TOPICS ON THE ROLE OF CHOLESKY FACTOR IN LEARNING HIGH DIMENSIONAL GRAPHICAL MODELS

A Dissertation

by

ARAMAYIS DALLAKYAN

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Mohsen Pourahmadi
Committee Members,	Anirban Bhattacharya
	Francis J. Narcowich
	Xianyang Zhang
Head of Department,	Brani Vidakovic

May 2021

Major Subject: Statistics

ABSTRACT

In modern multivariate statistics learning from “Big Data” is ubiquitous, and understanding relationships and dependencies of variables is imperative to develop learning algorithms. Unequivocally, the (inverse) covariance matrix and Bayesian Networks are the most fundamental objects that specify multivariate associations and dependencies.

Time series literature is rich in methods advocating utilization of the Cholesky factor to model temporal dependence and dynamics in data. Recently, a similar movement is evolving in modern statistical and machine learning literature where the focus is on the estimation of (inverse) covariance matrices and Bayesian Networks. The main contributions in this dissertation pivot around two topics: sparsity and smoothness of the Cholesky factor.

The smoothness of subdiagonals of the Cholesky factor of a large covariance matrix is closely related to the degree of nonstationarity of the autoregressive model for time series and longitudinal data. Heuristically, one expects for a nearly stationary covariance matrix, entries in each subdiagonal of the Cholesky factor of its inverse to be approximately the same, in the sense that the sum of absolute values of successive terms is small or can be bounded. Statistically, such smoothness is achieved by regularizing each subdiagonal using fused-type lasso penalties. In Chapter 2, we rely on the Cholesky factor as the new parameter within a regularized normal likelihood setup which guarantees: (1) joint convexity of the likelihood function, (2) strict convexity of the likelihood function restricted to each subdiagonal even when $n < p$, and (3) positive-definiteness of the estimated covariance matrix. A block coordinate descent algorithm, where each block is a subdiagonal, is proposed, and its convergence is established under mild conditions. Simulation results and real data analysis show the scope and good performance of the proposed methodology.

In Chapter 3, we propose an algorithm to learn Gaussian Bayesian Networks. The impetus of our work is the observation that the Cholesky factor of the inverse covariance matrix entails the structure of a directed acyclic graph (DAGs) when the ordering of variables is known. However, the combinatorial problem of learning the order of variables in DAGs is NP-hard and computationally infeasible for high-dimensional problems. We introduce the permutation matrix as a new parameter within a regularized Gaussian log-likelihood to estimate variable ordering. The proposed algorithm iteratively learns DAGs by optimizing the regularized likelihood function over the set of permutation and lower triangular matrices. First, by relaxation, it finds the permutation matrix, and then for a given ordering estimates a sparse Cholesky factor by decoupling row-wise. The convergence and statistical properties of the algorithm in each step are established under mild conditions. We use our methodology to analyze a macro-economic dataset.

DEDICATION

To my parents, my wife, and my son.

ACKNOWLEDGMENTS

I believe one can easily get lost on the beautiful, and at the same time terrifying, explorations of unknown terrains of Ph.D. if not surrounded by "lighthouses" that guide into a safe harbor. I offer my sincerest thanks and heartfelt gratitude to my adviser Professor Mohsen Pourahmadi for being an outstanding adviser, a dedicated teacher, and an amiable person with an extraordinary sense of humor. I am grateful for the exceptional freedom he allowed, as well as the guidance he provided when I was facing difficulties.

I want to thank Dr. Francis Narcowich for teaching me applied mathematics in the most beautiful and inspiring way I could have ever imagined. His explanation of Lebesgue integral, using the coins analogy, will be stuck with me my entire life. I am indebted to Dr. Anirban Batacharya for introducing me to the statistical learning field. His knowledge and teaching style were the impetus of my decision to change my major from Agricultural Economics to Statistics. I am sincerely thankful to Dr. Xianyang Zhang for teaching me the probability and asymptotic theory.

I have been fortunate to have attended courses taught by Dr. Irina Gaynanova and Dr. Depdeep Pati, some of the best teachers and researchers I have ever seen. I am also thankful to Dr. Jianhua Huang and Dr. Alan Dabney for their encouragement to teach undergraduate statistical classes.

As I look down the memory lane, I feel immensely grateful to Dr. Rafael Bakhtavoryan and Dr. Armen Asatryan at the Agribusiness Teaching Center, without whose support it would be difficult to make it here. I also feel indebted to Dr. David Bessler, my former adviser from the Department of Agricultural Economics, for his immense encouragement to change my major. Without his support, I wouldn't have probably dared to switch to Statistics.

Beyond academics, I feel honored to know Dr. John Nichols, Mrs. Carol Nichols, and their family. They never made us feel lonely and filled our lives with care and joy. Without their help, advice, and support, our days in the USA would have been much difficult.

Finally and most importantly, lots of love for my parents, Hambardzum Dallakyan and Rima Atoyan, who sacrificed lots of pleasures of their lives to see their children achieving newer and

newer heights. Neither this dissertation nor any success I had in the past six years would have been possible without the love, unwavering support, and encouragement from Mane, my best friend, life partner, and wife, who also brought into the world the most precious thing in my life, our six years old son Daniel. I dedicate this to them.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

All work for the dissertation was completed by the student, in collaboration with Professor Mohsen Pourahmadi.

Funding Sources

Graduate study was supported by the Department of Statistics at Texas A&M University.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

1. INTRODUCTION

The Cholesky factorization of a positive definite matrix, named after Andrè-Louis Cholesky, a French military officer involved in geodesy (Brezinski, 2006), has its deep roots in the statistical and numerical analysis literature. In numerical analysis, the widespread use of Cholesky decomposition is to solve a system of equations as opposed to a direct matrix inversion, which is computationally costlier (Golub and Van Loan, 1996). In Statistics, its usage extends from the least squares and generalized least squares problem, auto-regressive models, and Monte-Carlo simulations to the modern statistical learning algorithms. The Bartlett decomposition provides the distribution of the Cholesky factor of the sample covariance matrix (Bartlett, 1933) and Olkin (1985) discusses the bias of the Cholesky factor elements. For the matrix-valued functions, Chern and Dieci (2000) connect the smoothness of the covariance matrix and its Cholesky factor where the smoothness is in terms of the degree of differentiability.

In time series analysis, popular models such as moving average, autoregressive (AR), and ARMA rely on the modified Cholesky factorization of the inverse covariance or precision matrix to model stationary data (Ansley, 1979). The modified Cholesky factorization and Cholesky factorization of the precision matrix are defined $\Theta = T^{'}\Lambda^{-1}T = L^{'}L,$ where $L = \Lambda^{-1/2}T$, and T, L are lower unitriangular and triangular matrices, respectively. In this thesis, a stationary time series is defined as a finite variance process, such that the mean is constant and independent from time, and the autocovariance function depends on time stamps s and t only through their difference $|s-t|$. A time series is ARMA(p, q) if it is stationary and $X_t = \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q w_{t-q}$.

For a time-ordered random vector $X = (X_1, \ldots, X_p)$, regressing a variable on its preceding variables

$$X_t = \sum_{j=1}^{t-1} \phi_{tj} X_{t-j} + \varepsilon_t, \quad t = 1, 2, \ldots, p, \quad \phi_{11} = 0, \tag{1.1}$$

and letting $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_p)'$, the matrix representation of (1.1) can be written as

$$\varepsilon = TX, \tag{1.2}$$

where T is a unit lower triangular matrix with $-\phi_{tj}$ in the (t, j)th position for $2 \leq t \leq p$. Therefore, from (1.1) and (1.2), the stationary AR model of order p is closely related to a p-banded lower triangular matrix where all entries of its first subdiagonal are the same and equal to the negative of the lag-1 AR coefficient, and so on. For nonstationary time series, when data exhibit mild departures from stationarity (Dahlhaus, 1997; Adak, 1998; Davis et al., 2006), the focus has been on (time-)varying AR models (Gabriel, 1962; Rao, 1970; Kitagawa and Gersch, 1985; Dahlhaus, 1997; Zimmerman and Nunez-Anton, 2010) where coefficients of (1.1) are modeled as time-varying, smooth functions. For example, for piecewise stationary processes (Davis et al. (2006)), AR coefficients and corresponding subdiagonals of T could be modeled as certain step functions. To emphasize the time-varying nature of the coefficient ϕ_{tj} in (1.1) for fixed j, one can resort to a doubly indexed triangular array $X_{t,p}$ notation and rewrite (1.1) as

$$X_{t,p} = \sum_{j=1}^{t-1} \phi_j(\frac{t}{p}) X_{t-j,p} + \sigma(\frac{t}{p})\varepsilon_t, \quad t = 1, \ldots, p, \tag{1.3}$$

where $\phi_j(u)$ and $\sigma(u)$ are smooth functions of the rescaled time $u = \frac{t}{p} \in [0, 1]$ and ε's are i.i.d. random variables with mean zero and variance one. Ding and Zhou (2019) showed global approximation of the short-range dependent nonstationary and non-linear time series to an autoregressive process of slowly diverging order, assuming a positive definite covariance matrix.

1.1 Regularized Cholesky Factorization

Inspired by the functional view (1.3), Wu and Pourahmadi (2003) proposed nonparametric estimation procedure for subdiagonals of T. Pourahmadi (1999); Huang et al. (2006) and Levina et al. (2008) advocated the use of modified Cholesky factor T of the precision matrix for parsimony (GLM-based) and sparse (regularized) estimation of its Cholesky factor and hence the precision matrix.

In the linear regression setup, as in (1.1), the seminal lasso paper by Tibshirani (1996) laid the foundation for several sparse estimation techniques. The lasso problem minimizes the following convex objective function

$$\arg\min_{\boldsymbol{\phi}_t} \|X_t - \sum_{j=1}^{t-1} \phi_{tj} X_{t-j}\|_2^2 + \lambda \rho(\boldsymbol{\phi}_t);\ \rho(\boldsymbol{\phi}_t) = \|\boldsymbol{\phi}_t\|_1. \tag{1.4}$$

Here, the tuning parameter λ controls the sparsity level, i.e the larger value yields the sparser coefficient vector $\boldsymbol{\phi}_t = (\phi_{t1}, \ldots, \phi_{tt-1})'$ and vice versa. A rich literature exists on various forms of penalty function $\rho(\cdot)$, which enforces different structured forms on the coefficient vector. The most commonly used ones are fused lasso (Tibshirani et al., 2005) $\rho(\boldsymbol{\phi}_t) = \sum_{i=2}^{t-1} |\phi_{ti} - \phi_{ti-1}|$, and group lasso (Yuan and Lin, 2006) $\rho(\boldsymbol{\phi}_t) = \|\boldsymbol{\phi}_t\|_2$ penalties. Fused lasso applies ℓ_1 penalty to differences between corresponding elements of coefficient matrix and imposes a piecewise constant fit. It is used in settings where coordinates in the true model are closely related to their neighbors (Tibshirani and Taylor, 2011).

Meinshausen and Buhlmann (2006) impose sparsity on elements of inverse covariance matrix Θ by fitting a lasso model to each other variable, using others as predictors. On the other hand, Banerjee et al. (2008); Friedman et al. (2008) exploit the log-likelihood function to estimate sparse precision matrix.

In the longitudinal data setup, where the variables inherit a natural order, with a sample $X_1, \cdots,$ $X_n \sim N_p(0, \Sigma)$ and the sample covariance matrix $S = n^{-1} \sum_{i=1}^{n} X_i X_i'$, its log-likelihood function

$$\ell(\Theta) = \text{tr}(\Theta S) - \log \det\Theta \tag{1.5}$$

is used for penalized likelihood estimation of the Cholesky decomposition (T, Λ) in Huang et al.

(2006). In particular, the authors obtain a sparse T by iteratively minimizing

$$\arg\min_{T,\Lambda} \mathrm{tr}(T^{'}\Lambda^{-1}TS) + \log\det\Lambda + \lambda \sum_{1\leq i<j\leq p} |T_{ij}|, \tag{1.6}$$

with respect to T and Λ. However, the objective function (1.6) is not jointly convex or bi-convex. To ensure convexity, Khare et al. (2019) reparameterize the likelihood in terms of the standard Cholesky factor L rather than the customary (T, Λ)-parametrization and minimize the following objective function with respect to L

$$\arg\min_{L} \mathrm{tr}(L^{'}LS) - 2\log\det L + \lambda \sum_{1\leq i<j\leq p} |L_{ij}| \tag{1.7}$$

While T and L share the same sparsity patterns (since $L = T^{1/2}\Lambda$), the connection between the degree of smoothness of their subdiagonals is a bit more complicated and controlled by the boundedness and smoothness of the diagonal entries of Λ (see Lemma 2).

Both approaches in Huang et al. (2006) and Khare et al. (2019) assume a domain-specific ordering of variables, as in time series, longitudinal, location based or gene studies applications. In the Chapter 2, assuming that the order of variables is known, we incorporate nonstationarity assumption of varying coefficients in (1.3) onto Cholesky factor by smoothing its subdiagonals. The smoothness is achieved through regularizing subdiagonals of the Cholesky factor L of Θ using the family of fused lasso penalties (Tibshirani et al., 2005) as an alternative to their smooth (nonparametric) estimation.

Thus, using a family of fused lasso penalty functions on subdiagonals, we propose a novel *smooth Cholesky (SC) algorithm* to estimate subdiagonals of L and hence the (inverse) covariance matrix via a block coordinate descent algorithm.

1.2 Cholesky Factorization and Bayesian Networks

The SC algorithm relies on the pertinent assumption of domain-specific ordering. In applications where such ordering is not available a possible solution is to include the ordering as an additional parameter in the objective function (van de Geer and Bühlmann, 2013; Aragam and Zhou, 2015). In Chapter 3, we lift the stringent assumption of the domain-specific ordering of variables and focus on learning Bayesian Networks (BNs). Bayesian Networks are a popular class of graphical models, whose structure is represented by a directed acyclic graph (DAG). They are interdisciplinary subjects that have been used in many applications such as economics, etc (Dallakyan, 2020; Bessler and Akleman, 1998), finance (Neil et al., 2005), biology (Needham et al., 2007),etc.

Broad utilization of BNs is to encode probabilistic expert systems, for example, ALARM network (Beinlich et al., 1989), or explicitly express conditional independence assumptions, as in hidden Markov models. However, in many applications, interest relies on learning knowledge from data rather than encoding them. For instance, given observational data generated from a DAG model, the interest is in learning the underlying structure of the DAG.

The paramount challenge in learning DAG structure is that the problem is NP-hard (Chickering et al., 2004), and the space of DAGs is combinatorial and scales super-exponentially with the number of nodes (Robinson, 1977). In recent years, the following main approaches have been evolved to estimate underlying DAG structure from data: Independence-based (also called constraint-based) methods such as the inductive causation (IC) (Pearl, 2009) and PC (Spirtes and Glymour, 1991) algorithms, and score-based methods, which learn DAGs by searching over three different spaces: the DAG space (Heckerman et al., 1995), equivalence classes (Chickering, 2003) and ordering space of variables (Teyssier and Koller, 2005).

In this thesis, we resort to the salient connection of the structural equation model (SEM) and BN to propose a score-based algorithm for learning DAGs. In particular, for a p-dimensional random vector X, whose distribution factorizes according to a BN G:

$$P(X;G) = \prod_{j=1}^{p} P(X_j | X_{\Pi_j^{\mathcal{G}}}), \tag{1.8}$$

where $\Pi_j^{\mathcal{G}}$ is the set of parent nodes of the j-th node and a node j is a parent of its child k if the

DAG G contains a directed edge $j \to k$. The conditional distribution (1.8) can be equivalently represented by the linear SEM:

$$X_j = \sum_{k \in \Pi_j^{\mathcal{G}}} \beta_{jk} X_k + \varepsilon_j, \; j = 1, \dots, p, \tag{1.9}$$

where we assume $\varepsilon_j \sim N(0, \omega_j^2)$ are mutually independent and, as well, independent of variables in the parent set $\{X_k : k \in \Pi_j^{\mathcal{G}}\}$. In a vector form, the precision matrix Θ can be characterized by $\Theta = (I - B)'\Omega^{-1}(I - B)$, where $\Omega = \text{diag}(\omega_1^2, \dots, \omega_p^2)$ and $B = \{\beta_{jk}\}$. A series of recent papers established that under some suitable assumptions on error terms, the unique structure identification of B and the DAG G is possible from the joint distribution $P(X; G)$. For example, see Peters and Bühlmann (2013); Ghoshal and Honorio (2018); Chen et al. (2019), and Peters et al. (2017) for the review.

Ye et al. (2020) proposed a score-based Annealing on Regularized Cholesky Score (ARCS) algorithm to estimate BNs generated from the SEM. To develop their algorithm, Ye et al. (2020) utilize the fact that for each DAG there exist a topological ordering such that the coefficient matrix B in (1.9) is strictly lower traingular; i.e., there exist a permutation matrix P such that (1.9) can be written in a matrix form as $PX = B_\pi PX + P\varepsilon$, where $B_\pi = PBP'$ is a strictly lower triangular matrix obtained by permuting rows and columns of B, respectively. Here, a permutation π of the vertex set $V = \{1, \cdots p\}$ is a topological order for DAG if $\pi(j) < \pi(k)$ whenever $(j, k) \in E$. Thus, the Cholesky factor of the precision matrix

$$\Theta_\pi = L_\pi' L_\pi, \; L_\pi = (I - B_\pi)\Omega_\pi^{-1/2}, \tag{1.10}$$

preserves the DAG structure of B_π; i.e., non-zero elements in L_π correspond to directed edges in the DAG $\mathcal{G}$. Incorporating (1.10) into the Gaussian log-likelihood, Ye et al. (2020) introduced a permutation matrix, combined with the Cholesky factor of the precision matrix, as an additional parameter in the optimization problem, and resort to simulated annealing technique for a topological order estimation.

