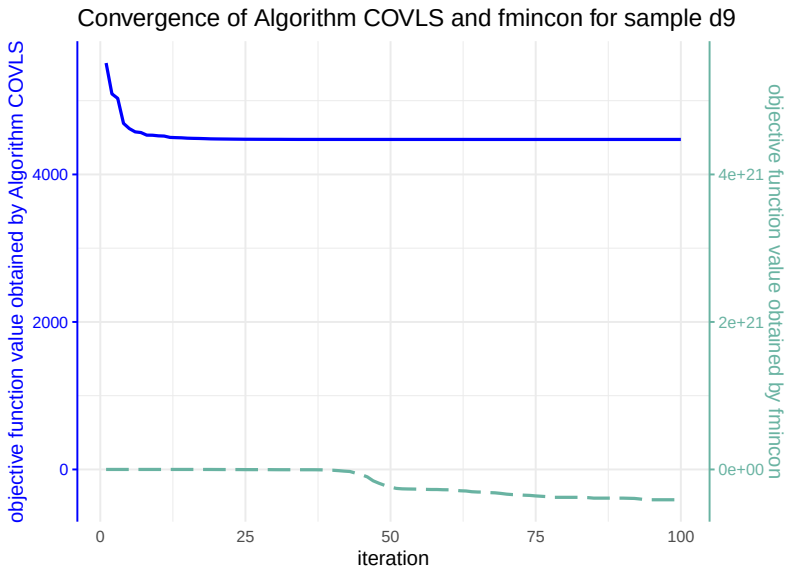


Figure 4.10: The COVLS algorithm and *fmincon* procedure convergence plot for Sample 9 of Model 2 (*fmincon* in grey green line kept descending)

precision matrix for the next iterate was not positive definite, we replaced it with its projection in the positive definite cone, i.e. a symmetric positive definite matrix with the minimum norm with τ starting from 0.01 to $\min(0.1, \theta_{11}, \ldots, \theta_{K,K})$, where θ_{ii} is the i-th element of $\mathbf{\Theta}$. When $\mathbf{\Theta}$ was guaranteed positive definite, we conducted the same *fmincon* procedure to the iterate for minimizing the negative log-likelihood function.

We studied the convergence plots of our algorithm COVLS for Samples d4, d10, d11 and d1 of Model 1 as shown in Figures 4.1, 4.2, 4.3, and 4.4, and d3, d5, d6, d7, d8 and d9 of Model 2 as shown in Figures 4.5, 4.6, 4.7, 4.8, 4.9 and 4.10. The trace of convergence of our algorithm COVLS was plotted in blue colour. The trace began with a line search using a projected precision matrix in a positive definite cone, when the precision matrix iterate was not positive definite. And then the trace was the constrained minimization estimates of the parameter vector. The estimates are solved by the *fmincon* function in MATLAB with the zero constraints. It could be seen in all these figures that the objective functions $f(= -l)$ decreased and their iterates were convergent to positive values. Since our samples contained noises, f would be convergent to values, which were close to the negative true log-likelihood values.

To investigate further the performance of the estimate, we measured the closeness between the true coefficient and precision parameters matrices and our estimates using root mean squares. We tabulated the root mean squares ($RMSE_{\mathbf{B}}$) for the coefficient matrix, $\mathbf{B}$ and $RMSE_{\mathbf{\Theta}}$ for the precision matrix ($\mathbf{\Theta}$) from the true matrices in Table 4.2.

The estimates generated by our algorithm COVLS gave reasonable root mean squares. The four tested Samples of Model 1 had $RMSE_{\mathbf{B}}$'s varying from 0.09 to 0.21 and $RMSE_{\mathbf{\Theta}}$'s varying from 0.22 to 0.6. As the sample size (T) increased, $RMSE_{\mathbf{B}}$ and $RMSE_{\mathbf{\Theta}}$ were getting smaller. Samples d4 and d10 had both the

Table 4.2: Summary of constrained graphical sparse VAR estimation using Algorithm COVLS

			Final solutions		
Model	Sample	T	f	$RMSE_{\mathbf{B}}$	$RMSE_{\mathbf{\Theta}}$
1	d1	100	901.0	0.2157	0.5955
1	d4	200	1826.1	0.1842	0.3973
1	d10	200	1798.9	0.1467	0.2496
1	d11	500	4559.9	0.0957	0.2258
2	d3	100	858.3	0.4213	0.6375
2	d5	200	1719.6	0.3233	0.3933
2	d6	200	1755.0	0.3595	0.2949
2	d7	200	1738.7	0.3295	0.4261
2	d8	500	4390.7	0.2624	0.2142
2	d9	500	4467.0	0.2884	0.2307

Remarks: Models 1 and 2 are CGsVAR(1) and CGsVAR(1) models respectively.

same sample sizes and their root mean squares were similar for $\mathbf{B}$ and $\mathbf{\Theta}$. We tested six samples with sizes (T) from 100 to 500. Their $RMSE_{\mathbf{B}}$ varied from 0.42 to 0.29 and were almost double the root mean squares of the lag 1 model having the same length. $RMSE_{\mathbf{\Theta}}$ varied from 0.63 to 0.21 and were a bit larger than root mean squares of the lag 1 model with the same length. The smaller the sample size, the larger the root mean squares of $\mathbf{B}$ and $\mathbf{\Theta}$. Samples d5, d6 and d7 had the same length and their root mean squares in $\mathbf{B}$ and $\mathbf{\Theta}$ were similar in magnitude. These phenomena could also be observed in Samples d8 and d9, which both had sample sizes (T) of 500.

4.4.2 Algorithms LSNCM and LSNCM_IV

Algorithms LSNCM and LSNCM_IV are matrix-based and we evaluate these two algorithms using a matrix time series model, MAR(p) model. Its model equation is

$$\mathbf{X}_t = \sum_{k=1}^{p} \mathbf{C}_k \mathbf{X}_{t-1} \mathbf{D}_k{}^T + \mathbf{E}_t, \quad vec(\mathbf{E}_t) \sim N(\mathbf{0}, \mathbf{\Theta}^{-1}),$$

where the dimensions of the data matrix at time t, $\mathbf{X}_t$, the left coefficient matrix, $\mathbf{C}_k$, the right coefficient matrix, $\mathbf{D}_k$ and the precision matrix, $\mathbf{\Theta}$ are $m \times n$, $m \times m$, $n \times n$ and $mn \times mn$ respectively.