In the Chapter 3, motivated by the Ye et al. (2020) framework, we propose a score-based two stage method for learning Gaussian BNs by minimizing a regularized negative log-likelihood function

$$\underset{P\in\mathcal{P}_p, L\in\mathcal{L}_p}{\arg\min}\ \frac{1}{2}\mathrm{tr}\Big(PSP^{'}L^{'}L\Big) - \sum_{j=1}^{p}\log L_{jj} + \sum_{1\leq j\leq i\leq p}\rho(|L_{ij}|;\lambda), \tag{1.11}$$

where $\rho(|L_{ij}|;\lambda)$ is a penalty term, $\mathcal{L}_p$ and $\mathcal{P}_p$ are the space of lower triangular and permutation matrices, respectively. In (1.11) we omitted subscript π from the matrix L.

Our proposal has the following distinct features and advantages. First, instead of an expensive search of a permutation matrix P in the non-convex space of permutation matrices, we propose the following relaxation: project P onto the Birkhoff polytope (the convex space of doubly stochastic matrices) and then find the "closest" permutation matrix to the optimal doubly stochastic matrix (See Figure 3.2). The projection step includes a concave regularization term, which pushes the projected doubly stochastic matrix "closer" to the permutation matrix if the penalization parameter is sufficiently large. The proposed relaxation is convex if the number of observations exceeds the number of variables (Lemma 6).

Second, given P, we recover the DAG structure entailed in the Cholesky factor L by decoupling row-wise. We show that the optimization reduces to p decoupled penalized regressions where each iteration has a closed form solution. Moreover, the convergence of iterates to the stationary point is guaranteed.

Third, on the statistical side, our method produces a consistent Cholesky factor estimator for the non-convex score function, assuming that the true permutation matrix is known. To the best of our knowledge, consistency results for the sparse Cholesky factor estimator were established only for the convex problems (Yu and Bien, 2017; Khare et al., 2019).

The rest of the dissertation is organized as follows: Chapters 2, and 3 describe in detail the two proposed methodologies. Simulation studies and real data applications of the two techniques are presented. Finally, in Chapter 4, we conclude with the discussion and some problems for the further research.

2. FUSED-LASSO REGULARIZED CHOLESKY FACTORS OF LARGE NONSTATIONARY COVARIANCE MATRICES OF LONGITUDINAL DATA

2.1 Introduction

A salient feature of stationary time series analysis is its reliance on the Cholesky decomposition to model temporal dependence and dynamics. Important examples include moving average models (Cholesky decomposition of a covariance matrix), autoregressive (AR) models (Cholesky decomposition of an inverse covariance matrix), ARMA models in the time-domain (Ansley, 1979), see Dai and Guo (2004); Rosen and Stoffer (2007) for explicit use of the Cholesky factors in the spectral-domain. For nonstationary time series the focus has been on (time-)varying coefficients AR models (Gabriel, 1962; Rao, 1970; Kitagawa and Gersch, 1985; Dahlhaus, 1997; Zimmerman and Nunez-Anton, 2010; Ding and Zhou, 2019).

Recently, a similar dichotomy is taking roots in the modern multivariate statistics and machine learning where the focus is on either estimation of large covariance or inverse covariance matrices of longitudinal data using Cholesky decomposition. Whereas the entries of a covariance matrix quantifies pairwise or marginal dependence, those of the precision or inverse covariance matrix specifies multivariate relationships among the variables in a p-dimensional random vector $X = (X_1, \ldots, X_p)' \in R^p$ with a positive-definite covariance matrix Σ_p. More precisely, when X follows a Gaussian distribution a zero off-diagonal entry of $\Omega_p = (\Omega_{j,k}) = \Sigma_p^{-1}$ or $\Omega_{j,k} = 0$ implies that X_j and X_k are conditionally independent given all other variables (Whittaker, 1990). When the number of observations n is less than the number of variables p, it is reasonable to impose structure or regularize Ω_p directly in the search for sparsity (Banerjee et al., 2008; Friedman et al., 2008), see Pourahmadi (2013) for an overview.

The use of the modified Cholesky decomposition of Ω_p was advocated in Pourahmadi (1999); Wu and Pourahmadi (2003), Huang et al. (2006) and Levina et al. (2008) for parsimony (GLM-based) and sparse (regularized) estimation of its Cholesky factor and hence the precision matrix.

Recall that the standard and modified Cholesky factors of a positive-definite precision matrix are defined and connected by

$$\Omega_p = L_p' L_p = T_p' \Lambda_p^{-1} T_p, \; L_p = \Lambda_p^{-1/2} T_p, \tag{2.1}$$

where $L_p = (L_{i,j})$ is a unique lower triangular matrix with positive diagonal entries and $T_p = (\phi_{i,j})$ is a unit lower triangular matrix with diagonal entries equal to 1, $\Lambda_p = \text{diag}(\sigma_1^2, \ldots, \sigma_p^2)$ is a diagonal matrix with positive diagonal entries. From now on, whenever there is no confusion in the context, we drop the subscript p from T, L, Σ and Ω.

For time series and longitudinal data the entries in each row of T have the useful interpretation as the regression coefficients and each diagonal entry of Λ as the variance of the residual ε_t of regressing a variable on its preceding variables:

$$X_t = \sum_{j=1}^{t-1} \phi_{tj} X_{t-j} + \varepsilon_t, \;\; t = 1, 2, \ldots, p, \;\; \phi_{11} = 0. \tag{2.2}$$

The genesis of this representation and interpretation of the coefficients for stationary processes can be traced to the rise of finite-parameter AR models in 1920's (Pourahmadi, 2001, Section 1.2); (Ansley, 1979). For example, a stationary AR model of order p_0 is closely related to a p_0-banded lower triangular matrix where all entries of its first subdiagonal are the same and equal to the negative of the lag-1 AR coefficient, and so on. Heuristically, one expects for a nearly stationary (Toeplitz) covariance matrix the entries in each subdiagonal of the Cholesky factor of the inverse covariance matrix to be nearly the same in the sense that sum of absolute values of its successive terms is small or can be bounded using a Baxter-type inequality. In fact, if the underlying process is a stationary AR(∞) with representation $X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \epsilon_t$ and one fits lower order AR(p_0) models, the aforementioned sum over the jth subdiagonal is bounded by

$$\sum_{t=j+2}^{p} |\phi_{tj} - \phi_{t-1j}| \leq 2C(\log p) \sum_{l=j}^{\infty} |\phi_l|,$$

which follows from $|\phi_{tj} - \phi_j|\leq \frac{C}{t} \sum_{l=j}^{\infty}|\phi_l|$ (Inoue and Kasahara, 2006, Theorem 3.3).

Important examples of mild departures from stationarity are locally stationary (Dahlhaus, 1997) and piecewise stationary (Adak, 1998; Davis et al., 2006) processes where in the latter the subdiagonals could be certain step functions. Figure 2.10 illustrates the potential adverse effect of learning a genuinely nonstationary covariance matrix of the cattle data (Kenward, 1987) using a (misspecified) stationary AR model.

We emphasize the time-varying nature of the coefficients ϕ_{tj} in (2.2) for fixed j using a doubly indexed triangular array $X_{t,p}$ (Dahlhaus, 1997) and rewriting our data generating model as

$$X_{t,p} = \sum_{j=1}^{p_0} \phi_j(\frac{t}{p})X_{t-j,p} + \sigma(\frac{t}{p})\varepsilon_t, \;\; t = 1, \dots, p, \tag{2.3}$$

where $\phi_j(u)$ and $\sigma(u)$ are smooth functions of the rescaled time $u = \frac{t}{p} \in [0, 1]$ and ε's are i.i.d. random variables with mean zero and variance one. For $p_0 < p$, this rescaling enables one to view the (sub)diagonals of T and Λ as realizations of smooth functions (see Figure 2.1) which brings the estimation problem within the familiar nonparametric infill asymptotic setup where one observes the smooth functions $\phi_j(u)$ and $\sigma(u)$ on a finer grids for a larger p. Interestingly, choosing $\phi_j(u)$ and $\sigma(u)$ as functions of bounded variation guarantees that, under mild conditions, the solutions of (2.3) are locally stationary processes (Dahlhaus and Polonik, 2009, Proposition 2.4).

The functional view of (2.3) for longitudinal data has been a major source of inspiration for nonparametric estimation of the subdiagonals of T, see Wu and Pourahmadi (2003) and Huang et al. (2007). Furthermore, within the smoothing spline ANOVA framework, Blake (2018) treats the AR coefficients $\phi_{tj}, t > j$ as a bivariate smooth function and decomposes it in the stationary direction of the lag $\ell = t - j$ and the nonstationary (additive) direction $m = \frac{t+j}{2}$ and a possible interaction term. Finally, she regularizes the nonstationary direction more heavily which amounts to shrinking the covariance estimator toward the more parsimonious and desirable stationary structures.

We focus on the longitudinal data (replicated time series) setup, with a sample $X_1, \cdots, X_n \sim$

$N_p(0, \Sigma)$ and the sample covariance matrix $S = n^{-1} \sum_{i=1}^{n} X_i X_i'$. The corresponding log-likelihood function $\ell(\Omega) = \text{tr}(\Omega S) - \log\det(\Omega)$ was used for penalized likelihood estimation of the parameters (T, Λ) in Huang et al. (2006), see also Levina et al. (2008) and Khare et al. (2019) for a comprehensive review. The lack of convexity of the likelihood in (T, Λ) was noted first in Khare et al. (2019) and Yu and Bien (2017). They ensure convexity by reparameterizing the likelihood in terms of the standard Cholesky factor L rather than the customary (T, Λ)-parametrization. While the last identity in (2.1) reveals that T and L share the same sparsity patterns, the connection between the degree of smoothness of their subdiagonals is a bit more complicated and controlled by the boundedness of the diagonal entries of Λ (see Theorem 2).

In this paper we achieve smoothness through regularizing the subdiagonals of the Cholesky factor L of Ω using the fused lasso penalties (Tibshirani et al., 2005) as an alternative to smooth (nonparametric) estimation of subdiagonals. More specifically, using the family of fused lasso penalty functions on the subdiagonals we propose a novel *smooth Cholesky (SC) algorithm* for estimating the subdiagonals of L and hence the (inverse) covariance matrix via a block coordinate decent algorithm. The SC objective function is convex in L, and compared to the recent algorithms in Khare et al. (2019) and Yu and Bien (2017) when $n << p$, the update of each block is obtained by solving a strictly convex optimization problem. We establish the convergence of the iterates to stationary points of the objective function, and elaborate on the connection between the smoothness of the subdiagonals of L and those of T under the assumption of boundedness of the diagonal entries of Λ.

Our SC algorithm and the corresponding methodology for longitudinal data can be specialized to the setup of a long stretch of a single stationary time series, namely for $n = 1$ and p large. To this end, banded estimates of Toeplitz covariance matrices and properties of the corresponding optimal linear predictors are studied in Wu and Pourahmadi (2009); Bickel and Gel (2011) and McMurry and Politis (2010, 2015). For covariance estimation and prediction of locally stationary processes, see Das and Politis (2020).

In the rest of this section, we introduce notation used throughout the paper. For a vector $x =$

$(x_1, \ldots, x_p) \in \mathcal{R}^p$, we define its norm $\|x\|_q = (\sum_{i=1}^q |x_i|^q)^{1/q}$ for $q \geq 1$. We denote by $\mathcal{L}_p$ the space of all lower triangular matrices with positive diagonal elements. Given a $p \times p$ lower-triangular matrix L, the $p^2 \times 1$ vector $V = (v_i) = vec(L)$ is its standard vectorization formed by stacking up its column vectors including the zero (redundant) entries. Each vector of (sub)diagonal entries of L corresponds to those from V with the following set of indices:

$$I_j = \{k(p+1) + j + 1 : k = 0, \ldots, (p - j - 1)\}, j = 0, 1, \ldots, p - 1,$$

so that I_0 corresponds to the main diagonal entries, $L^{[j]} = V_{I_j} = (v_i)_{i \in I_j}$ is the $|I_j|$-subvector of the jth subdiagonal entries. We denote by $L^{-[j]} = (v_i)_{\{i \in I_k, k \neq j\}}$ a vector of diagonal and subdiagonals, except for the jth subdiagonal. For simplicity in notation, we replace I_j by j so that for a given $p^2 \times p^2$ matrix A and index sets I_j, I_k, $A_{\cdot j}$ denotes the $p^2 \times |I_j|$ submatrix with column indices selected from I_j, and A_{jk} is the $|I_j| \times |I_k|$ submatrix with rows and columns of A indexed by I_j and I_k, respectively.

2.2 The Smooth Cholesky Algorithm

In this section, we develop the SC algorithm for a convex penalized likelihood function using fused-type Lasso penalties on the subdiagonals of the standard Cholesky factor. Such penalties are bound to induce various degrees of sparsity and smoothness on the subdiagonals, but our main focus is on smoothness. The objective functions turn out to be conditionally separable. Computational and statistical properties of a block coordinate descent algorithm for its minimization are studied.

2.2.1 The Gaussian-Likelihood and Fused Lasso Penalties

Let $\ell(\Omega)$ be the Gaussian log-likelihood function for a sample of size n from a zero-mean normal distribution with the precision matrix Ω. Its convexity is ensured by reparametrizing it in terms of the standard Cholesky factor L, see Khare et al. (2019) and Yu and Bien (2017). More precisely, we consider

$$Q(L) = tr(L^{'}LS) - 2\log\det(L) + \lambda P(L), \tag{2.4}$$

where $P(L)$ is a convex penalty function and S is a sample covariance matrix.. There are two recent important choices of $P(L)$ designed to induce sparsity in the rows of the Cholesky factor.

The method of Convex Sparse Cholesky Selection (CSCS) of L in Khare et al. (2019) employs the penalty $P(L) = \|L\|_1$. The ensuing objective function turns out to be jointly convex in the (nonredundant) entries of L, bounded away from $-\infty$ even if $n < p$; but it is not strictly convex in the high-dimensional case. A cyclic coordinatewise minimization algorithm is developed in Khare et al. (2019) to compute L. Note that once L is computed using the CSCS or other methods considered here, then one can compute (T, Λ), and the (inverse) covariance matrix Σ and Ω. Sparsity of Ω is not guaranteed since the sparsity pattern of the estimated L in Khare et al. (2019), as in Huang et al. (2006) and Shojaie and Michailidis (2010), has no particular structure. Fortunately, a more structured sparse L which guarantees sparsity of the precision matrix is developed in Yu and Bien (2017). Their hierarchical sparse Cholesky (HSC) method relies on the hierarchical group penalty $P(L) = \sum_{r=2}^{p}\sum_{l=1}^{r-1}(\sum_{m=1}^{l} w_{lm}^2 L_{rm}^2)^{1/2}$ where the w_{lm}'s are quadratically decaying weights. The HSC method has the goal of learning the local dependence among the variables and leads to a more structured sparsity with a contiguous stretch of zeros in each row away from the main diagonal. Its flexibility is similar to that of the nested lasso in Rothman et al. (2010). Yu and Bien (2017) relies on an alternating direction method of multipliers (ADMM) approach to compute L. Computationally, both penalty functions lead to a decoupling of the above objective function into p separate and parallelizable optimization problems each involving a separate row of L.

For the SC algorithm developed in this paper, we employ a number of *fused lasso* penalty functions on the Cholesky factor or its subdiagonals. However, unless stated otherwise the phrase *fused lasso* refers to

$$P(L)=\sum_{i=1}^{p-1}P_{\nabla}(L^{[i]}),\;\; P_{\nabla}(L^{[i]})=\sum_{j=2}^{p-i}|L_j^{[i]}-L_{j-1}^{[i]}|,\;\; L^{[i]}\in\mathcal{R}^{p-i}\text{ and } i=1,\dots,p-1,$$

based on the ℓ_1-norm of the first differences, and we do not penalize diagonal elements. In Appendix A.5 we give an illustrative example of the penalty form. Note that this penalty is slightly different from the more general *sparse fused lasso* penalty function in Tibshirani et al. (2005) and Tibshirani and Taylor (2011) which is of the form

$$\lambda_1\sum_{j=1}^{p-i}|L_j^{[i]}|+\lambda_2 P_{\nabla}(L^{[i]}).$$

The latter includes an additional lasso penalty term to achieve sparsity on top of smoothness of the subdiagonals. In fact, our usage of *fused lasso* is more in the spirit of the total variation penalty in Rudin et al. (1992).

When higher-order smoothness of the subdiagonals is desirable, then it is natural to penalize sum of higher-order differences such as $\|D_2y\|_1$, the ℓ_1-trend filtering (Kim et al., 2009), and $\|D_2y\|_2^2$ (Hodrick and Prescott, 1997), referred to as H-P hereafter, where D_2 is the matrix of second-order differences. For other higher order difference matrices belonging to the family of generalized lasso penalties, see Tibshirani et al. (2005);Tibshirani and Taylor (2011).

2.2.2 The Conditionally Separable Convex Objective Function

We express the objective function (2.4) as the sum of p quadratic functions each involving distinct (sub)diagonals of L (given the others), so that it is conditionally separable in the sense made precise in Lemma 1(b). This is in sharp contrast to the objective functions in Khare et al. (2019) and Yu and Bien (2017) which decouple over the rows of the matrix L with nice computational consequences. Nevertheless, our objective function is jointly convex in L, in general, and strictly convex when $n < p$.

Let $B = S \otimes I_p$ be the Kronecker product of the sample covariance matrix from a sample of size n and the identity matrix. The structure of the matrix B and the $(p-i) \times (p-j)$ submatrices B_{ij}, $0 \leq i, j \leq p-1$, introduced in the proof of the following lemma play a vital role in proving properties of our SC algorithm.

Lemma 1. *For the lower triangular matrix L it holds that:*

(a) The first term in (2.4) can be rewritten as

$$tr(LSL') = V'(S \otimes I_p)V = \sum_{i=0}^{p-1} \sum_{j=0}^{p-1} L^{[i]} B_{ij} L^{[j]} \tag{2.5}$$

(b) The objective function $Q(L)$ is conditionally separable in that

$$Q(L) = \sum_{i=0}^{p-1} Q_i(L^{[i]} | L^{-[i]}), \tag{2.6}$$

where for $i = 0, 1, \ldots, p-1$ and fixed $L^{-[i]}$,

$$Q_i(L^{[i]} | L^{-[i]}) = q_i(L^{[i]} | L^{-[i]}) + \lambda P_{\nabla}(L^{[i]}),\ Q_0(L^{[0]} | L^{-[0]}) = q_0(L^{[0]} | L^{-[0]}) - 2 \sum_{j=1}^{p} \log L_j^{[0]} \tag{2.7}$$

and

$$q_i(L^{[i]} | L^{-[i]}) = (L^{[i]})' B_{ii} L^{[i]} + (L^{[i]})' (\sum_{j \neq i} B_{ij} L^{[j]}), \tag{2.8}$$

(c) $Q_i(\cdot)$'s are strictly convex in $L^{[i]}$ even when $n < p$.

A proof of the lemma is provided in the Appendix. Parts (a) and (b) are fundamental for constructing our SC algorithm in the spirit of the coordinate descent algorithm in Khare et al. (2019, Lemma 2.3). However, since our objective function is not separable over the subdiagonals, the details of the proof of our block coordinate descend algorithm differ considerably from those in Khare et al. (2019).

2.2.3 A Block Coordinate Descent Algorithm

In this section, relying on the conditional separability as expressed in (2.6) we minimize $Q(L)$ using a block coordinate descent algorithm where each block corresponds to a subdiagonal of L given the values of the others. The minimization of $Q(L)$ is done *sequentially* over the summands $Q_i(\cdot)$, $0 \leq i \leq p-1$. In this sense, our SC algorithm is different from the recent approaches in covariance estimation where the objective functions are either minimized by iterating over the columns of a covariance matrix (Banerjee et al., 2008; Friedman et al., 2008) or the rows of its Cholesky factor (Khare et al., 2019; Yu and Bien, 2017). However, it inherits some of the desirable convergence properties of the latter two algorithms even though their optimization problems decouples into p parallel problems over the rows of the matrix L.

The following two generic functions stand for the objective function restricted to each (sub)diagonal:

$$h_0(x|y_0) = 2x'y_0 + x'C_0x - 2\sum_{j=1}^{p-1} \log x_j \tag{2.9}$$

and

$$h_i(x|y_i) = 2x'y_i + x'C_ix + \lambda\|Dx\|_1, \tag{2.10}$$

where $C_i = B_{ii}$ is a diagonal matrix introduced in Lemma 1, and $y_i = \sum_{j\neq i} B_{ij}L^{[j]},\ 0 \leq i \leq p-1$ is a $(p-i)\times 1$ vector. Note that the function h_0 is from R_+^p to R and h_i is from R^{p-i} to R for $1 \leq i \leq p-1$. These functions are simpler than those in Khare et al. (2019, equation (2.8)) since the matrices C_i are diagonal with positive diagonal entries so that for a fixed vector y_i, h_i's are strictly convex functions (Lemma 6). We note that a block coordinate descent algorithm which sequentially optimizes h_i with respect to each $L^{[i]}$ will also optimize the objective function $Q(L)$.

Consider the global minimizers of h_0 and h_i:

$$x_0^* = \arg\min_{x\in\mathcal{R}_+^p} h_0(x|y_0) \text{ and } x_i^* = \arg\min_{x\in\mathcal{R}^{p-i}} h_i(x|y_i). \tag{2.11}$$

Next, we show that the vector x_0^* has a closed-form and provide methods to compute $\{x_i^*\}_{i=1}^{p-1}$

for various members of the fused-type Lasso family. A proof of the lemma is provided in the Appendix.

Lemma 2. *(a) For a given y_0, x_0^* is unique and its entries have the closed-form:*

$$(x_0^*)_1 = 1/\sqrt{(C_0)_{1,1}}, \ \textit{for} \ \ i = 2, \ldots, p, \quad (x_0^*)_i = \frac{-(y_0)_i + \sqrt{(y_0)_i^2 + 4(C_0)_{i,i}}}{2(C_0)_{i,i}}. \tag{2.12}$$

(b) For a given y_i ($1 \leq i \leq p-1$), x_i^* corresponds to the unique solution of the fused lasso problem (Tibshirani and Taylor, 2011, Algorithm 1) for the ith subdiagonal of L.

(c) When D in (2.10) is the matrix of second-order differences, then

(1) x_i^* corresponds to the solution of the ℓ_1-trend filtering (Kim et al., 2009, Section 6).

(2) For $h_i(x|y_i) = 2x'y_i + x'C_ix + \lambda\|Dx\|_2^2$, ($1 \leq i \leq p-1$), x_i^* has a closed form and corresponds to the H-P solution:

$$x_i^* = -\frac{1}{2}(C_i + \lambda(D'D))^{-1}y_i$$

(d) For $\lambda_1 > 0$, the solution of sparse fused lasso,

$$\underset{x \in \mathcal{R}^{p-i}}{\arg\min}\, \tilde{h}_i(x|y) = h_i(x|y) + \lambda_1\|x\|_1, \ 1 \leq i \leq p-1 \tag{2.13}$$

is given by

$$\hat{x}_i(\lambda_1, \lambda_2) = \textit{sign}(\hat{x}_i(0, \lambda_2))(|\hat{x}_i(0, \lambda_2)| - \frac{1}{2}(\textit{diag}(C_i^{-1}))\lambda_1)_+,$$

where $\hat{x}_i(0, \lambda_2)$ is the solution of (2.13) when $\lambda_1 = 0$ and $\lambda_2 \geq 0$.

It is well-known that the tuning parameter λ controls the bias-variance trade-off of the ℓ_1 penalization (Hastie et al., 2001), i.e bias increases as λ increases and vice-versa. Fan and Li (2001) and Zhang (2010) attack this issue by introducing SCAD and MCP penalties to produce nearly unbiased estimates for large coefficients.

Even though the sample covariance matrix is an unbiased estimate of the covariance matrix, its Cholesky factor is biased for its population counterpart (Olkin, 1985). Moreover, for $p < n-1$, the inverse sample covariance matrix is a biased estimate of the inverse covariance matrix (Anderson, 2003). To empirically capture the bias and variance of our estimator, we estimate 1-banded true Cholesky factor generated from the first subdiagonal of the Case B described in Section 2.4 and illustrated in Figure 2.2. Here, the goal is to estimate bias for the three different values of the step function using fused lasso penalty. As illustrated in Figure A.5 in the Appendix, for jumps with the large magnitude bias is large in absolute value and variance is relatively flat with small bumps at jump points.

Lemma 2 provides the necessary ingredients for minimizing the objective function (2.6) via the following block coordinate descent algorithm where each block is a (sub)diagonal of the standard Cholesky factor L.

Algorithm 1 The SC algorithm

```
input:
$\epsilon, \lambda, k_{max} \leftarrow$ Stopping criteria, Tuning Parameter, and max. number of iteration
$L^{(0)} \leftarrow$ Initial Cholesky factor
Set $B \leftarrow S \otimes I_p;\ C_i \leftarrow B_{ii}$
while $\|L^{(k+1)} - L^{(k)}\|_\infty > \epsilon$ or $k < k_{max}$:
    $L^{(k)} \leftarrow L^{(0)}$
    for $i = 0, \ldots, p-1$ do:
        $\hat{L}^{[i]} = \arg\min h_i(L^{[i]}|y_i)$
        Update $L^{(k)}$ by replacing the $i$th subdiagonal by $\hat{L}^{[i]}$
    $L^{(0)} \leftarrow L^{(k)};\ k = k+1$
Output: $L$
```

We note that the Algorithm 1 is well-defined so long as the diagonal entries of sample covariance matrix and the initial Cholesky factor are positive. That is the minimum in the optimization appearing in line 6 of the algorithm is attained. This follows from Part (b) of Theorem 1 and the fact that h_i's are strictly convex functions of $L^{[i]},\ 0 \leq i \leq p-1$. There exists rich literature on analyzing properties of block coordinate descent. It is known (Beck and Tetruashvili, 2013) that, under suitable conditions, the block coordinate descent achieves sublinear rate of convergence.

2.2.4 Convergence of the SC Algorithm

In this section, we establish convergence of the SC algorithm under the weak restriction that the diagonal entries of S are positive.

A key step is to reduce the objective function (2.6) to the following widely used objective function in the statistics and machine learning communities (Khare and Rajaratnam, 2014):

$$h(x) = x'E'Ex - \sum_{i \in C^c} \log x_i + \lambda \sum_{i \in C} |x_i| \tag{2.14}$$

where $\lambda > 0$ is a tuning parameter, C is a given subset of indices and the matrix E does not have a zero column. Since the objective function restricted to each subdiagonal (line 6 in Algorithm 1) is strictly convex, a unique global minimum with respect to each subdiagonal is guaranteed even when $n < p$. This additional strict convexity property along with Theorems 2.1 and 2.2 in Khare and Rajaratnam (2014) are the key ingredients for showing that the iterates in SC algorithm converge to the global minimum of the objective function Q.

Theorem 1. *(a) The objective function $Q(L)$ with the fused Lasso penalty admits the generic form:*

$$h(x) = x'E'Ex - \sum_{i=1}^{p} \log x_i + \lambda \sum_{j \in C} |x_i|, \tag{2.15}$$

where,

$$x = [L_{1,1}, \ldots, L_{p,p}, L_{3,2} - L_{2,1}, \ldots, L_{p,p-1} - L_{p-1,p-2}, \ldots, L_{p,2} - L_{p-1,1}, L_{p,1}]',$$

and the set C of indices consists of the last element of x and along with those of difference forms, and E is a suitable matrix with no 0 columns.

(b) If $diag(S) > 0$, then the sequence of iterates $\{L^{(k)}\}$ in Algorithm 1 converges to a global minimum of Q.

Proof of the theorem given in the Appendix relies on the following:

Lemma 3. *For every n and p*

$$\inf_{L \in \mathcal{L}_p} Q(L) \geq -\mathbf{1}_p' K \mathbf{1}_p > -\infty,$$

where $\mathbf{1}_p$ is a $p \times 1$ vector of 1's and K is a positive semi-definite matrix. Moreover, any global minimizer of $Q(L)$ over the open set $\mathcal{L}_p$ lies in $\mathcal{L}_p$.

A discussion of convergence of the sequence of iterates for ℓ_1-trend filtering and HP is provided in the Appendix A.1.

2.2.5 Computational Complexity of the SC Algorithm

The sequential SC algorithm in each iteration sweeps over the diagonal and subdiagonals of L where in each sweep it must compute y_i and h_i. For example, for fused lasso penalty, from Lemma 6, updating each subdiagonal requires solving a fused lasso problem. Therefore, the computational cost of each subdiagonal update depends on the chosen penalty function. Denoting by R_p the computational cost for the chosen penalty to minimize $h_i,\ 1 \leq i \leq p-1$, the next lemma provides the computational cost for each iteration of SC algorithm.

Lemma 4. *The computational cost of Algorithm 1 in each iteration is $min(O(np^2 + pR_p), O(p^3 + pR_p))$.*

The proof is provided in Appendix A.1. For example, $R_p = O(p)$ for x_i^* for the ℓ_1-trend filtering penalty (Kim et al., 2009). Thus, the computational cost of the SC algorithm is $min(O(np^2), O(p^3))$ which is comparable to the cost of the existing sequential algorithms such as GLasso (Friedman et al., 2008), SPACE (Peng et al., 2009) and CONCORD (Khare et al., 2015) and CSCS (Khare et al., 2019) when iterations have been run sequentially.

2.3 Smoothness of L, T and Local Stationarity

A promising feature of using fused lasso penalty and the ensuing SC algorithm seems to be its ability to capture aspects of the smoothness of subdiagonals of the Cholesky factors through regularized likelihood estimation rather than the traditional (non)parameteric methods. In this section, we discuss some details on AR data generating process, and explore the connection between smoothness of T, L, and Ω when the diagonal elements of Λ are bounded away from zero.

Smoothness of the subdiagonals of L, T can be studied by considering a doubly indexed sequence $X_{t,p}$ (triangular arrays), and functions defined on the rescaled time $u \in [0, 1]$. For example, Figure 2.1 provides a simple illustration of the correspondence between the time-varying AR(1) model in (2.3) and the subdiagonals of T.

$$
\begin{aligned}
X_{1,p} &= \phi_1\Big(\frac{1}{p}\Big) X_{0,p} + \sigma\Big(\frac{1}{p}\Big)\epsilon_1 \\
&\vdots \\
X_{p,p} &= \phi_1\Big(\frac{p}{p}\Big) X_{p-1,p} + \sigma\Big(\frac{p}{p}\Big)\epsilon_p
\end{aligned}
\qquad
T = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -\phi_1\big(\frac{1}{p}\big) & 1 & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & -\phi_1(\frac{p}{p}) & 1 \end{bmatrix}
$$

Figure 2.1: Depiction of a Time-Varying AR(1) and the matrix T

Motivated by the connection between the coefficients of the time-varying AR (2.3) and the subdiagonals of the Cholesky factor, it is of interest to connect the smoothness of the entries of the ith subdiagonal of the Cholesky factors L, T, the diagonal entries of Λ and the inverse covariance matrix viewed as functions of the rescaled time u by writing: $L^{[i]}(\cdot) : [0,1] \to R$ where $L^{[i]}(u) = L^{[i]}(j/p) = L^{[i]}_{up}$ for $j = i, \ldots, p-1$ and $L^{[i]}(u) = L^{[i]}_{\lfloor up \rfloor}$ otherwise. Here $\lfloor x \rfloor$ is the largest integer smaller or equal to x and $L^{[i]}_{\lfloor up \rfloor}$ stands for its $\lfloor up \rfloor$th element. In this section, smoothness of a function refers to it being of bounded total variation (TV):

$$TV(L^{[i]}) = \sup\Big\{ \sum_{j=1}^{l} |L^{[i]}(x_j) - L^{[i]}(x_{j-1})|: 0 \leq x_0 < \cdot < x_l \leq 1 \Big\}, \tag{2.16}$$

for x_i's of the form $\frac{i}{p}$ and $TV(L^{[i]}) < \infty$, $i = 0, \ldots, p-1$. It is instructive to note that (Chern and Dieci, 2000, Lemma 2.8) smoothness of a covariance (positive-definite matrix-valued) function is also inherited by its unique standard Cholesky factor when smoothness is in terms of degree of differentiability.

Next, we establish the connections among the subdiagonals of the standard Cholesky factor and related matrices so far as they being of bounded variation is concerned.

Theorem 2. *(a) For any $u, v \in [0,1]$ of the form t/p, we have*

$$|L^{[i]}(u) - L^{[i]}(v)| \leq c^{-1}|T^{[i]}(u) - T^{[i]}(v)| + c^{-2}|T^{[i]}(u)||\sigma(u) - \sigma(v)|,$$

provided that $\sigma(i) > c > 0, \;\; i = 1, \ldots, p$.

(b) If in addition, $\sigma(\cdot)$ and the ith subdiagonal $T^{[i]}(\cdot)$ are functions of bounded total variation on the rescaled interval $[0,1]$ with $TV(T^{[i]}) \leq K_1$, $TV(\sigma) \leq K_2$, and $\|T^{[i]}\|_\infty < m$, then $L^{[i]}$ is of bounded total variation with

$$TV(L^{[i]}) \leq c^{-1}K_1 + c^{-2}K_2 m. \tag{2.17}$$

(c) If the (sub)diagonals of the Cholesky factor L are of bounded variation on the rescaled interval $[0,1]$ with $TV(L^{[i]}) \leq K_i, \;\; \|L^{[i]}\|_\infty \leq m_i$ $(0 \leq i \leq p-1)$, then the (sub)diagonals of the

matrix $\Omega = L'L$ are of bounded variation with

$$TV(\Omega^{[i]}) \leq \sum_{j=0}^{p-i-1} (m_j K_{j+i} + m_{j+i} K_j).$$

Moreover, the converse of (c) is true.

The proof is provided in the Appendix A.1. Theorem 2(a) is a motivation for penalizing L in the reparametrized log-likelihood (2.4) rather than the traditional modified Cholesky factor T.