The optimization problem of the MAR(p) model is given in Equation (3.4). The algorithm in Section 3.3.1 adopts a block coordinate gradient descent approach to minimize the negative log-likelihood function in coefficient and precision matrices form; however, it stops for some samples.

We chose the samples, which could not be estimated, because the algorithm stopped whenever a non-positive definite precision matrix iterate, $\mathbf{\Theta}^{(1,j)}$ (for some j), in the first iteration was generated and the $\log(\det(\mathbf{\Theta}))$ of the objective function gave the numerical errors.

We also selected other samples, in which initial precision matrices caused numerical errors in the $\log(\det(\mathbf{\Theta}))$ term of the objective function. It is natural to choose the maximum likelihood estimate of the corresponding vector autoregressive model form as an initial value. The initial values, $\mathbf{C}_0$ and $\mathbf{D}_0$ were obtained by the nearest Kronecker product decomposition from the coefficient of the VAR model, while the $\mathbf{\Theta}_0$ term were the corresponding precision matrix. In our models, we observed some cases having the determinant of $\mathbf{\Theta}_0$ very close to zero and the idea of replacement of a non-positive definite precision matrix could also be applied to the initial value.

The details of selected bilinear matrix MAR(p) time series models were tabulated in Tables 4.3 and 4.4 and their model forms are listed below.

1. Model p823: $m = 6, n = 4$,

$$\mathbf{X}_t = \mathbf{C}_1^{(1)}\mathbf{X}_{t-1}{\mathbf{D}_1^{(1)}}^T + \mathbf{C}_2^{(1)}\mathbf{X}_{t-2}{\mathbf{D}_2^{(1)}}^T + \mathbf{C}_3^{(1)}\mathbf{X}_{t-3}{\mathbf{D}_3^{(1)}}^T + \mathbf{E}_t,$$
$$vec(\mathbf{E}_t) \sim N(\mathbf{0}, \mathbf{\Theta}_1^{-1}),$$

2. Model p824: $m = 9, n = 6$,

$$\mathbf{X}_t = \mathbf{C}_1^{(2)}\mathbf{X}_{t-1}{\mathbf{D}_1^{(2)}}^T + \mathbf{C}_2^{(2)}\mathbf{X}_{t-2}{\mathbf{D}_2^{(2)}}^T + \mathbf{E}_t, \;\; vec(\mathbf{E}_t) \sim N(\mathbf{0}, \mathbf{\Theta}_2^{-1}),$$

3. Model p825: $m = 9, n = 6$,

$$\mathbf{X}_t = \mathbf{C}_1^{(3)}\mathbf{X}_{t-1}\mathbf{D}_1^{(3)^T} + \mathbf{C}_2^{(3)}\mathbf{X}_{t-2}\mathbf{D}_2^{(3)^T} + \mathbf{C}_3^{(3)}\mathbf{X}_{t-3}\mathbf{D}_3^{(3)^T} + \mathbf{E}_t,$$
$$vec(\mathbf{E}_t) \sim N(\mathbf{0}, \mathbf{\Theta}_3^{-1}).$$

The left and right coefficient matrices are randomly generated from a square matrix with a specified spectral radius and then the left coefficient matrices are denormalized. The precision matrix is generated from mn random positive eigenvalues. Each model satisfies the stability condition stated in Section 3.1.2. For the first model, the spectral radii of $\mathbf{C}_i^{(1)}(i = 1, 2, 3)$ and $\mathbf{D}_i^{(1)}(i = 1, 2, 3)$ are 0.5521, 0.5558, 0.5866 and 0.8269, 0.8154, 0.7914 respectively and $\det(\mathbf{\Theta}_1)$ is 69.02. For the second model, the spectral radii of $\mathbf{C}_i^{(2)}(i = 1, 2)$ and $\mathbf{D}_i^{(2)}(i = 1, 2)$ are 0.5608, 0.5585 and 1.08567 0.6512 respectively and $\det(\mathbf{\Theta}_2)$ is 0.0168. For the third model, the spectral radii of $\mathbf{C}_i^{(3)}(i = 1, 2, 3)$ and $\mathbf{D}_i^{(3)}(i = 1, 2, 3)$ are 0.4545, 0.4978, 0.4778 and 0.5960, 0.5597, 0.5914 respectively and $\det(\mathbf{\Theta}_3)$ is 42.67. These three models fulfill the stability conditions of the time series. The left and right coefficient matrices of the first model are tabulated as below for reference.

$$\mathbf{C}_1^{(1)} = \begin{pmatrix} -0.0432 & -0.088 & 0.2079 & -0.2285 & -0.0163 & 0.0123 \\ 0.0187 & -0.0394 & 0.0339 & -0.0662 & 0.2406 & -0.1085 \\ -0.1192 & -0.4294 & 0.1034 & 0.1793 & -0.1255 & 0.2806 \\ 0.2001 & 0.177 & -0.0294 & 0.3293 & -0.0889 & 0.1211 \\ -0.0859 & 0.1044 & -0.0612 & -0.3454 & 0.0878 & -0.2659 \\ -0.019 & 0.012 & 0.1536 & 0.18 & -0.042 & 0.0208 \end{pmatrix},$$

$$\mathbf{C}_2^{(1)} = \begin{pmatrix} -0.0162 & -0.2708 & 0.0317 & -0.0833 & -0.0284 & -0.1319 \\ -0.0504 & 0.0386 & 0.2334 & 0.0385 & 0.1857 & -0.1183 \\ 0.0843 & 0.2862 & -0.0006 & 0.0558 & -0.0075 & -0.2305 \\ 0.2235 & 0.1237 & 0.2083 & 0.2049 & 0.1118 & -0.404 \\ -0.0452 & -0.0989 & 0.0768 & -0.1895 & -0.0981 & 0.0257 \\ -0.1816 & -0.0446 & 0.0884 & 0.405 & -0.1902 & 0.1148 \end{pmatrix},$$

$$\mathbf{C}_3^{(1)} = \begin{pmatrix} -0.1064 & 0.0595 & -0.0876 & -0.0413 & 0.0195 & 0.0913 \\ 0.2 & -0.0487 & -0.0258 & 0.1296 & -0.0803 & 0.0205 \\ -0.0246 & -0.0287 & -0.2915 & -0.038 & 0.1352 & -0.1165 \\ -0.1142 & -0.1177 & 0.0623 & 0.0219 & 0.0665 & -0.0367 \\ -0.1656 & 0.2476 & 0.1247 & -0.3681 & -0.0432 & -0.2244 \\ 0.0784 & 0.0421 & -0.1527 & 0.3359 & 0.0689 & -0.5503 \end{pmatrix},$$