The appearance of the bounded variation property on σ, $T^{[i]}$ and other matrices opens up a window to connect and approximate nonstationary processes by the class of time-varying AR models. For locally stationary processes with a time-varying MA(∞)-representation see Dahlhaus (1997); Dahlhaus and Polonik (2009); Dahlhaus (2012). Recently, Ding and Zhou (2019) has considered a different class of nonlinear, nonstationary processes which can be approximated well by AR processes of increasing orders. It includes the linear process

$$X_{t,p} = \sum_{j=1}^{\infty} a_{j,p}(t) \epsilon_{t-j},\ t = 1, 2, \ldots, p,$$

where ϵ_k's are i.i.d random variables and

$$\sup_{t \in [0,1]} |a_{j,p}(t)|^2 \leq C a^j,\ j \geq 1, \text{ and } 0 < a < 1. \tag{2.18}$$

This geometrically decaying bound on the time-varying coefficients implies that the series has short-memory when temporal dependence is assessed using the physical dependence measure (Ding and Zhou, 2018, Example 2.4). Moreover, they establish that the coefficients ϕ_{tj} of the increasing order AR approximants can be bounded by

$$|\phi_{tj}| \leq C \max\{p^{-2}, a^{j/2}\} \tag{2.19}$$

under Assumption 2.3 in (Ding and Zhou, 2019, Theorems 2.5 and 3.6, Remark 2.8), for some constant C. Consequently, the sum of absolute differences over the jth subdiagonal of the Cholesky factor of the inverse covariance matrix is controlled by

$$\sum_{t=j+2}^{p} |\phi_{tj} - \phi_{t-1j}| \leq 2C \max\{p^{-1}, pa^{j/2}\}. \tag{2.20}$$

These bounds within the rather general nonlinear and nonstationary setup of Ding and Zhou (2019) reveal that, under reasonable conditions, one can apply our SC algorithm and the related methodology to estimate/approximate the underlying stochastic structure.

2.4 Simulation and Data Analysis

In this section, we illustrate and gauge the performance of our methodology using simulated and real datasets. We use three commonly used penalty functions: fused lasso, ℓ_1-trend filtering and Hodrick-Prescott (H-P) filtering (Hodrick and Prescott, 1997). The corresponding SC algorithm is referred to as SC-Fused, SC-Trend and SC- HP, respectively.

2.4.1 The Simulation Setup: Four Cases of T

In all simulations, the sample sizes are $n = 50, 100$, and dimensions $p = 50, 150$, covering settings where $p < n$ and $p > n$, respectively. Each simulated dataset is centered to zero and scaled to unit variance. The tuning parameter λ is chosen from the range $[0.1, 1]$ over 100 equally spaced grid points using the BIC and CV criterion described in the Appendix A.2. We repeat the simulation 20 times. As inputs to the Algorithm 1, we set the tolerance $\epsilon = 10^{-4}$ and the initial Cholesky factor is the diagonal matrix with diagonal elements equal to $\sqrt{diag(S)}$.

We start with a pair (Λ, T) and use the parameterization $L = \Lambda^{-1/2}T$ as in (Khare et al., 2019) where Λ is a diagonal matrix and T is a unit lower-triangular matrix constructed for the four cases A-D described below. For given pairs $(n, p), (T, \Lambda)$, sample data are drawn independently from $N_p(0, (L'L)^{-1})$. In each case, except for the Case B, where the number of nonzero subdiagonals is equal 2, the number of non-zero subdiagonals is restricted to be 5, that is in each iteration the SC algorithm sweeps only over the first 5 subdiagonals and the rest of subdiagonals are set to 0. Except for the Cases A and B, construction of the matrix T starts with generating its first subdiagonal, and then filling the rest of its subdiagonals by eliminating the last element of the previous subdiagonal. The diagonal elements of $\Lambda^{1/2}$, for the Cases A and B are equal one and are of the form $\log((1:p)/10+2)$ for the Cases C and D.

The four cases of T with varying degrees of smoothness (nonstationarity) of their subdiagonals and the diagonal matrix Λ considered are:

Case A: A stationary AR(1) model where T is a Toeplitz matrix with the value for the first subdiagonal randomly chosen from the uniform distribution on $[-0.7, 0.7]$.

Case B: Resembles an AR(2) model as in Davis et al. (2006, Section 4,1) dealing with piecewise stationary processes:

$$X_t = \begin{cases} -0.7X_{t-1} + \epsilon_t & 1 \leq t \leq p/2 \\ 0.4X_{t-1} - 0.81X_{t-2} + \epsilon_t & p/2 < t \leq 3p/4 \\ -0.3X_{t-1} - 0.81X_{t-2} + \epsilon_t & 3p/4 < t \leq p \end{cases},$$

where $\epsilon_t \sim N(0,1)$. The matrix T here is 2-banded and the diagonal elements of $\Lambda^{1/2}$ are equal to 1 (See Figure 2.2).

Case C: The first subdiagonal of T is given by $T_i^{[1]} = 2(i/p)^2 - 0.5$, $i = 1, \ldots, p-1$, corresponding to a (time) varying-coefficient AR model (Wu and Pourahmadi, 2003).

Case D: The first subdiagonal of T is generated according to

$$T_i^{[1]} = x_i + z_i,\ i = 1, \ldots, p-1,\ \ x_{i+1} = x_i + v_i,\ i = 1, \ldots, p-2,$$

with $x_1 = 0, z_i \sim N(0,1)$ and v_t is a simple Markov process (Kim et al., 2009, Section 4). That is with probability m, $v_{i+1} = v_i$ and with probability $1-m$ it is chosen from the uniform distribution $[-b, b]$ where $m = 0.8, b = 0.5$.

Figure 2.2 illustrates plots of the first subdiagonal of the matrix T versus the rescaled time in $[0,1]$

for the four cases with $p = 50$.

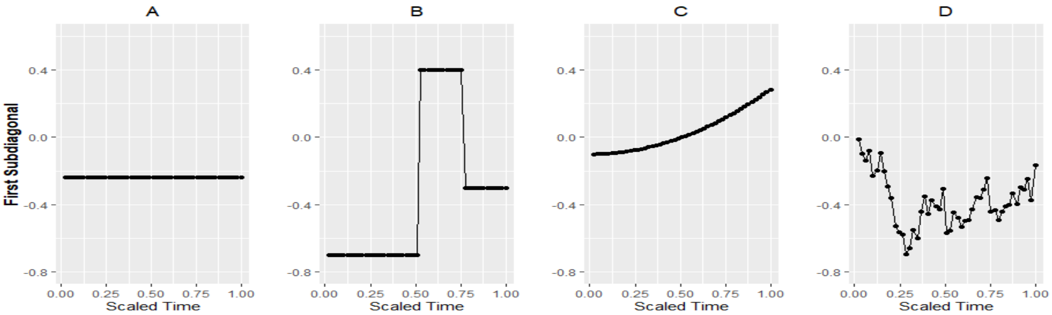


Figure 2.2: Cases A-D, plots of the first subdiagonal of T vs rescaled time ($p = 50$).

2.4.2 Capturing Smoothness: A Graphical Comparison

First, we assess graphically the ability of our methodology to learn the varying degrees of smoothness of the first subdiagonal for the four cases introduced above. Figures 2.3 and 2.4 illustrate the simulation results using the SC algorithm for $p = 50$ and 150, respectively. In each 2 by 4 layout, each column corresponds to one of the four cases and the row to the criteria (BIC or CV) for choosing the tuning parameters. The results for $n = 50$ and $n = 100$ were similar, therefore we report only those for the larger sample size.

The simulation results in both figures provide ample evidence on the good performance of the SC method for estimating time-varying subdiagonals. In particular, for the Case A, as expected, the SC-Fused learns perfectly the flatness (stationarity) of the first subdiagonal, showing only some wiggliness for the BIC. For the Case B, which corresponds to a piecewise stationary process, estimators tuned using CV and BIC correctly identify the jumps and show small oscillation around the flat segments. The CV criterion shows an advantage over the BIC for the Case C. More specifically, the SC-Trend learns better the quadratic structure of the first subdiagonal than the other estimators. For the case D the SC-Trend and SC-HP provide nearly identical estimates of the first subdiagonal. The results for the other subdiagonals nearly match those in Figures 2.3 and 2.4, and are omitted. As p gets larger, there seems to be evidence of improvement in performance of the SC algorithm.

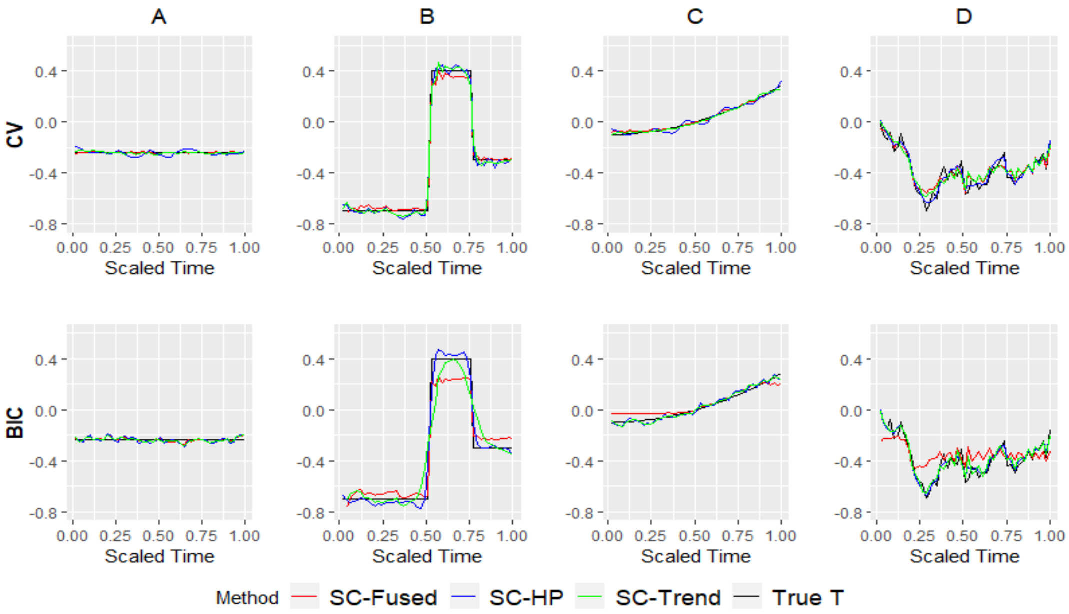


Figure 2.3: Estimated first subdiagonal of T for SC-HP, SC-Fused and SC-Trend ($p = 50$).

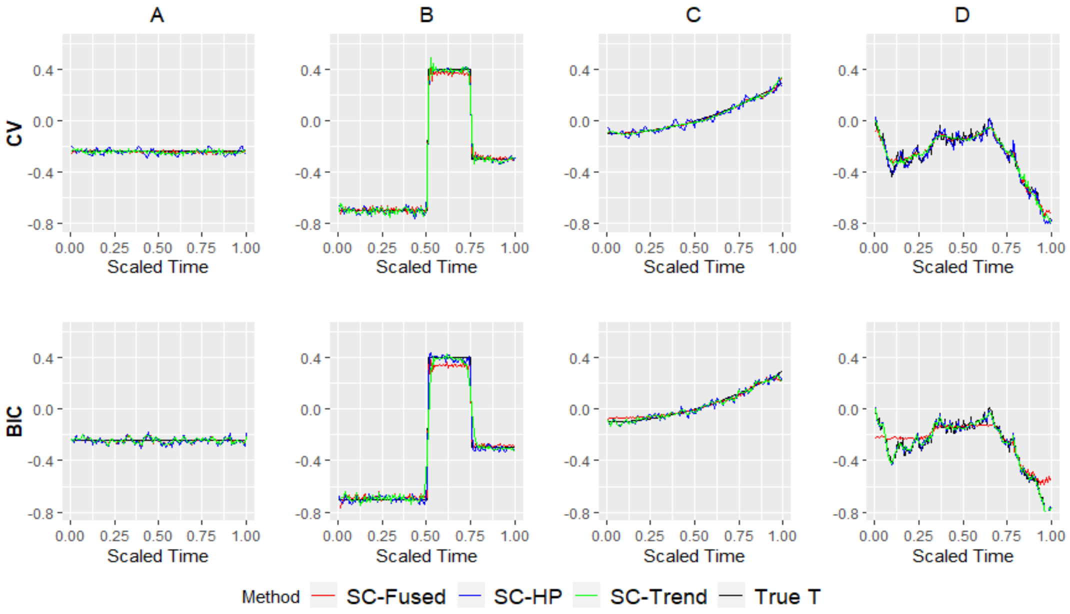


Figure 2.4: Estimated first subdiagonal of T for SC-HP, SC-Fused and SC-Trend ($p = 150$).

In Appendix A.3, we provide an additional simulation results to illustrate the variability of our method for each penalty function (SC-Fused, SC-HP, SC- Trend) and Case A-D.

2.4.3 Comparing Estimation Accuracies

In this section, we compare the accuracies of the three SC estimators: SC-HP, SC-Fused and SC-Trend. The overall measures of performance involve magnitudes of the estimation errors $\hat{T}-T$ and $\hat{L}-L$, as measured by the scaled Frobenius norm $\frac{1}{p}\|\hat{A}-A\|_F^2$, and the matrix infinity norm $|||\hat{A}-A|||_\infty$ for a $p \times p$ matrix A.

Boxplots of the overall estimation errors for the matrix T are reported in Figures 2.5 through 2.8, where each figure corresponds to a particular case, each row to a value of p and the two columns correspond to using BIC and CV criteria, respectively. They corroborate the findings in the graphical explorations Figures 2.3 and 2.4, in that the SC-Fused shows tendency to capture well cases with constant subdiagonals, SC-Trend and HP are better in capturing the wiggliness and smoothness of the subdiagonal. The corresponding estimation errors for the matrix L show similar patterns, and are thus omitted.

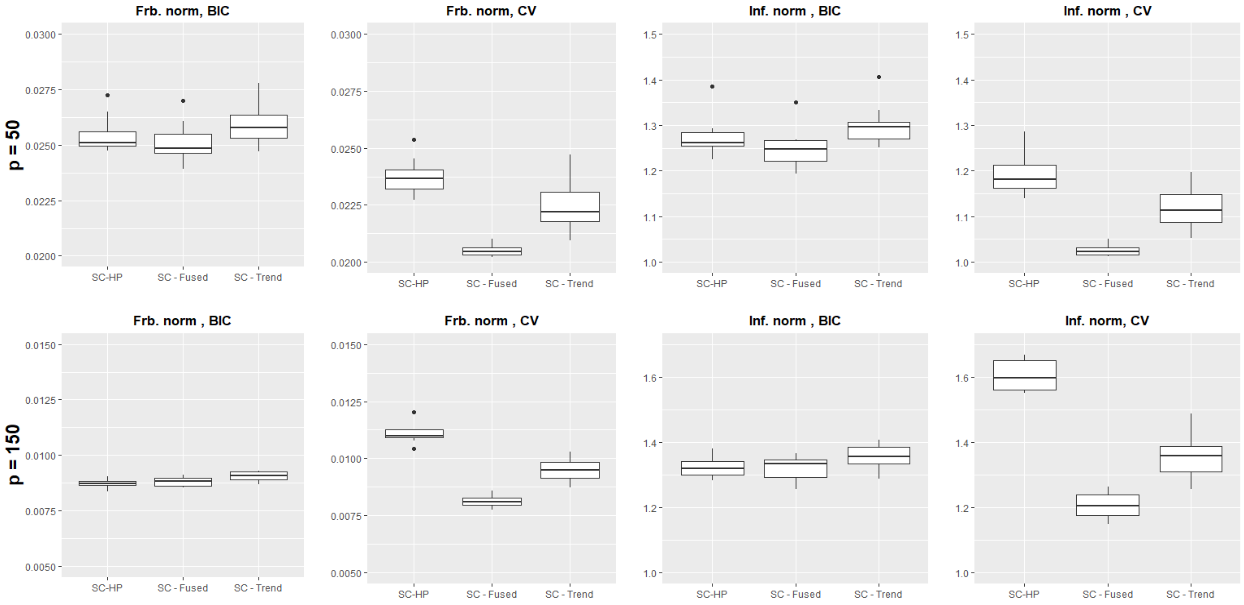


Figure 2.5: Estimation accuracy when data are generated from Case A.

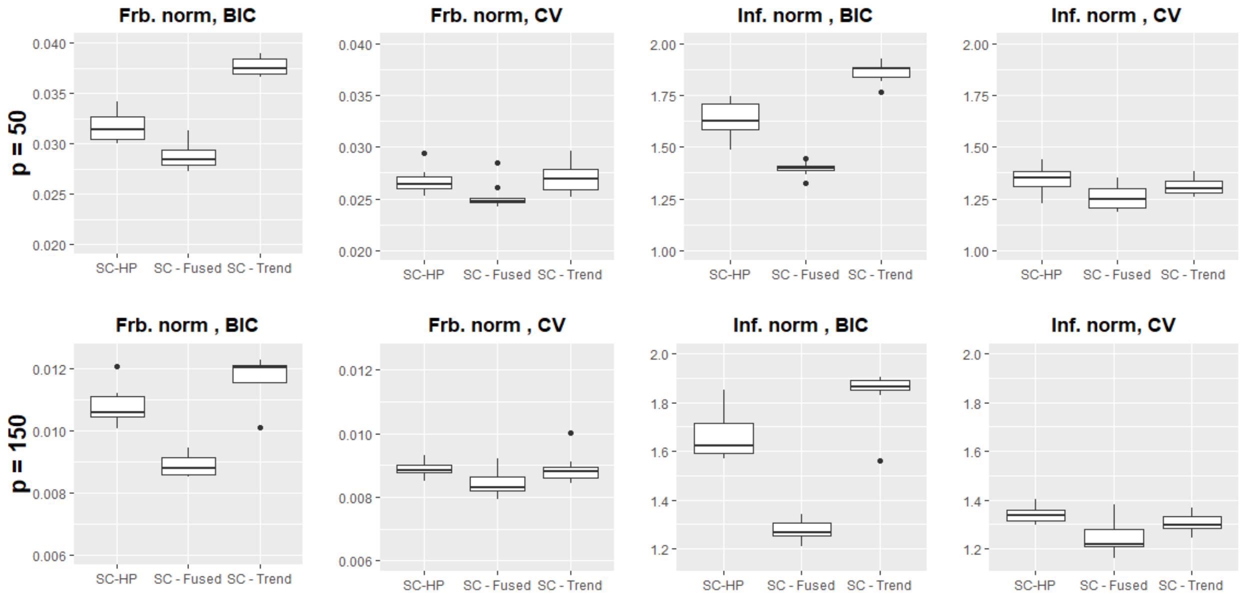


Figure 2.6: Estimation accuracy when data are generated from Case B.

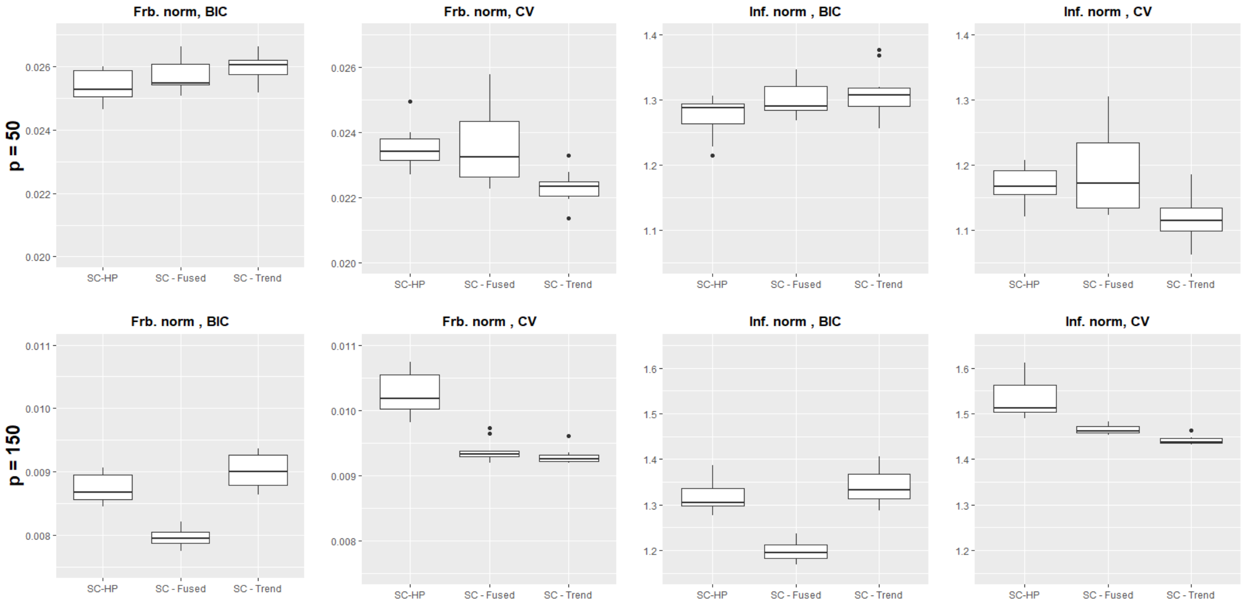


Figure 2.7: Estimation accuracy when data are generated from Case C.

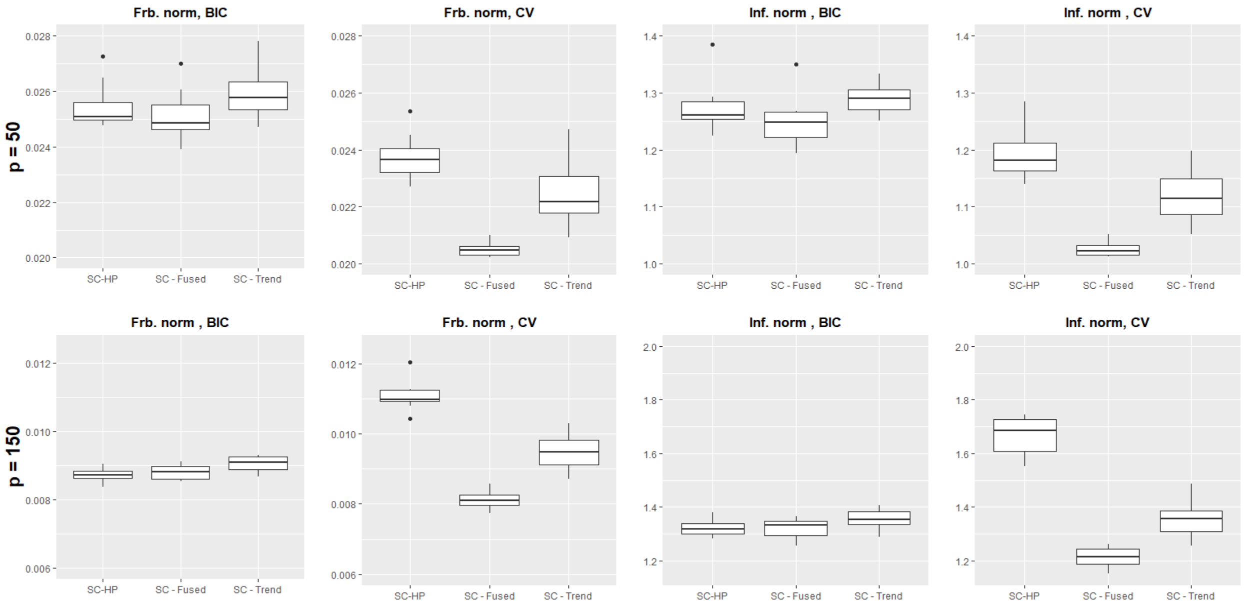


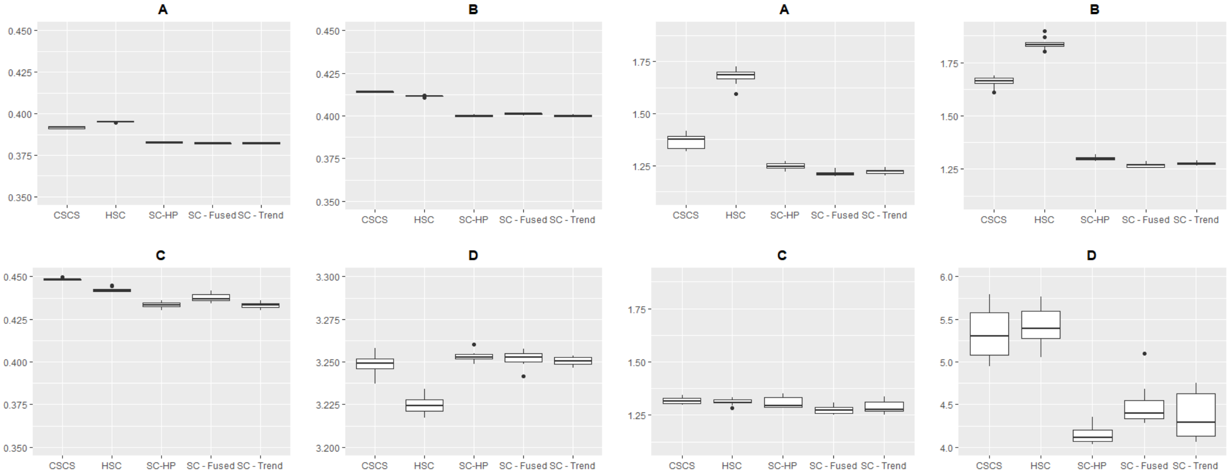
Figure 2.8: Estimation accuracy when data are generated from Case D.

In the Appendix A.3 we provide two additional simulations for a more general matrix T (L) than those discussed previously, and to compare our sparse SC with the existing sparse Cholesky estimators (CSCS, HSC) so far support recovery is concerned. The results confirm the good performance of the SC method. The two general matrices are: (1) T is a full lower triangular matrix and its subdiagonals are chosen randomly from the Cases (A-D), (2) T has a nonhierarchical structure (Yu and Bien, 2017), that is nonzero subdiagonals are followed by block zero subdiagonals and again by nonzero subdiagonals.

2.4.4 Covariance Estimators

In this section, we assess the performance of our method on learning (inverse) covariance matrices for the Cases A-D. We compare our SC method (Fused, HP, Trend) with the CSCS and HSC methods. To make them comparable, instead of limiting the SC algorithm to run over the first five subdiagonals, as in the last two sections, here we use the more general sparse SC estimator (see Lemma 2) with the two tuning parameters λ_1 and λ_2, respectively. Due to space limitation, we report results only for the $p = 150$ with the tuning parameters selected using the CV criterion.

We evaluate performance of the estimators using the scaled Kullback-Leibler loss $\frac{1}{p}\Big[tr(\hat{\Omega}\Sigma) - \ln|\hat{\Omega}\Sigma| - p\Big]$ for the inverse covariance and scaled Frobenious norm for the covariance matrix. From results reported in Figures 2.9a and 2.9b for cases A, B, and C, it is evident that the SC algorithm learns the covariance matrix better than the SCSC and HSC methods. In particular, for the case A, SC-Fused provides the lowest error measure and for cases B and C, SC-Trend and HP are the lowest. For the Case D, the HSC is the best. For learning the inverse covariance matrix, the SC performs better for all the four cases.



(a) Frobenious norm

(b) Kullback - Leibler loss

Figure 2.9: Performance of covariance and inverse covariance matrix estimators for $p = 150$.

2.4.5 The Cattle Data

This dataset Kenward (1987) is from an experiment in which cattle were assigned randomly to two treatment groups A and B. The weights of animals were recorded to study the effect of treatments on intestinal parasites. The animals were weighed $p = 11$ times over 122 days. Of 60 cattle $n = 30$ received treatment A and the other 30 received treatment B. The dataset has been widely used in the literature of longitudinal data analysis (Wu and Pourahmadi, 2003);Huang et al. (2007).

The classical likelihood ratio test rejected equality of the two within-group covariance matrices, thus it is recommended to study each treatment group's covariance matrix separately. In this paper, we report our results for the group A cattle. It is known (Zimmerman and Nunez-Anton, 2010) that the variances and the same-lag correlations are not constant, but tend to increase over time , so that the covariance exhibits nonstationarity features. To learn the 11×11 covariance matrix, we apply the following methods : SC (HP, Fused, Trend), sample covariance S, unstructured antedependence (AD) (Zimmerman and Nunez-Anton, 2010, Section 2.1), autoregression process (AR), variable-order antedependece (VAD) (Zimmerman and Nunez-Anton, 2010, Section 2.6) and the structured AD model in (Pourahmadi, 1999), referred to as POU in the following plot. More specifically, following Zimmerman and Nunez-Anton (2010, Section 8.2) we consider AD(2), VAD(0,1,1,1,1,1,1,2,2,1,1), AR(2), and POU model for which the log-innovation variances are a cubic function of time and the autoregressive coefficients are a cubic function of lag. Tuning parameters for all three SC methods were selected using a $5-$fold cross-validation.

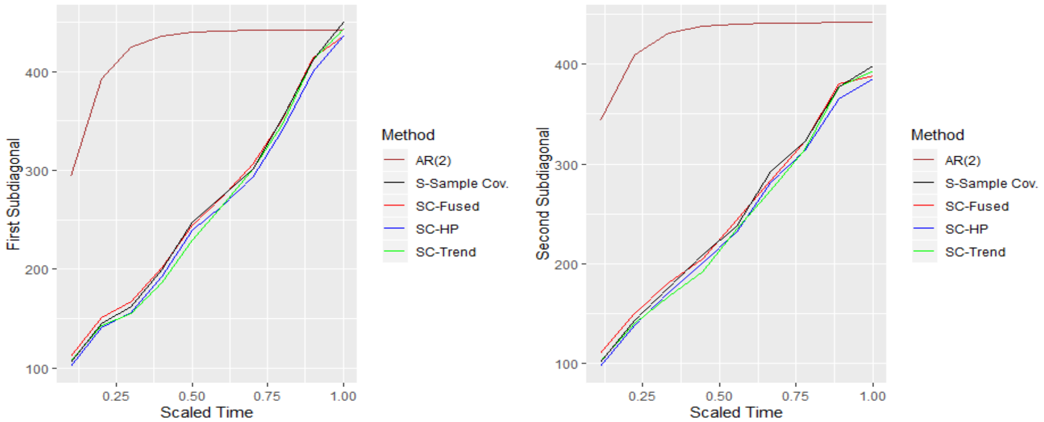


Figure 2.10: Plots of estimated first and second subdiagonals of the covariance matrix for the various estimation methods. The difference of the structure of subdiagonal curves from the AR model and Sample covariance matrix suggests adverse affect of the stationary assumption.

We plot the first two subdiagonals of estimated covariance matrices for the SC(HP, Fused, Trend), S and AR(2) methods in Figure 2.7. It can be seen that the estimators of subdiagonals provided by the SC methods are almost identical to those of the sample covariance matrix. However, the estimated subdiagonals from AR(2) illustrate a different behavior suggesting that the data does not support the underlying AR model. In Appendix we provide similar plots for all eight estimators. In Table 2.1 we report the values of the negative log-likelihood for various methods which also confirm the results in Figure 2.10. The maximum log-likelihood is in bold.

Table 2.1: Log-likelihood values for various estimation methods.

Method	
SC-HP	-541.836
SC-Fused	-547.129
SC-Trend	-546.573
AD(2)	-541.451
VAD	-542.861
AR(2)	-1,637.894
POU	-862.430
S	$\mathbf{-529.4207}$

2.4.6 The Call Center Data

In this section, we assess the forecast performance of the SC, CSCS, and HSC algorithms by analyzing the call center data (Huang et al., 2006), from a call center in a major U.S. northeastern financial organization. For each day in 2002 phone calls were recorded from 7:00 AM until midnight, the 17-hour interval was divided into 102 10-minute subintervals, and the number of calls arrived at the service queue during each interval were counted. Here, we focus on weekdays only, since the arrival patterns on weekdays and weekends differ.

We denote the counts for day i by the vector $N_i = (N_{i,1}, \dots, N_{i,102})',\ i = 1, \dots, 239$, where $N_{i,t}$ is the number of calls arriving at the call center for the tth 10-minute interval on day i. The square root transformation $x_{it} = \sqrt{N_{it} + 1/4},\ i = 1, \dots, 239,\ t = 1, \dots, 102$, is expected to make the distribution closer to normal. The estimation and forecast performances are assessed by splitting the 239 days into training and test datasets. In particular, to estimate the mean vector and the covariance matrix, we form the training dataset from the first T days ($T = 205, 150, 100, 75$). Six covariance estimators, five penalized likelihood methods, SC (HP, Fused, and Trend), CSCS and HSC, along with S were used to estimate the 102×102 covariance matrix of the data. The tuning (penalty) parameters were selected using 5-fold cross validation described in Section A.2. We report the log-likelihood (Khare et al., 2019) for the test dataset evaluated at all above estimators in Table 2.2, where the largest value in each column is in bold. For all training data sizes, the SC algorithm demonstrates superior performance compared to the other methods. In particular, for $T = 205, 150$, the SC-Trend is the best, but for $T = 100, 75$ the SC-Fused provides better results.

Table 2.2: Test data log-likelihood values for various estimation methods with training data size 205,150, 100, 75.

Methods		Training data size			
		205	150	100	75
SC	HP	-14,435.700	-9,018.556	-7,472.817	-7,467.412
	Fused	-13,123.300	-8,587.785	**−7,034.868**	**−7,097.938**
	Trend	**−12,274.970**	**−8,477.271**	-7,040.924	-7,222.989
Sparse Cholesky	CSCS	-16,814.450	-9,754.996	-7,484.153	-7,365.298
	HSC	-14,382.330	-8,971.729	-7,395.206	-7,342.343

Next, we focus on forecasting the number of call arrivals in the later half of the day using arrival patterns in the earlier half of the day (Huang et al., 2006). In particular, for a random vector $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,102})'$, we partition $\mathbf{x}_i = ((x_i^{(1)})', (x_i^{(2)})')'$ where $x_i^{(1)}$ and $x_i^{(2)}$ are 51-dimensional vectors that correspond to early and later arrival patterns for day i. Assuming multivariate normality, the optimal mean squared error forecast of $x_i^{(2)}$ given $x_i^{(1)}$ is

$$E(x_i^{(2)} | x_i^{(1)}) = \mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(x_i^{(1)} - \mu_1), \tag{2.21}$$

Table 2.3: Number of times (out of 51) each estimation method achieves the minimum forecast error for training data size 205, 150, 100, 75.