Table 4.3: Performance of Algorithm LSNCM for MAR(p) models failure cases (All samples have non-pd $\mathbf{\Theta}^{(1)}$ in line search in the first iteration)

					Algorithm LSNCM			
Model	m	n	lag order	T	no. of times $\mathbf{\Theta}^{(k)}$ calibrated	$RMSE_{\mathbf{A}}$	$RMSE_{\mathbf{B}}$	$RMSE_{\mathbf{\Theta}}$
p824	9	6	2	200	21	0.0488	0.0763	0.2883
p824	9	6	2	200	34	0.0483	0.0705	0.2829
p825	9	6	3	200	72	0.0534	0.0870	0.3844
p825	9	6	3	200	94	0.0599	0.0806	0.4331

$$\mathbf{D}_1^{(1)} = \begin{pmatrix} -0.0199 & -0.4389 & 0.136 & 0.1424 \\ 0.1778 & -0.0369 & -0.4683 & -0.3217 \\ -0.5169 & 0.3497 & -0.023 & 0.4342 \\ 0.4581 & 0.0892 & 0.4955 & -0.519 \end{pmatrix}, \mathbf{D}_2^{(1)} = \begin{pmatrix} 0.6528 & 0.4144 & -0.0988 & 0.0862 \\ 0.3046 & -0.7051 & 0.233 & -0.3457 \\ 0.1781 & -0.2759 & 0.1001 & 0.5187 \\ -0.4753 & 0.0573 & 0.1455 & 0.5294 \end{pmatrix},$$

$$\mathbf{D}_3^{(1)} = \begin{pmatrix} 0.0256 & 0.4615 & 0.9347 & -0.1765 \\ 0.4779 & 0.2214 & -0.1995 & -0.1102 \\ -0.5671 & 0.2235 & -0.0768 & 0.3953 \\ 0.1364 & 0.6246 & -0.0192 & -0.3386 \end{pmatrix}.$$

Note that these three models are not sparse. Model p824 is a MAR(2) model and Models p823 and p825 are MAR(3) models.

We repeated to estimate these models by LSNCM algorithm so that the occurrence of the non-positive definite precision matrix would not generate numerical errors by replacing the matrix with a 'closest positive definite symmetric matrix in the iterations and to estimate the model with invalid precision matrix initial values by LSNCM_IV algorithm. Then both algorithms skipped the non-positive definite precision matrix and could produce the next iterate until the convergence of iterates.

Figures 4.11, 4.12, 4.13 and 4.14 are the convergence plots of the models and they illustrate that the block coordinate gradient descent algorithm using line search (LS) stopped in the first iteration. Only pink dots are marked at iteration zero for the block coordinate gradient descent algorithm using line search (LS), while the f value generated by the LSNCM algorithm, represented by blue lines was convergent.

Similar convergent patterns for the LSNCM_IV algorithm were observed for the case, which had an invalid precision matrix initial value, in the convergence plots in

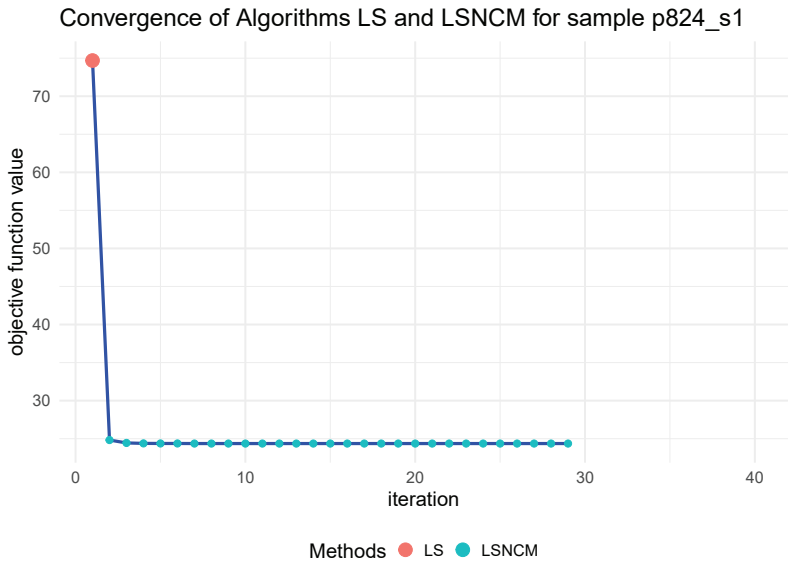


Figure 4.11: LSNCM and LS algorithms convergence plot for Sample 1 of Model p824

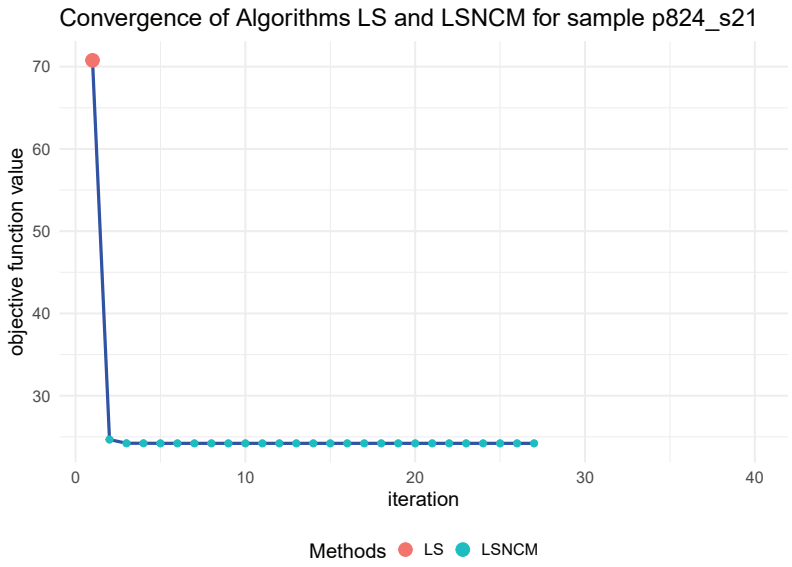


Figure 4.12: LSNCM and LS algorithms convergence plot for Sample 21 of Model p824