	Training data size			
Method	205	150	100	75
SC-HP	1	5	9	**20**
SC-Fused	3	7	2	3
SC-Trend	7	8	**16**	1
CSCS	3	8	10	15
HSC	12	**13**	14	12
S	**25**	10	-	-

corresponding to partitioning of the mean and covariance matrix of the full vector:

$$\mu' = (\mu_1', \mu_2'),\ \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22}. \end{bmatrix}$$

We compare the forecast performance of six covariance estimators (SC (HP, Fused, Trend), CSCS, HSC, and S) by using training and test datasets described above. The sample mean and covariance matrix are computed from the training data for each T . Using (2.21), the 51 first half of a day arrival counts were used to forecast the second half of the day arrival counts. For each time interval $t = 52, \ldots, 102$, we define the forecast error (FE) by the average

$$FE_t = \frac{1}{239 - T} \sum_{i=T+1}^{239} |\hat{x}_{it} - x_{it}|,$$

where x_{it} and $\hat{x}_{it}$ are the observed and forecast values, respectively (Huang et al., 2006). Table 2.3 reports the number of times each of the six forecast methods has the minimum forecast error values out of the total 51 trials. The maximum of the number of times the method achieves the minimum forecast error in each column is in bold. When $T = 205$ and the training data size is larger than the number of variables, the forecast based on the Sample covariance matrix performs the best in terms of the number of times it achieves the minimum forecast error. For $T = 150$, the HSC is a bit better than the sample covariance matrix. However, as the training data size decreases, the forecasting ability of the SC algorithm increases. In particular, the SC-Trend and HP report the best result in terms of the number of times they achieve the minimum forecast error for $T = 100$ and $T = 75$, respectively. Most of the result in Table 2.3 is supported by the aggregate forecast errors reported in Table 2.4, where aggregate forecast error is the sum of forecasted errors over $t = 52, \dots, 102$. The minimum aggregate forecast error the method achieves is in bold. The discrepancies between Table 2.3 and Table 2.4 can be explained by looking on Figure 2.11, which illustrates a plot of FE_t for varying values of the training data size. For example, for $T = 150$ the HSC achieves the minimum forecast error the most in terms of the number of times, however SC-Fused is the lowest in terms of the aggregate forecast error. This discrepancy explained from the top right plot of Figure 2.11, where it can be seen that when the FE_t of HSC is lowest, the FE_t of SC-Fused does not concede to much, but when the FE_t of SC-Fused is the lowest, HSC takes higher values, which forces the aggregate error of SC-Fused to be lower than the error of HSC,

Table 2.4: Aggregate forecast error for training data sizes 205,150,100,75.

	Training data size			
Methods	205	150	100	75
SC-HP	403.555	34.093	24.186	$\mathbf{9.363}$
SC-Fused	377.938	$\mathbf{25.299}$	31.377	44.403
SC-Trend	371.936	31.981	23.202	23.565
CSCS	307.096	38.130	28.578	13.799
HSC	151.817	28.299	$\mathbf{22.917}$	11.214
S	$\mathbf{111.276}$	42.432	$-$	$-$

An R (R Core Team, 2019) package, named `SC`, is available on Github repository (Dallakyan, 2019). The core functions are coded in C++, allowing us to solve large-scale problems in substantially less time.

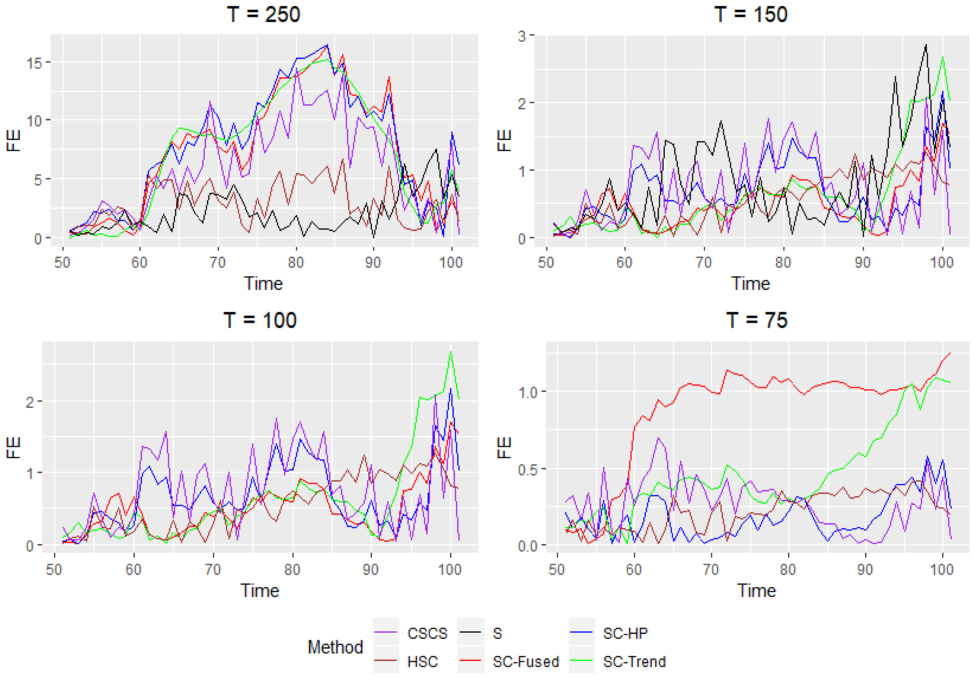


Figure 2.11: Forecast Error for each estimation method for training data size 205,150,100,75

3. LEARNING BAYESIAN NETWORKS THROUGH BIRKHOFF POLYTOPE

3.1 Introduction

Bayesian Networks (BNs) are a popular class of graphical models whose structure is represented by a DAG $\mathcal{G}$. BNs are interdisciplinary subjects that have been used in many applications such as economics, finance, biology, etc (Bessler and Akleman, 1998; Neil et al., 2005; Needham et al., 2007; Dallakyan, 2020). In recent years the following main approaches have been evolved to learn the structure of the underlying DAG from data: Independence-based (also called constraint-based) methods (Pearl, 2009; Spirtes and Glymour, 1991) and score-based methods (Heckerman et al., 1995; Chickering, 2002; Teyssier and Koller, 2005; Loh and Bühlmann, 2014). Here, structure learning refers to recovering DAG from observational data.

Independence-based methods, such as the inductive causation (IC) (Pearl, 2009) and PC (Peter-Clark) (Spirtes and Glymour, 1991) algorithm, utilize conditional independence tests to detect the existence of edges between each pair of variables. The majority of independence-based methods require a faithfulness assumption for the joint distribution $\mathcal{P}$ to the DAG, where $\mathcal{P}$ is faithful to the DAG $\mathcal{G}$ if all conditional independencies in $\mathcal{P}$ are entailed in $\mathcal{G}$.

In contrast, score-based methods measure the goodness of fit of different graphs over data by optimizing a score function with respect to the unknown (weighted) adjacency matrix B with a combinatorial constraint that the graph is DAG. Then use a search procedure to find the best graph. Commonly used search procedures include hill-climbing (Heckerman et al., 1995; Tsamardinos et al., 2006), forward-backward search (Chickering, 2002), dynamic, and integer programming (Silander and Myllymäki, 2006; Koivisto, 2006; Jaakkola et al., 2010; Studený and Haws, 2014; Hemmecke et al., 2012). Recently, Zheng et al. (2018, 2020) proposed a fully continuous program for structure learning by introducing a novel characterization of acyclicity constraint. Generally, the DAG search space is intractable for a large number of nodes p and the task of finding a DAG is NP-hard (Chickering, 2002). To make the space tractable, approximate methods have been proposed with additional assumptions such as bounded maximum indegree of the node (Cooper and Herskovits, 1992) and tree-like structures (Chow and Liu, 1968).

In parallel with searching in DAG space, ordering space (or the space of topological ordering) has been exploited for score-based methods (Teyssier and Koller, 2005; van de Geer and Bühlmann, 2013; Aragam and Zhou, 2015; Ye et al., 2020) in which the topological ordering is considered as an additional parameter in the score function. Here, a score of particular order is defined as the score of the best DAG consistent with it (Teyssier and Koller, 2005), and a permutation π of the vertex set $V = \{1, \cdots p\}$ is a topological order for DAG if $\pi(j) < \pi(k)$ whenever $(j, k) \in E$. Such a topological order exists for all DAGs, but it may not be unique (Koller and Friedman, 2009). The order-based search has two main advantages: The ordering space ($2^{O(p \log p)}$) is significantly smaller than the DAG search space ($2^{O(p^2)}$), and the existence of ordering guarantees satisfaction of the acyclicity constraint.

Annealing on Regularized Cholesky Score (ARCS) algorithm, proposed in Ye et al. (2020), represents an ordering by the corresponding permutation matrix P, and then given the ordering, encodes the weighted adjacency matrix B into the Cholesky factor L of the inverse covariance matrix. ARCS optimizes a regularized likelihood score function to recover a sparse DAG structure and utilizes simulated annealing (SA) to search over P. In SA, using pre-specified constant m and a temperature schedule $\{T^{(i)}, i = 0, \ldots, N\}$, in each ith iteration the new permutation matrix P^* is proposed by flipping a fixed-length m random interval in the current permutation $\hat{P}$, and checking whether to stay at the current $\hat{P}$ or move to the proposed P^* with some probability.

Motivated by the ARCS two-part framework, we propose an order-based method for learning Gaussian DAGs by optimizing a non-convex regularized likelihood score function. Our proposal has the following distinct features and advantages. First, instead of an expensive search of a permutation matrix P in the non-convex space of permutation matrices, we propose the following relaxation: project P onto the Birkhoff polytope (the convex space of doubly stochastic matrices) and then find the "closest" permutation matrix to the optimal doubly stochastic matrix (See Figure 3.2). The projection step includes a concave regularization term, which pushes the projected doubly stochastic matrix "closer" to the permutation matrix if the penalization parameter is sufficiently large. The proposed relaxation is convex if the number of observations exceeds the number of variables (Lemma 6).

Second, given P, we resort to the cyclic coordinatewise algorithm to recover the DAG structure entailed in the Cholesky factor L. We show that the optimization reduces to p decoupled penalized regressions where each iteration of the cyclic coordinatewise algorithm has a closed form solution. Moreover, the convergence of iterates to the stationary point is guaranteed.

Third, on the statistical side, our method produces a consistent Cholesky factor estimator for the non-convex score function, assuming that the true permutation matrix is known. To the best of our knowledge, consistency results for the sparse Cholesky factor estimator were established only for convex problems (Yu and Bien, 2017; Khare et al., 2019).

3.2 Bayesian Networks

We start by introducing the following graphical concepts. If the graph $\mathcal{G}$ contains a directed edge from the node $k \to j$, then k is a parent of its child j. We write $\Pi_j^{\mathcal{G}}$ for the set of all parents of a node j. If there exist a directed path $k \to \ldots \to j$, then k is an ancestor of its descendant j. A *Bayesian Network* is a directed acyclic graph $\mathcal{G}$ whose nodes represent random variables $X_1, \ldots, X_p$. Then $\mathcal{G}$ encodes a set of conditional independence assumptions and conditional probability distributions for each variable. It is well-known that for a BN, the joint distribution factorizes as:

$$P(X_1, \ldots, X_p) = \prod_{j=1}^{p} P(X_j | \Pi_j^{\mathcal{G}})$$

The DAG $\mathcal{G} = (V, E)$ is characterized by the node set $V = \{1, \ldots, p\}$ and the edge set $E = \{(i, j) : i \in \Pi_j^{\mathcal{G}}\} \subset V \times V$.

3.2.1 Gaussian BN and Structural Equation Models

In this section, we focus on recovering the DAG of the Gaussian BN. Saliently, the Gaussian BN can be equivalently represented by the linear SEM:

$$X_j = \sum_{k \in \Pi_j^{\mathcal{G}}} \beta_{jk} X_k + \varepsilon_j,\ j = 1, \ldots, p, \tag{3.1}$$

where $\varepsilon_j \sim N(0, \omega_j^2)$ are mutually independent and independent of $\{X_k : k \in \Pi_j^{\mathcal{G}}\}$. Denoting $B = (\beta_{jk})$ with zeros along the diagonal, the vector representation of (3.1) is

$$X = BX + \varepsilon, \tag{3.2}$$

where $\varepsilon := (\varepsilon_1, \ldots, \varepsilon_p)^t$ and $X := (X_1, \ldots, X_p)^t$. We can characterize the linear SEM $X \sim (B, \Omega)$ by the weighted adjacency matrix B and the noise variance matrix $\Omega = \text{diag}(\omega_1^2, \ldots, \omega_p^2)$. From (3.2), the inverse covariance matrix of $X \sim N_p(0, \Sigma)$ is $\Sigma^{-1} = (I - B)^t \Omega^{-1} (I - B)$, where B^{-t} denotes the inverse transpose of the matrix B, and the edge set of the underlying DAG is equal to the support of the weighted adjacency matrix B; i.e., $E = \{(k, j) : \beta_{jk} \neq 0\}$, which defines the structure of DAG $\mathcal{G}$. Consequently, B should satisfy the acyclicity constraint so that $\mathcal{G}$ is indeed a DAG.

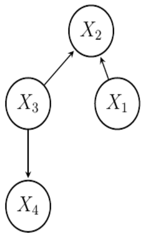
As discussed in the Introduction, each DAG admits a topological ordering π. To each π, we associate a $p \times p$ permutation matrix P_π, such that $P_\pi x = (x_{\pi(1)}, \ldots, x_{\pi(p)})$, for $x \in R^p$. The existence of a topological order leads to the permutation-similarity of B to a strictly lower triangular matrix $B_\pi = P_\pi B P_\pi^t$ by permuting rows and columns of B, respectively (see Figure 3.1 for the illustrative example). Therefore, the stringent acyclicity constraint on B transforms into the constraint that B_π is a strictly lower triangular matrix, and the linear SEM model can be represented by

$$P_\pi X = B_\pi P_\pi X + P_\pi \varepsilon, \tag{3.3}$$

since $P_\pi^t P_\pi = I$. From (3.3), the inverse covariance matrix can be expressed as

$$\Sigma_\pi^{-1} = (I - B_\pi)^t \Omega_\pi^{-1} (I - B_\pi), \tag{3.4}$$

where $\Omega_\pi = P_\pi \Omega P_\pi^t$. From (3.3) and (3.4), defining $L_\pi = (I - B_\pi)\Omega_\pi^{-1/2}$, the relationship between



$$B = \begin{array}{c|cccc} & X_1 & X_2 & X_3 & X_4 \\ \hline X_1 & 0 & 0 & 0 & 0 \\ X_2 & \beta_{21} & 0 & \beta_{23} & 0 \\ X_3 & 0 & 0 & 0 & 0 \\ X_4 & 0 & 0 & \beta_{43} & 0 \end{array} \quad P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B_\pi = \begin{array}{c|cccc} & X_1 & X_3 & X_2 & X_4 \\ \hline X_1 & 0 & 0 & 0 & 0 \\ X_3 & 0 & 0 & 0 & 0 \\ X_2 & \beta_{21} & \beta_{23} & 0 & 0 \\ X_4 & 0 & \beta_{43} & 0 & 0 \end{array}$$

Figure 3.1: Illustration of DAG $\mathcal{G}$, corresponding coefficient matrix B, permutation matrix P, and permuted strictly lower triangular matrix B_π.

the Cholesky factor L_π of the inverse covariance matrix $\Sigma_\pi^{-1} = L_\pi^t L_\pi$ and the matrix B_π is

$$\begin{aligned} &(L_\pi)_{ij} = -(B_\pi)_{ij}/\sqrt{\omega_j}, \text{ and} \\ &(L_\pi)_{ij} = 0 \iff (B_\pi)_{ij} = 0 \text{ for every } i \geq j \end{aligned} \tag{3.5}$$

Hence, L_π preserves the DAG structure of B_π; i.e., non-zero elements in L_π correspond to directed edges in DAG $\mathcal{G}$.

3.3 Score Function

In this section, given data from the Gaussian BN (or SEM), we derive the form of the score function to recover the underlying DAG structure. We assume that each row of data matrix $\mathbf{X} = (X_1, \ldots, X_p) \in R^{n \times p}$ is an i.i.d observation from (3.1). Using reformulation (3.3),

$$\mathbf{X}P_\pi^t = \mathbf{X}P_\pi^t B_\pi^t + \mathbf{E}P_\pi^t, \tag{3.6}$$

where each row of $\mathbf{E}$ is an i.i.d vector and follows $N_p(0, \Omega)$. Thus, each row of $\mathbf{X}P_\pi^t$ is, again, an i.i.d from $N_p(0, \Sigma_\pi)$, and the negative log-likelihood for (3.6) is:

$$\begin{aligned}\ell(B_\pi, \Omega_\pi, P_\pi|\mathbf{X}) = &\frac{1}{2}\text{tr}\Big(P_\pi \mathbf{X}^t \mathbf{X} P_\pi^t (I - B_\pi)^t \Omega_\pi^{-1} (I - B_\pi)\Big) \\ &+ \frac{n}{2}\log|\Omega_\pi|,\end{aligned} \tag{3.7}$$

where we used $\Sigma_\pi = \text{cov}(P_\pi X) = P_\pi \Sigma P_\pi^t = (I - B_\pi)^{-1}\Omega_\pi (I - B_\pi)^{-t}$, and B_π is a strictly lower triangular matrix. From now on, whenever there is no confusion in the context, we drop the subscript π from P_π, B_π, Ω_π and Σ_π.

After reparametrizing (3.7) in terms of L and P, we obtain the following log-likelihood function:

$$\ell(L, P|\mathbf{X}) = \frac{1}{2}\text{tr}\Big(PSP^t L^t L\Big) - \sum_{j=1}^{p} \log L_{jj}, \tag{3.8}$$

where we used $S = \mathbf{X}^t\mathbf{X}/n$ for the sample covariance and matrix determinant $|\Omega|^{-1/2} = |L| = \prod_{j=1}^{p} L_{jj}$. Given the permutation matrix, we denote the optimal value of (3.7) as

$$\ell^*(P) = \min_{L \in \mathcal{L}_p} \ell(L, P), \tag{3.9}$$

where $\mathcal{L}_p$ is the space of lower triangular matrices with positive diagonal entries. Ye et al. (2020, Proposition 1) showed that $\ell^*(\cdot)$ is invariant to permutations, and maximum likelihood does not favor any particular ordering. Consequently, all maximum likelihood DAGs corresponding to a different permutation produce the same Gaussian likelihood.

In order to break the permutation invariance in (3.9), we follow Ye et al. (2020) and regularize the negative log-likelihood function to favor sparse DAGs. We consider the following penalized loss function:

$$\begin{aligned}Q(L; P) = &\frac{1}{2}\text{tr}\Big(PSP^t L^t L\Big) - \sum_{j=1}^{p} \log L_{jj} \\ &+ \sum_{1 \le j \le i \le p} \rho(|L_{ij}|; \lambda),\end{aligned} \tag{3.10}$$

where the penalty function $\rho(\cdot, \lambda) : \mathcal{R} \to \mathcal{R}$ satisfies conditions listed in Loh and Wainwright (2015) and reiterated in Appendix for convenience. Those conditions are important for establishing theoretical properties of our estimator in Section 3.6:

From (3.10), a permutation leading to a sparser Cholesky factor L has a smaller loss value $Q(L; P)$, and a DAG learning problem can be restricted to finding a permutation matrix, which gives the sparsest solution L; i.e.,

$$\begin{aligned} \min_{L \in \mathcal{L}_p, P \in \mathcal{P}_p} Q(L, P) = \min_{L,P} \Big\{ &\frac{1}{2} \text{tr}\Big(PSP^t L^t L \Big) \\ &- \sum_{j=1}^{p} \log L_{jj} \\ &+ \sum_{1 \le j \le i \le p} \rho(|L_{ij}|; \lambda) \Big\}, \end{aligned} \tag{3.11}$$

where $\mathcal{P}_p$ is the set of all $p \times p$ permutation matrices.

3.4 A Minimization Algorithm

We now provide an algorithm to minimize the score function (3.11), called Relaxed Regularized Cholesky Factor (RRCF), with respect to P and L, respectively. It has two main steps formulated in Algorithm 2. First, we propose a regularized relaxation to solve the optimization problem in line 5 through a gradient projection algorithm (see Algorithm 4). Then estimate a Cholesky factor in line 6 utilizing a cyclic coordinatewise algorithm (see Algorithm 5). We show that in the first step, a convex relaxation can be achieved when the number of observations exceeds the number of variables.

3.4.1 Optimization Over the Permutation Space

A paramount issue in finding an optimal permutation matrix is that the number of fixed arrangements of variables is $p!$. Ye et al. (2020) mitigate the problem by using simulated annealing technique to search over the permutation space. Our approach is significantly different and relies on enlarging the non-convex set of permutation matrices by the convex set of doubly stochastic matrices (Birkhoff polytope) and finding the “closest” permutation matrix to the optimal doubly

Algorithm 2 RRCF algorithm

```
1: *input*:
2: $\lambda, k_{max} \leftarrow$ *Tuning Parameter, iteration*
3: $L^{(0)}, P^{(0)} \leftarrow$ *Initial matrices*
4: *while* $k < k_{max}$:
5:     $\hat{P}^{(k)} = \arg\min_{P \in \mathcal{P}_p} Q_{RRCF}(L^{(k-1)}, P)$
6:     $\hat{L}^{(k)} = \arg\min_{L \in \mathcal{L}_p} Q_{RRCF}(L, P^{(k)})$
7:     $k = k + 1$
8: *Output*: $(\hat{L}, \hat{P})$
```

stochastic matrix. Inspired from the recent advances in Seriation (Fogel et al., 2013) and Graph Matching problems (Zaslavskiy et al., 2009; Wolstenholme and Walden, 2016), we propose a relaxation to the hard combinatorial problem.

3.4.1.1 Relaxation

The impetus of this section is the framework developed in Fogel et al. (2013, Section 3.2). The optimization in line 5 of Algorithm 2 can be written as:

$$\begin{aligned} \min_{P} \quad & \frac{1}{2} tr(LPSP^tL^t) \\ \text{s.t.} \quad & P \in \mathcal{P}_p, \end{aligned} \tag{3.12}$$

where we eliminate terms that are constant with respect to P. We denote the Birkhoff polytope by $\mathcal{D}_p$ (the space of doubly stochastic matrices), where $\mathcal{D}_p = \{A \in R^{p \times p} : A \geq 0, A\mathbf{1} = \mathbf{1}, A^t\mathbf{1} = \mathbf{1}\}$. The polytope $\mathcal{D}_p$ has $p!$ vertices and dimension of $(p-1)^2$. It is informative to note that every permutation matrix is a doubly stochastic matrix, and a matrix is a permutation if and only if it is both doubly stochastic and orthogonal; i.e., $\mathcal{P}_p = \mathcal{D}_p \cap \mathcal{O}_p$, where $\mathcal{O}_p$ is the set of $p \times p$ orthogonal matrices. Moreover, from Birkhoff's Theorem, every doubly stochastic matrix can be written as a convex combination of permutation matrices and the set of doubly stochastic matrices is the convex hull of the set of permutation matrices (Horn and Johnson, 2012, Theorem 8.7.2), where permutation matrices are vertices (extreme points) of the polytope. More on Birkhoff polytopes and its properties can be found in Brualdi and Gibson (1977).

Since the sample covariance matrix $S \succcurlyeq 0$ is positive semi-definite, we can introduce a convex relaxation to the combinatorial problem (3.12) by replacing $\mathcal{P}_p$ with its convex hull $\mathcal{D}_p$:

$$\begin{aligned} \min_P \quad & \frac{1}{2} tr(LPSP^tL^t) \\ \text{s.t.} \quad & P \in \mathcal{D}_p, \end{aligned} \tag{3.13}$$

However, as we show in Corollary 1, the solution of (3.13) is not an acceptable candidate for developing our algorithm. Before introducing the corollary, in the next lemma, we list well-known properties of doubly stochastic matrices that are used to establish the framework for the relaxation. Since we are not aware of a source to cite, we give proof in the Appendix for completeness. We denote by $J \in \mathcal{D}_p$ the $p \times p$ matrix all of whose entries are 1.

Lemma 5. *For any $p \times p$ doubly stochastic matrix $P \in \mathcal{D}_p$,*

$$1 \leq \|P\|_F \leq \sqrt{p}$$

The left and right equalities hold if and only if $P = J/p$ and P is a permutation matrix, respectively.

From the Lemma 5, the following corollary easily follows.

Corollary 1. *The optimal solution of (3.13) is $\hat{P} = J/p$.*

Thus, the solution of (3.13) is the center of the Birkhoff polytope (Ziegler, 1995, page 20) and far from vertices where permutation matrices are located. To force it closer to the vertex, we utilize Lemma 5 to add a proper penalty to the objective function (See Figure 3.2 for the geometric depiction.)

$$\begin{aligned} \min_P \quad & \frac{1}{2} tr(LPSP^tL^t) - \frac{1}{2}\mu\|P\|_F^2 \\ \text{s.t.} \quad & P \geq 0, P\mathbf{1} = \mathbf{1}, P^t\mathbf{1} = \mathbf{1}, \end{aligned} \tag{3.14}$$

where for the large enough value $\mu > 0$, $\|P\|_F^2$ achieves its upper bound, which is p from the Lemma 5; i.e., the larger μ, the closer the solution of (3.14) is to a permutation matrix. Similar

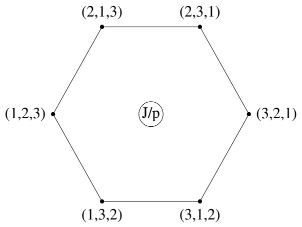


Figure 3.2: A geometric depiction of relaxation (3.14) for $\mathcal{D}_3$ Birkhoff polytope. Here, vertices represent permutations, and matrix J/p indicates the center of the polytope.

to Fogel et al. (2013, Proposition 3.5), the next lemma shows that the convexity of the objective function (3.14) depends on the intertwined values of μ and the smallest eigenvalue of S and L^tL. As a result, the convexity is untenable when $n << p$. We mitigate this problem by introducing an additional transformation to maintain convexity when $n \approx p$ (Lemma 6(b)). The following notation is used in the lemma: we write $\lambda_1 < \lambda_2 < \cdots < \lambda_m$ as an ordered, distinct eigenvalues of the $p \times p$ matrix. The proof is provided in the Appendix for completeness.

Lemma 6. *a. If $\mu \leq \lambda_1(S)\lambda_1(L^tL)$, optimization problem (3.14) is convex in P.*

b. If $\mu \leq \lambda_2(S)\lambda_1(L^tL)$ and $T = \mathbf{I} - \frac{1}{p}\mathbf{1}\mathbf{1}^t$ is the projection matrix into the orthogonal complement of $\mathbf{1}$, then the optimization problem

$$\begin{aligned} \min_{P} \quad & \frac{1}{2}tr(LPSP^tL^t) - \frac{1}{2}\mu\|TP\|_F^2 \\ \textit{s.t.} \quad & P \geq 0, P\mathbf{1} = \mathbf{1}, P^t\mathbf{1} = \mathbf{1}, \end{aligned} \tag{3.15}$$

is equivalent to problem (3.14) and is convex in P.

c. If $\mu > \lambda_m(S)\lambda_m(L^tL)$, optimization problem (3.14) is concave in P and the solution is a permutation matrix.

From Lemma 6(a) and (b), for $n << p$, $\lambda_2(S)$ is zero, and there is no $\mu > 0$ that validates convexity of (3.15). The question we investigate next is whether, under the convexity assumption of Lemma 6(a) or (b), there is a value of μ that asymptotically achieves “closeness” to the permutation matrix in terms of Frobenius norm. We provide the answer only for (3.14), but the similar result holds for (3.15) by analogy. Recall that we tacitly assume the condition $n > p$ to maintain convexity.

Lemma 7. *Under convexity condition in Lemma 6(a), for $\mu > 0$*

$$\|\hat{P} - P\|_F \not\to 0, \text{ as } n \to \infty, \tag{3.16}$$

where $P \in \mathcal{P}_p$ and $\hat{P}$ is the solution of (3.14).

The proof can be found in Appendix. Lemma 7 suggests that under a convexity condition, the solution of (3.14) does not get "close" to the permutation matrix, even when $n \to \infty$. The result may encourage the use of higher values of μ, resulting in a non-convex objective function. However, this approach is not recommended. Our empirical results suggest that for comparably large μ, the RRCF algorithm becomes independent from the data and highly dependent on the initial choice of P. Consequently, it gets stuck at one of the extreme points of the Birkhoff polytope. The choice of μ for this setting is an open question and left for further investigation. Here, when $n < p$, we propose to treat μ as a tuning parameter and use information criteria or cross-validation for the selection.

3.4.1.2 Gradient Projection Algorithm

We provide details for solving (3.15), but the procedure similarly applies to (3.14). Optimization (3.15) is a quadratic program (QP), and rich literature exists on solving this class of problems. In this section, we rely on the Gradient Projection (Bertsekas, 2015) method and show the convergence of the algorithm. Algorithm 3 outlines general steps, where $[\cdot]^+$ denotes projection on the space of doubly stochastic matrices $\mathcal{D}_p$.

Line 6 of the algorithm requires projection onto the Birkhoff polytope, which can be efficiently implemented by the block coordinate ascent, where each iteration has a closed form solution. The details on the block coordinate ascent algorithm are given in the next section. The following lemma guarantees convergence of Algorithm 3, where T is the projection matrix defined in Lemma 6.

Algorithm 3 Gradient Projection

```
1: input:
2: $k_{max}, \mu, \eta \leftarrow$ the number of iterations and positive scalars
3: $L, P^{(0)} \leftarrow$ Cholesky and Initial Permutation matrix
4: Set initial stepsize and update: $\bar{P}^0 = P^0$
5: while $\|P^{(k+1)} - P^{(k)}\| > \epsilon$ :
6:     $\hat{P}^{(k+1)} = [P^{(k)} - \eta \nabla Q_{RRCF}(P^{(k)}, L)]^+$  via Algorithm 8
7:     $P^{(k+1)} = P^{(k)} + \alpha^k(\hat{P}^{(k+1)} - P^{(k)})$
8:     $k = k + 1$
9: Output: Doubly Stochastic Matrix $P$
```

Lemma 8. *For $\mu \leq \lambda_2(S)\lambda_1(L^tL)$ and with a constant stepsize $\eta \in (0, 2/(\|S\|\|L^tL\|+\mu\|T\|))$, Algorithm 3 converges to the global minimum.*

The proof is provided in Appendix.

3.4.1.3 Projection onto the Birkhoff Polytope

Here, we give details on solving line 6 of Algorithm 3. For a given matrix P_0, the projection onto $P \in \mathcal{D}_p$ can be written as

$$\begin{aligned} \min_{P} \quad & \frac{1}{2}\|P - P_o\|_F^2 \\ \text{s.t.} \quad & P \geq 0, P\mathbf{1} = \mathbf{1}, P^t\mathbf{1} = \mathbf{1}. \end{aligned} \tag{3.17}$$

The Lagrangian of (3.17) is (Bertsekas, 2015)

$$\begin{aligned} \mathcal{L}(P, u, v, U) = \frac{1}{2}&\|P - P_0\|_F^2 + u^t(P\mathbf{1} - \mathbf{1}) \\ &+ v^t(P^t\mathbf{1} - \mathbf{1}) - tr(U^tP), \end{aligned}$$

and the dual objective function is defined as:

$$\mathcal{L}_*(u, v, U) = \inf_{P} \mathcal{L}(P, u, v, U). \tag{3.18}$$

Consequently, the dual problem of (3.17) is (see Appendix B.2 for details)

$$\begin{aligned}\max_{u,v,U} \quad & -\frac{1}{2}\|u\mathbf{1}^t + \mathbf{1}v^t - U\|_F^2 - tr(U^t P_0) \\ & + u^t(P_0\mathbf{1} - \mathbf{1}) + v^t(P_0^t\mathbf{1} - \mathbf{1}) \\ \text{s.t.} \quad & U \geq 0,\end{aligned} \tag{3.19}$$

Following Fogel et al. (2013, Section 4.2), we use the block coordinate ascent algorithm to optimize the dual problem (3.19). We show that each block update has a closed form solution. Details of the algorithm and the derivation of closed form solutions are relegated to the Appendix.

In the next section, we propose a framework to find the "closest" permutation matrix to the doubly stochastic matrix solution (3.14) or (3.15).

3.4.1.4 Sampling Permutations from the Doubly Stochastic Matrix

From Lemma 7, under a convexity condition or for moderately large values of μ, the solution of a convex relaxation (3.14) or (3.15) does not result in a permutation matrix P. Thus, after finding a doubly stochastic matrix, we need to project the solution to the "closest" matrix $P \in \mathcal{P}_p$. If we denote by $\tilde{P}$ a doubly stochastic matrix solution of (3.14) or (3.15), then a common method to project matrix to a permutation space is through the following optimization (Zaslavskiy et al., 2009, Section 2.1):

$$\arg\min_{P \in \mathcal{P}_p} \|\tilde{P} - P\|_F^2 = \arg\max_{P \in \mathcal{P}_p} tr\{\tilde{P}^t P\}, \tag{3.20}$$

which is a linear assignment problem and usually solved by the Hungarian algorithm (Burkard et al., 2012, Section 4.2.1), which takes $O(p^3)$ time.

However, (3.20) suffers a serious drawback since it only delivers one candidate solution to (3.14) or (3.15), and if it is not "close" to the true permutation matrix P, it is unclear how to continue (Wolstenholme and Walden, 2016, Section 3). For this class of problems, as an alternative, the literature suggests a permutation sampling procedure initially proposed for the orthogonal matrices in Barvinok (2005). The idea is to "round" an orthogonal matrix Q to a permutation matrix P by considering its action on the random vector $x \in R^n$ sampled from a Gaussian distribution. Consider a sample $x \in R^p$ from a Gaussian distribution and an ordering vector $r(x)$ such that $r(x)_i = k$ where x_i is the kth smallest value of x. For example, if $x = [4.7, -2.1, 2.5]^t \Rightarrow r(x) = [3, 1, 2]^t$. Barvonik argues, if the permutation matrix P satisfies

$$P(r(x)) = r(Qx) \tag{3.21}$$

then it is "close" in Frobinius norm to Q with respect to x, as they both act on x in a similar way (Barvinok, 2005, Theorem 1.6). In other words, P matches the kth smallest coordinate of x with the kth smallest coordinate of Qx, and P represents a "rounding" of Q. This provides a framework to project an orthogonal matrix to a distribution of permutation matrices.