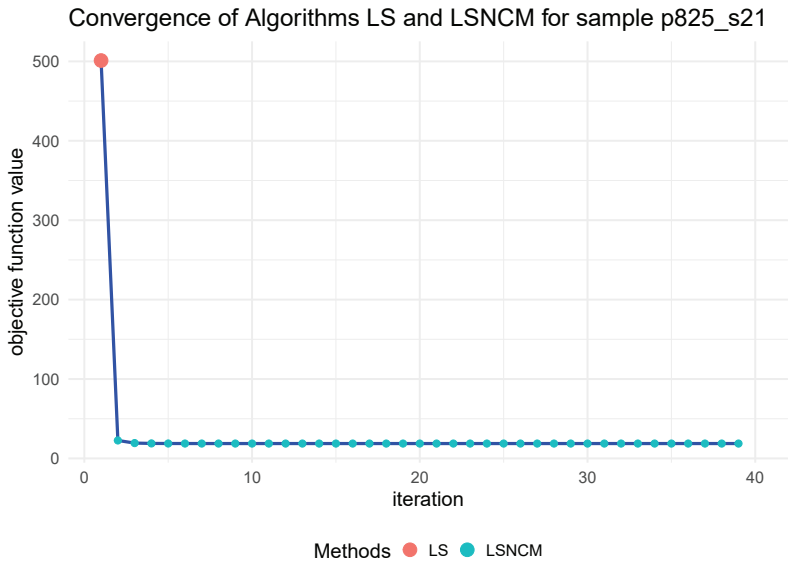


Figure 4.13: LSNCM and GIST algorithms convergence plot for Sample 21 of Model p825

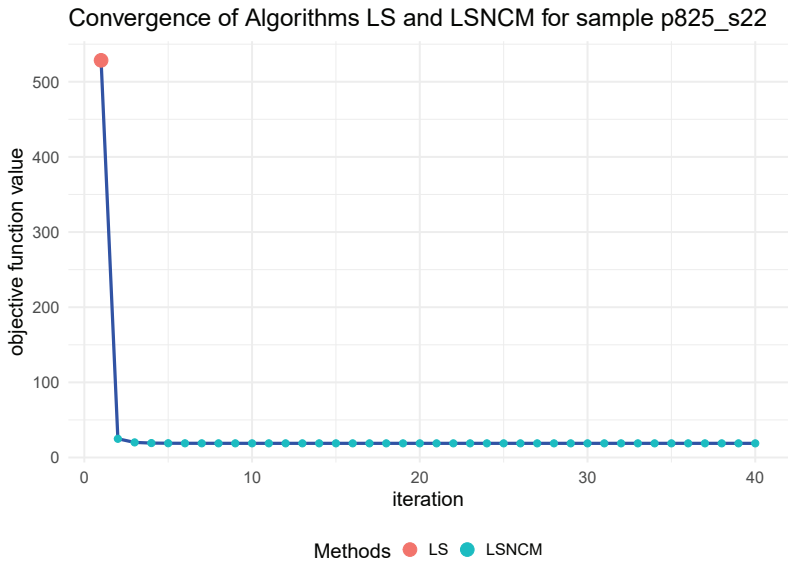


Figure 4.14: LSNCM and GIST algorithms convergence plot for Sample 22 of Model p825

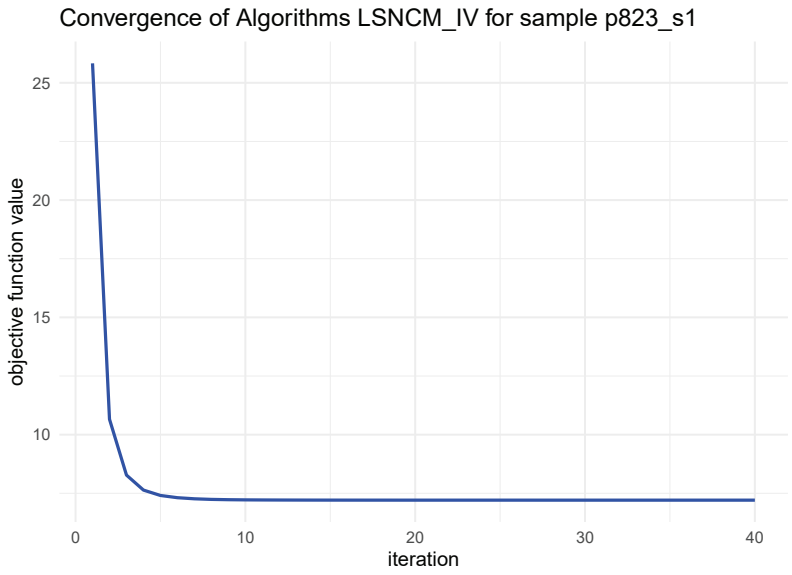


Figure 4.15: LSNCM_IV and LS algorithms convergence plot for Sample 1 of Model p823

Figures 4.15, 4.16, 4.17 and 4.18. As the initial precision matrix had almost zero determinant, the calculation of the initial objective function (f) value failed.

We evaluated the performance of LSNCM in a similar way as the evaluation of COVLS. We calculated the root mean squares for $\mathbf{C}$, $\mathbf{D}$ and $\mathbf{\Theta}$ in Table 4.3. All root mean squares for $\mathbf{C}$, $\mathbf{D}$ and $\mathbf{\Theta}$ using LSNCM were around 0.05, 0.08 and 0.3 to 0.4 respectively. This indicated the algorithm worked well on these samples.

The LSNCM_IV algorithm was also evaluated by the root mean squares, tabulated in Table 4.4. All root mean squares for $\mathbf{C}$, $\mathbf{D}$ and $\mathbf{\Theta}$ using LSNCM were around 0.06, 0.08 to 0.09 and 0.4 for Model p825 and 0.73 for Model p823 respectively. This indicated the algorithm worked well on these samples.

4.4.3 Discussion of computational efficiency of the algorithms

In this section, we discuss the computation efficiency of the algorithms, when the dimension is very large. Note that our algorithms consist of the matrix calibration

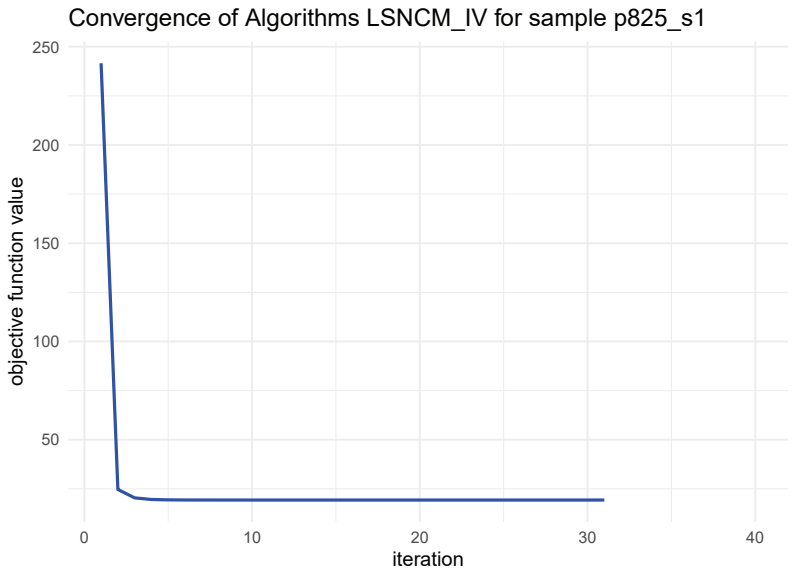


Figure 4.16: LSNCM_IV and LS algorithms convergence plot for Sample 1 of Model p825

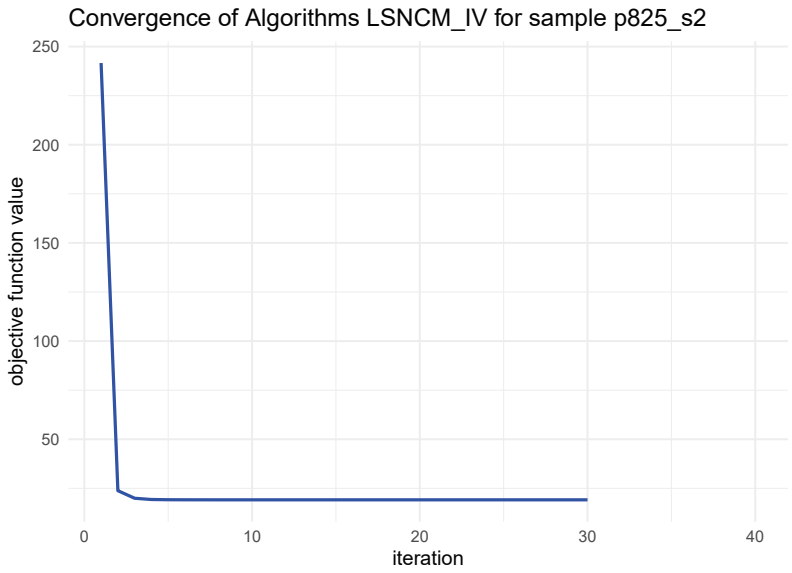


Figure 4.17: LSNCM_IV and LS algorithms convergence plot for Sample 2 of Model p825

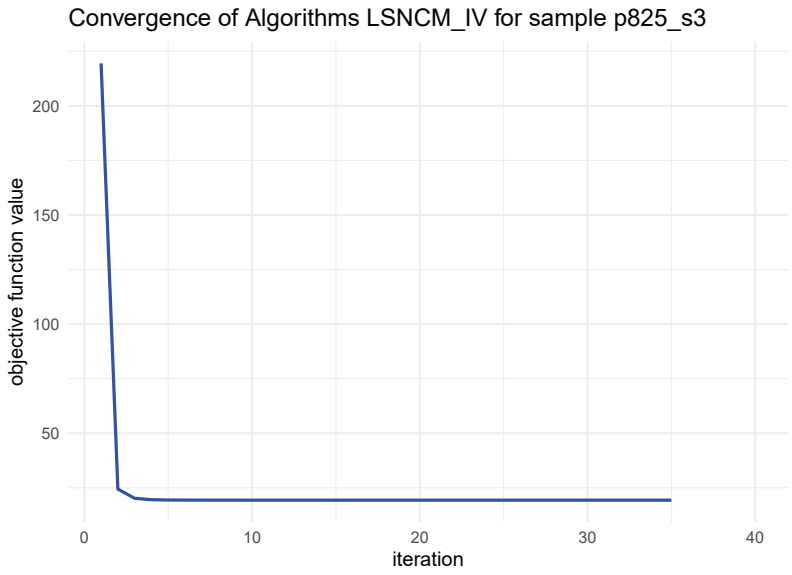


Figure 4.18: LSNCM_IV and LS algorithms convergence plot for Sample 3 of Model p825

Table 4.4: Performance of Algorithm LSNCM_IV for MAR(3) models with almost zero determinants of initial precision matrices

							Algorithm LSNCM_IV				
Model	Sample	$\det(\boldsymbol{\Theta})$	m	n	T	$\det(\boldsymbol{\Theta}_0)$	τ	$\det(\boldsymbol{\Theta}_\tau)$	$RMSE_{\mathbf{C}}$	$RMSE_{\mathbf{D}}$	$RMSE_{\boldsymbol{\Theta}}$
p823	1	69.0	6	4	80	-1.1E-283	0.32	1.68E-08	0.0693	0.1084	0.7367
p825	1	42.7	9	6	200	-4.8E-244	0.03	4.79E-08	0.0587	0.0834	0.3523
p825	2	42.7	9	6	200	6.0E-243	0.03	2.80E-07	0.0531	0.0805	0.3383
p825	3	42.7	9	6	200	-2.9E-246	0.04	1.10E-05	0.0598	0.0816	0.3774

procedure by Qi and Sun (2006) in a single loop and a line search algorithm outside the loop. This implies that the computation time is a sum of the total of line search time used and a product of the number of times of the matrix calibration procedure called and the computation time of the matrix calibration procedure.

We summarize the computation time of two main examples from the numerical experiments in Qi and Sun (2006) in Table 4.5. The matrix calibration procedure is developed based on the Newton method and therefore is marked as "Newton" and the traditional method is the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm.

Case	Algorithm	dim (K)	α	CPU time	Iterations	Residuals
1	Newton	1000	0.01	2 m 13 s	1	2.6×10^{-8}
		1000	0.1	2 m 58 s	3	2.0×10^{-8}
		1000	0.01	3 m 38 s	5	2.7×10^{-8}
		1000	0.01	4 m 13 s	7	9.9×10^{-8}
	BFGS	1000	0.01	2 m 19 s	2	2.3×10^{-7}
		1000	0.1	3 m 03 s	5	8.0×10^{-7}
		1000	0.01	6 m 27 s	18	9.7×10^{-6}
		1000	0.01	15 m 10 s	53	6.4×10^{-6}
2	Newton	500	random	0 m 34 s	8	3.7×10^{-9}
		1000	random	4 m 55 s	9	3.1×10^{-9}
		1500	random	14 m 04 s	9	4.5×10^{-7}
		2000	random	33 m 52 s	9	2.6×10^{-6}
	BFGS	500	random	4 m 46 s	88	9.4×10^{-6}
		1000	random	Failed	110	2.3×10^{-5}
		1500	random	Failed	111	4.7×10^{-5}
		2000	random	Failed	112	8.1×10^{-5}

Table 4.5: Numerical results of matrix calibration used

The models are listed in the following:

1: $\mathbf{G} = \mathbf{C} + \alpha\mathbf{R}$, where $\mathbf{C}$ is a random correlation matrix and $\mathbf{R} = (r_{ij})$ and r_{ij}'s are randomly selected from $[-1, 1]$.

2: $\mathbf{G} + \mathbf{P}$, where $\mathbf{G} = (g_{ij})$ with g_{ii} being randomly selected from $[-2 \times 10^4, 2 \times 10^4]$, $\mathbf{P} = \mathbf{P}^T = (p_{ij})$ and p_{ij}'s are randomly selected from $[-\alpha, \alpha]$, and $\alpha \in \{0, 0.01, 0.1, 1\}$.

Together with the line search in our algorithms, the total running time has two summands. The first summand is less than the sum of the line search running time and the second summand is the product of the number of iterations used in the line search and the running time of the matrix calibration method used above. As given in Table 4.5, it is observed that the CPU time varied from less than one minute up to 34 minutes. The maximum CPU time is less than 34 minutes for a dimension of 2000.

In our simulation study, the number of iterations required for vector autoregressive models with dimension 6 is less than 25, as observed in Figures 4.1 to 4.10. For the cases of matrix autoregressive models with the dimension being up to $(m, n) = (9, 6)$, its covariance matrix is 54×54 and the maximum no. of calibration required is 94 (see Table 4.3). It seems that the no. of matrix calibrations required is less than the number of elements in the matrix required to be estimated. The idea of this line search method comes from the gradient descent algorithm and if the function to be minimized is strongly convex, it is linearly convergent. Note that the matrix calibration method is quadratically convergent. The computation time seems not bad. Further analysis of computational time is required to be conducted.

4.5 Conclusion

Positive definiteness constraints on covariance matrices or precision matrices ($\mathbf{\Theta}$) in the log-likelihood estimation of models are required, but it is complicated to be imposed. When any iterates of these matrices in the maximization are non-positive definite, i.e. in the infeasible region, numerical errors would occur in the $\log(\det(\mathbf{\Theta}))$ term of the log-likelihood function. As a result, the estimation fails.

Traditional methods increase the chances of successful estimation by introducing different model structures so that estimation of the covariance matrix and precision matrix could be estimated via highly reduced dimensions of positive definite matrices. For the precision matrix, the most traditional method is the use of a conditional dependence structure to reduce the number of parameters needed. As the dimension of the problem increases dramatically, the current dimension reduction method might also be large in scale. When a sparse matrix is used for the precision matrix, the matrix will result in a low-ranked matrix and might have negative eigenvalues. Numerical computation errors may not be avoidable and thus, non-positive

definite matrices might be generated in the series of iterates. As a result, both the covariance and precision matrix estimation may stop somewhere or converge to an infeasible point.

This chapter discusses how to replace the positive definiteness constraints by calibrating the precision matrices into symmetric positive definite matrices at every step of the iteration. We have proved that the algorithms are descent. It has been discussed that the proposed algorithms are convergent under certain regularity conditions and our convergence analysis was conducted with a vector and a matrix time series model examples for illustration. In addition, our method has been applied to the cases, where the initial values of the precision matrices were non-positive definite.

The success of the simulation study of our algorithms inspires us to study further the problem and extend our algorithm to the estimation of covariance matrix in other statistical models. In addition, it would be much better to embed calibration of covariance matrix techniques in the line search algorithm and the constrained Newton Method so that covariance or precision matrices of some statistical models are properly estimated with ease. Computation efficiency is also valuable to be investigated.

Chapter 5

Conclusions

This chapter draws conclusions on the thesis, and discusses some possible relevant research directions in near future.

5.1 Discussions and Conclusions

1. Chapter 2 proposes a new sparse graphical time series model. It selects a final model from all possible sparse Gaussian graphical models and sparse vector autoregressive models, using the minimum Bayesian Information Criterion. Therefore, the sparsity combination of AR coefficients and precision matrices is optimal. No pre-estimates based on AR coefficients, partial correlations or spectral coherence are required for the sparsity structure identification and it allows any free combination of sparsity between AR coefficient and precision matrices. We have proved that the penalized maximum likelihood estimators of the model are consistent and converge to asymptotic normal distributions. A new, effective and convergent iterative alternating algorithm based on LASSO, SCAD and MCP penalized likelihood estimation for the sparse model was set up. The MCP penalty is a non-convex penalty and normally needs a linearization in the penalized estimation. But our algorithm allows the linearization in the penalized likelihood estimation and gives estimates. Our algorithm does

not require a Hessian matrix and enables us to obtain the iterative estimates using independent elementwise closed-form solutions, which allow parallel programming within the same iteration. This makes complexity of the algorithm not increase much, as the dimension increases.

2. Chapter 3 studies the matrix time series model, where Chen et al. (2021) adopted the bilinear regression model onto a time series with matrix-variate with a structured covariance tensor. We extend the model to a higher lag order and introduce the general covariance structure, so that any data with the imperfect independent relationship over two classifications could be modelled. In addition, a graphical model is merged with our matrix time series model to form a graphical matrix time series model. We adopt the optimal sparsity concept as in Chapter 2 for our sparse model and LASSO penalized estimation is used. The economic indicator example demonstrated that our MAR model had higher lag and had a lower residual sum of squares value and a prediction error sum of squares value. The sparse model of the example exhibited an intuitively correct economic relationship between the five countries.

3. Chapter 4 discusses the problems arising from estimating high-dimensional covariance matrix. Due to a limited number of observations, the covariance might be low rank or not positive definite. When this happens in a covariance estimation algorithm, the algorithm fails to have a solution. However, the positive definiteness property of the covariance matrix is not easily coded as equality or inequality constraints in optimization. We aim to replace the positive definiteness constraints by calibrating the precision matrices iterates into symmetric positive definite matrices in every step of iteration. We have discussed that the algorithms are descent under certain regularity conditions. Convergence analysis were conducted with a vector and a matrix time series model examples for

illustration. In addition, our method can also be applied to the cases, where the precision matrix initial values are non-positive definite.

5.2 Future Works

We have started with the sparse graphical vector time series modelling and this model has been extended to a matrix form, using the LASSO penalty. As observed in our works the MCP penalty on sparse graphical vector time series model was the best, it is suggested to study further on the use of MCP penalty on sparse matrix time series model. As the sparse matrix time series model uses a general precision matrix, further reduction on dimension by using a structured covariance tensor will lead to a simpler model for data with a strong relationship between row-wise interactions and column-wise dependency. Furthermore, we can consider applying the MAR model to volatility modelling, for example on GARCH modelling (Engle (1982)).

On the estimation algorithm, we have demonstrated that the use of calibration of a covariance/precision matrix to replace the non-positive definite matrix iterate in two algorithms for minimization of a negative log-likelihood function is successful and it is much simpler than imposing positive definiteness constraints of the covariance/precision matrix. It is interesting to further investigate the algorithm and combine the theory of calibration of covariance/precision matrix with the theory of the constrained/unconstrained minimization of a negative log-likelihood function so that the approximate optimization problem can be solved in a more efficient algorithm. This idea can also be extended to more probabilistic models.

Bibliography

Akaike, H. (1973), "Information Theory and an Extension of the Maximum Likelihood Principle," in *2nd International Symposium on Information Theory*, eds. B. Petrov and F. Csáki, pp. 267–281, Académiai Kiadó, Budapest.

Bai, J. and Ng, S. (2011), "Determining the Number of Factors in Approximate Factor Models," *Econometrica*, 70, 191–221.

Bai, J. and Shi, S. (2011), "Estimating High Dimensional Covariance Matrices and its Applications," *Annals of economics and finance*, 12, 199–215.

Banerjee, O. and d'Aspremont, A. (2007), "Model selection through sparse maximum likelihood estimation," *arXiv.org*.

Banerjee, O., El Ghaoui, L., and d'Aspremont, A. (2008), "Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data," *Journal of Machine Learning Research*, 9, 485–516.

Barzilai, J. and Borwein, J. M. (1988), "Two-point step size gradient methods," *IMA Journal of Numerical Analysis*, 8, 141–148.

Basu, S. and Michailidis, G. (2015), "Regularized estimation in sparse high-dimensional time series models," *The Annals of Statistics*, 43, 1535–1567.

Boyd, S. and Lin, X. (2005), "Least-squares covariance matrix adjustment," *SIAM journal on matrix analysis and applications*, 27, 532–546.

Brillinger, D. R. (1996), "Remarks concerning graphical models for time series and point processes," *Revisita de Econometrica*, 16, 1–23.

Champion, M., Picheny, V., and Vignes, M. (2017), "Inferring large graphs using ℓ_1 -penalized likelihood," *Statistics and Computing*, 28, 905–921.

Chen, R., Xiao, H., and Yang, D. (2021), "Autoregressive models for matrix-valued time series," *Journal of Econometrics*, 222, 539–560.

Chu, T., Zhu, J., and Wang, H. (2011a), "Penalized maximum likelihood estimation and variable selection in geostatistics," *The Annals of statistics*, 39, 2607–2625.

Chu, T., Zhu, J., and Wang, H. (2011b), "Penalized maximum likelihood estimation and variable selection in geostatistics," *arXiv: 1109.0320v1 [stat.ME]*.

Dahl, J., Vandenberghe, L., and Roychowdhury, V. (2008), "Covariance selection for nonchordal graphs via chordal embedding," *Optim Method Softw*, 23, 501–520.

Dahlhaus, R. (2000), "Graphical interaction models for multivariate time series," *Metrika*, 51, 157–172.

Darroch, J. N., Lauritzen, S. L., and Speed, T. P. (1980), "Markov fields and log-linear interaction models for contingency tables," *The Annals of Statistics*, 8, 522–539.

Davis, R. A., Zang, P., and Zheng, T. (2016), "Sparse vector autoregressive modeling," *Journal of Computational and Graphical Statistics*, 25, 1077–1096.

Dempster, A. P. (1972), "Covariance selection," *Biometrics*, 28, 157–175.

Edwards, D. (1995), *Introduction to Graphical Modelling*, Springer-Verlag, New York, N.Y.

Engle, R. F. (1982), "Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation," *Econometrica*, 50, 987–1007.

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

Filzmoser, P., Gschwandtner, M., and Todorov, V. (2012), "Review of sparse methods in regression and classification with application to chemometrics," *Journal of Chemometrics*, 26, 42–51.

Forni, M., Hallin, M., Lippi, M., and Reichlin, L. (2000), "The Generalized Dynamic-Factor Model: Identification and Estimation," *The Review of Economics and Statistics*, 82, 540–554.

Friedman, J., Hastie, T., and Tibshirani, R. (2007), "Sparse inverse covariance estimation with the lasso," *arXiv.org*.

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

Gong, P., Zhang, C., Lu, Z., Huang, J., and Ye, J. (2013), "A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems," in *Proceedings of the 30th International Conference on Machine Learning*, Altanta, USA.

Hallin, M. and Liška, R. (2011), "Dynamic Factors in the Presence of Blocks," *Journal of econometrics*, 163, 29–41.

Hannan, E. J. and Quinn, B. G. (1979), "The determination of the order of an autoregression," *Journal of the Royal Statistical Society: Series B (Methodological)*, 41, 190–195.

Higham, N. (1998), "Matrix nearness problems and applications," in *Applications of matrix theory : based on the proceedings of a conference organized by the Institute of Mathematics and its Applications on Applications of Matrix Theory, held in the University of Bradford in July, 1988*, ed. S. Gover, M. J. C.and Barnett, pp. 1–27, Oxford, Oxford University Press.

Higham, N. J. (1999), "Computing a Nearest Symmetric Positive Semidefinite Matrix," *Linear algebra and its applications*, 103, 103–118.

Higham, N. J. (2002), "Computing the Nearest Correlation Matrix—a Problem from Finance," *IMA Journal of Numerical Analysis*, 22, 329–343.

Hsu, N.-J., Hung, H.-L., and Chang, Y.-M. (2008), "Subset selection for vector autoregressive processes using Lasso," *Computational Statistics and Data Analysis*, 52, 3645–3657.

Hu, F., Lu, Z., Wong, H., and Yuen, T. P. (2016), "Analysis of air quality time series of Hong Kong with graphical modeling: ANALYSIS OF AIR QUALITY TIME SERIES," *Environmetrics*, 27, 169–181.

Lam, C., Yao, Q., and Bathia, N. (2011), "Estimation of latent factors for high-dimensional time series," *Biometrika*, 98, 901–918.

Lauritzen, S. L. (1996), *Graphical Models*, Clarendon Press, Oxford [England].

Lauritzen, S. L. and Wermuth, N. (1989), "Graphical Models for Associations between Variables, some of which are Qualitative and some Quantitative," *The Annals of statistics*, 17, 31–57.

Lee, W. and Liu, Y. (2012), "Simultaneous multiple response regression and inverse covariance matrix estimation via penalized Gaussian maximum likelihood," *J Multivar Anal*, 111, 241–255.

Lütkepohl, H. (2005), *New Introduction to Multiple Time Series Analysis*, Multiple time series analysis, Springer Berlin Heidelberg, Berlin, Heidelberg.

Meinshausen, N. and Bühlmann, P. (2006), "High-dimensional graphs and variable selection with the Lasso," *The Annals of Statistics*, 34, 1436–1462.

Pearl, J., Morgan, M. B., Jowell, S., Genderen, R. V., Pearl, D., and Medoff, L. (1988), *Probabilistic reasoning in intelligent systems : networks of plausible inference*, Morgan Kaufmann Publishers, Inc., San Francisco, California.

Qi, H. and Sun, D. (2006), "A Quadratically Convergent Newton Method for Computing the Nearest Correlation Matrix," *SIAM Journal on Matrix Analysis and Applications*, 28, 360–385.

Ren, Y., Xiao, Z., and Zhang, X. (2013), "Two-step adaptive model selection for vector autoregressive processes," *Journal of Multivariate Analysis*, 116, 349–364.

Rothman, A. J., Levina, E., and Zhu, J. (2010), "Sparse multivariate regression with covariance estimation," *Journal of Computational and Graphical Statistics*, 19, 947–962.

Schwarz, G. (1978), "Estimating the dimension of a model," *The Annals of Statistics*, 6, 461–464.

Sofer, T., Dicker, L., and Lin, X. (2014), "Variable selection for high dimensional multivariate outcomes," *Statistica Sinica*, 24, 1633–1654.

Song, S. and Bickel, P. J. (2011), "Large vector auto regressions," .

Songsiri, J., Dahl, J., and Vandenberghe, L. (2009a), *Graphical models of autoregressive processes*, pp. 89–116, Cambridge University Press.

Songsiri, J., Dahl, J., and Vandenberghe, L. (2009b), "Maximum-likelihood estimation of autoregressive models with conditional independence constraints," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1701–1704.

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

Tsai, H. and Tsay, R. S. (2010), "Constrained Factor Models," *Journal of the American Statistical Association*, 105, 1593–1605.

Tunnicliffe Wilson, G., Reale, M., and Haywood, J. (2015), *Models for Dependent Time Series*, Chapman and Hall/CRC, London, 1 edn.

von Rosen, D. (2018), *Bilinear regression analysis : an introduction*, Springer, Cham, Switzerland, 1 edn.

Wermuth, N. and Lauritzen, S. L. (1990), "On Substantive Research Hypotheses, Conditional Independence Graphs and Graphical Chain Models," *Journal of the Royal Statistical Society. Series B, Methodological*, 52, 21–50.

Whittaker, J. (1990), *Graphical models in applied multivariate statistics*, Wiley, Chichester [England].

Yu, P. and Pong, T. K. (2019), "Iteratively reweighted ℓ_1 algorithms with extrapolation," *Computational Optimization and Applications*, 73, 353–386.

Yuen, T. P., Wong, H., and Yiu, K. F. C. (2018), "On constrained estimation of graphical time series models," *Computational Statistics and Data Analysis*, 124, 27–52.

Zhang, C. H. (2010a), "Nearly unbiased variable selection under minimax concave penalty," *The Annals of Statistics*, 38, 894–942.

Zhang, T. (2010b), "Analysis of multi-stage convex relaxation for sparse regularization," *Journal of Machine Learning Research*, 11, 1081–1107.

Zhao, J. and Leng, C. (2014), "Structured Lasso for Regression with Marix Covariates," *Statistica Sinica*, 24, 799–814.