The proof of Barvinok (2005, Theorem 1.6) reveals that the argument is not restricted to orthogonal matrices and successfully extends to doubly stochastic matrices (Wolstenholme and Walden, 2016, Section 4A). We use (3.21), selecting a doubly stochastic matrix $\tilde{P}$ instead of Q, to generate N permutation matrices each "close" to the doubly stochastic matrix $\tilde{P}$. Then a usual way to select the "best" permutation matrix from the N sampled matrices is to pick a matrix that provides the lowest cost to (3.13) (Fogel et al., 2013, Section 3.2.4).

Finally, Algorithm 4 combines necessary steps to estimate a permutation matrix P in line 5: estimation of the doubly stochastic matrix (3.15), and its approximation to the "closest" permutation matrix via (3.21).

Algorithm 4 Optimization over permutation matrices

```
1: input:
2: $N_{max} \leftarrow$ max. number of sampling
3: Find $\tilde{P}$ via Algorithm 3
4: if $\tilde{P} \notin \mathcal{P}_p$:
5:     while $j < N_{max}$:
6:         Sample: $x^{(j)} \sim N(0, \mathbf{I}_p)\}$
7:         Solve for $P^{(j)}$ using (3.21):
8: From $\{P^{(j)}\}_{j=1}^{N_{max}}$ choose $P$ that minimizes (3.13) .
9: Output: Permutation Matrix $P$
```

3.4.2 Cholesky Factor Estimation

In this section, we focus on estimating a Cholesky factor L from line 6 of Algorithm 2. We fix a permutation matrix P and update the Cholesky factor L using a **non-convex** objective function (3.10). Recall that a Cholesky factor L entails the DAG structure, and by learning L, accordingly, we learn the DAG structure in B.

Recently, there has been active research on estimating the sparse Cholesky factor L with the known order of variables, using various penalty forms and a convex objective function. For example, Shojaie and Michailidis (2010) and convex sparse Cholesky selection (CSCS) algorithm proposed in Khare et al. (2019) achieve sparsity through the lasso penalty. Yu and Bien (2017) used a structured penalty form, which also guarantees sparsity of the precision matrix. In a different approach, the SC algorithm, proposed in Dallakyan and Pourahmadi (2020), estimates matrix L focusing on capturing the smoothness of subdiagonals instead.

We propose a cyclic coordinatewise algorithm to estimate the Cholesky factor L for a fixed permutation matrix P. We show that the non-convex objective function of RRCF (3.10), can be decoupled into p parallel penalized regression problems. From (3.10) and denoting $S^P = PSP^t$, S_i^P the $i \times i$ sub-matrix of S^p, $L_{i\cdot}$ the ith row of L, and β^i non-zero values of the $L_{i\cdot}$ it follows

$$\begin{aligned} Q_{RRCF}(L) &= tr(LS^pL^t) - 2\sum_{i=1}^{p} \log L_{ii} \\ &+ \sum_{1\le j<i\le p} \rho(|L_{ij}|, \lambda) = \sum_{i=1}^{p} (\beta^i)^t S_i^P \beta^i - 2\sum_{i=1}^{p} \log(\beta^i_i) \\ &+ \sum_{i=2}^{p}\sum_{j=1}^{i-1} \rho(|\beta^i_j|, \lambda) = \sum_{i=1}^{p} Q_{RRCF,i}(\beta^i), \end{aligned} \tag{3.22}$$

where in arguments of $Q_{RRCF}(\cdot)$ we omit the dependence from P, and

$$\begin{aligned} Q_{RRCF,i}(\beta^i) &= (\beta^i)^t S_i^P \beta^i - 2\log \beta^i_i \\ &+ \sum_{j=1}^{i-1} \rho(|\beta^i_j|, \lambda) \end{aligned} \tag{3.23}$$

for $2 \leq i \leq p$, and

$$Q_{RRCF,1}(L_{11}) = L_{11}^2 S_{11}^P - 2\log L_{11} \tag{3.24}$$

Here, as in Ye et al. (2020), we focus on the class of penalties called the minimax concave penalty (MCP) proposed in Zhang (2010). The MCP satisfies condition stated in Appendix B.2 and utilizes convexity of the penalized loss near the sparse regions and turns concave outside. It includes ℓ_1 and ℓ_0 as extreme cases. The MCP with two parameters (γ, λ) is given

$$\rho(\theta, \lambda, \gamma) = \begin{cases} \lambda|\theta| - \frac{\theta^2}{2\gamma} & |\theta| < \gamma\lambda, \\ \frac{1}{2}\gamma\lambda^2 & |\theta| \geq \gamma\lambda, \end{cases} \tag{3.25}$$

where $\lambda \geq 0$ and $\gamma > 1$.

Next, we derive steps to minimize the score function $Q_{RRCF}(L)$ with respect to non-zero values of L for the fixed P. We assume that diagonal entries of the sample covariance matrix S are strictly positive. Since $\{\beta^i\}_{i=1}^p$ disjointly partition the parameters in L, then optimizing $Q_{RRCF}(L)$ can be implemented by separately optimizing $Q_{RRCF}(\beta^i)$ for $1 \leq i \leq p$.

We define a generic function $h : R^{k-1} \times R_+ \to R$ of the form

$$h_{k,A,\lambda,\gamma}(x) = -2\log x_k + x^t A x + \sum_{i=1}^{k-1} \rho(|x_i|, \lambda, \gamma), \tag{3.26}$$

where $\lambda > 0$, $\gamma > 1$ and A is a positive semi-definite matrix with positive diagonal entries. It is instructive note that $Q_{RRCF,i}(\beta^i) = h_{i,S_i,\lambda,\gamma}(\beta^i)$ for every $1 \leq i \leq p$, and it suffices to develop an algorithm which minimizes a function of the form $h_{i,A,\lambda,\gamma}$. For every $1 \leq j \leq k$, we define

$$x_j^* = \inf_{x_j} h_{k,A,\lambda,\gamma}(x).$$

Next lemma shows that $\{x_j^*\}_{j=1}^k$ can be computed in the closed form. The proof is provided in Appendix.

Lemma 9. *The optimal solution* $\{x_j^*\}_{j=1}^k$ *can be computed in the closed form.*

$$x_k^* = \frac{-\sum_{l \neq k} A_{lk} x_l + \sqrt{(\sum_{l \neq k} A_{lk} x_l)^2 + 4A_{kk}}}{2A_{kk}} \tag{3.27}$$

and for $1 \leq j \leq k-1$,

$$x_j^* = \frac{S_\lambda(-2\sum_{l \neq k} A_{lk} x_l)}{2A_{jj} - 1/\gamma} \tag{3.28}$$

Here, S_λ is the soft-thresholding operator given by $S_\lambda(x) = sign(x)(|x| - \lambda)_+$. Using Lemma 9, we can construct a cyclic coordinatewise minimization algorithm for $h_{k,A,\lambda,\gamma}$. We use Algorithm 5 to minimize $Q_{RRCF}(\beta^i)$ for $1 \leq i \leq p$, and combine outputs to obtain the estimated Cholesky factor L in Algorithm 6.

Algorithm 5 Cyclic coordinatewise algorithm

```
input:
$k_{max}, A, \lambda, \gamma, \epsilon$
$x^{(0)} \leftarrow$ Initial estimate
Set $x^{current} = x^{(0)}$; Converged = FALSE
while Converged == FALSE or $k < k_{max}$ :
    $x^{old} \leftarrow x^{current}$
    For j = 1,2,..., k - 1
        $x_j^{current} = x_j^*$ via (3.28)
    $x_k^{current} = x_k^*$ via (3.27)
    if $\|x^{current} - x^{old}\| < \epsilon$
        Converged = TRUE
    else   $k = k + 1$
Output: $x$
```

3.4.2.1 Convergence of the Cyclic Coordintewise Algorithm

As discussed, the score function $Q_{RRCF}(L)$ is non-convex with respect to L, and the convergence of iterates in Algorithm 6 is guaranteed only to a local minimum. We begin by showing that for fixed permutation matrix P, the objective function $Q_{RRCF}(L)$ is lower bounded, a local

Algorithm 6 Cholesky Factor Estimation

```
*input*:
$k_{max}, \mathbf{X}, \lambda, \gamma, \epsilon$
$L^{(0)} \leftarrow$ *Initial Cholesky factor*
*For* i = 1,2,..., p
    $\beta^i = \arg\min_{\beta_i} Q_{RRCF,i}(\beta^i)$ *via Algorithm 5*
*Construct* $L \in \mathcal{L}_p$ *by setting its non-zero values as* $\beta^i$
*Output*: *Lower diagonal matrix* $L$
```

minimum lies in the space of lower triangular matrices $\mathcal{L}_p$ with positive diagonal entries, and for certain values of γ, the generic function $h(\cdot)$ is strictly convex.

Lemma 10. *a. If* $A_{ii} > 0$*, for* $1 \leq i \leq p$

$$h_{k,A,\lambda,\gamma}(x) \geq 2x_k - 2.$$

b. For $\gamma > \max\{1/2A_{ii}, 1\}$*,* $h_{k,A,\lambda,\gamma}(x)$ *is a strictly convex function of* x_i *for* $1 \leq i \leq k-1$*.*

c. For every n *and* p

$$\inf_{L \in \mathcal{L}_p} Q_{RRCF}(L) = \sum_{i=1}^{p} \inf_{\beta^i} Q_{RRCF,i}(\beta^i) \geq -2p > -\infty$$

and any local minimum of Q_{RRCF} *over the open set* $\mathcal{L}_p$ *lies in* $\mathcal{L}_p$*.*

From this lemma, we can establish the convergence of the cyclic coordintewise algorithm.

Theorem 3. *Under the Assumptions of Lemma 10, Algorithm 6 converges to a local minimum of* $Q_{RRCF}(L)$*.*

3.5 Simulation and Data Analysis

In this section, we assay the empirical performance of our estimator on simulated and macroeconomic datasets.

3.5.1 Simulation Study

For comparison, we include three other BN learning algorithms: **ARCS**:(Ye et al., 2020), **CCDr**:(Aragam and Zhou, 2015), and **RRCF-L** (this method differs from our proposed method only by having lasso penalty $\lambda \sum_{j<i} |L_{ij}|$ in (3.11) instead of MCP penalty).

In all simulations, the sample size $n = 150$ and each sample follows p-dimensional Gaussian distribution $N(0, (L^t L)^{-1})$. We compare the performance of our method with the three algorithms above, both in terms of the structure learning, and how well the weighted adjacency matrix $\hat{B}$ estimates B. The weighted adjacency matrix B is constructed following Kalisch and Bühlmann (2007, Section 4.1) framework. We adapt parameterization $L = (I - B)\Omega^{-1/2}$ to generate data where we choose $\Omega = I_p$. We consider $p \in \{100, 200\}$ and expected sparsity level $s \in \{p, 2p\}$ which corresponds to the expected number of edges in the DAG. Each of the simulation settings (p, s) is repeated over 20 datasets. The optimization parameters η, μ for the RRCF algorithm are chosen according to Lemmas 6 and 8, $\alpha = 1$, and for the tuning parameter selection, we rely on the extended BIC criterion (Foygel and Drton, 2010) (See Appendix B.3 for details).

3.5.1.1 Structure Learning and Estimation Accuracy

We compare the four algorithms using the following four metrics: True Positive Rate (TPR), False Positive Rate (FPR), Structural Hamming Distance (SHD), and scaled Frobenius norm, which estimates how far is weighted adjacency matrix $\hat{B}$ from B; i.e., $\frac{1}{p}\|\hat{B} - B\|_F$.

Table 3.1 and Figure 3.3 report the simulation results based on the above four metrics. The best average score for each metrics and (p, s) setting are highlighted in bold. Results suggest that RRCF performance improves when s is higher for fixed p. In particular, for the $(100, 100)$ case, CCDr provides the best results for the TPR metric, followed by RRCF and ARCS, which provide similar output. For the FPR metric, RRCF, CCDr, and ARCS provide identical results. The situation changes for the $(100, 200)$ case, where RRCF provides the best TPR average score, and ARCS provides the best FPR average score. A similar pattern follows when we increase the dimension from 100 to 200. RRCF_L provides the best scaled Frobenius norm result for all four

Table 3.1: Average of three metrics over 20 replication for four (p, m) settings.

(p, s)	Method	TPR	FPR	FRB. NORM
(100,100)	RRCF_L	0.569	0.002	**6.238**
	RRCF	0.600	**0.001**	6.868
	CCDr	**0.621**	**0.001**	9.930
	ARCS	0.601	**0.001**	7.052
(100,200)	RRCF_L	0.566	0.011	**8.989**
	RRCF	**0.652**	0.009	10.299
	CCDr	0.615	0.005	14.305
	ARCS	0.635	**0.004**	10.569
(200,200)	RRCF_L	0.589	0.003	**11.119**
	RRCF	0.622	0.003	12.509
	CCDr	**0.657**	0.007	19.845
	ARCS	0.641	**0.001**	12.883
(200,400)	RRCF_L	0.600	0.002	**10.617**
	RRCF	**0.653**	**0.001**	11.752
	CCDr	0.658	**0.001**	17.679
	ARCS	0.655	**0.001**	12.043

(p, m) settings.

From Figure 3.3, for the $(100, 100)$ case, the performance of RRCF is compatible with the other algorithms. However, its performance, with respect to the SHD metric, deteriorates for the other settings. ARCS and CCDr provide similar SHD scores for all four (p, s) settings.

3.5.2 Additional Simulation Results

Finally, we compare RRCF, RRCF_L, and ARCS performance using the receiver operating characteristic (ROC) curve for the case $(200, 400)$. ROC compares the true positive rate (TPR) and the false positive rate (FPR). Curves closer to the upper-left corner indicate a better performance.

We fix the tuning parameter for RRCF and ARCS at $\gamma = 2$ and use a grid of 40 λ values for comparison. Figure 3.4 shows the ROC curve for each estimator. From it follows that RRCF is superior compare to the other algorithms.

3.5.3 Macro-Economic Application

We illustrate the practical use of RRCF by applying it to the macro-economic dataset. In particular, we utilize our methodology to estimate the order of shocks (impulses) in the structural

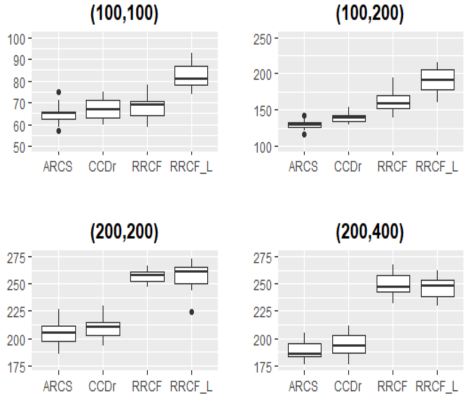


Figure 3.3: Structural Hamming Distance boxplot for four (p, s) settings.

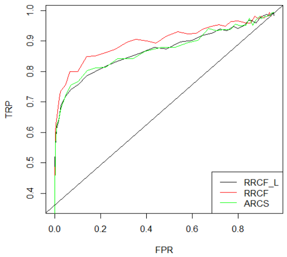


Figure 3.4: ROC curve over the grid of λ values.

vector autoregression (SVAR) model. Impulse response functions are one of the main techniques employed to analyze the dynamics in SVAR models (Lütkepohl, 2007, Section 2.3.2). They are used to discover the future effects of a shock on variables.

The estimation of the impulse response function requires knowledge of the ordering among contemporaneous error terms. There is growing literature on the use of DAGs for recovering such an order. For example, see Swanson and Granger (1997); Bessler and Akleman (1998); Demiralp and Hoover (2003) for Gaussian data and Dallakyan (2020) for non-Gaussian data. Algorithm 1 in Dallakyan (2020) provides details on the use of DAGs for the SVAR estimation. We iterate it in Algorithm 7 by incorporating the RRCF step in line 7 to recover the ordering of error terms.

Algorithm 7 SVAR procedure with RRCF

```
 1: procedure
 2: input:
 3:     $y_1, \dots, y_t \leftarrow K$ dimensional stationary series
 4: top:
 5:     Estimate the VAR model $y_t = A_1 y_{t-1} + \cdots + A_p y_{t-p} + u_t$,
 6:     Estimate the residuals $\hat{u}_t = y_t - \hat{A}_1 y_{t-1} - \cdots - \hat{A}_p y_{t-p}$
 7:     Perform  RRCF algorithm on residuals to recover ordering among residuals $\hat{u}_{1t}, \dots, \hat{u}_{Kt}$.
 8:     Estimate  matrix $K$ of structural coefficients by maximizing likelihood function (3.29)
 9: Output:
10:     $B$
```

The log-likelihood in line 8 of the Algorithm 7 is given by (Lütkepohl, 2007, Chapter 9.3.1)

$$\ln l_c(\mathsf{K}) = \text{constant} + \frac{T}{2}\ln|K|^2 - \frac{T}{2} tr\{K^{-T}K^{-1}\tilde{\Sigma}_u\}, \tag{3.29}$$

where $\tilde{\Sigma}_u = T^{-1}(Y - \hat{A}X)(Y - \hat{A}X)^t$, T is the length of series and matrix K contains structural coefficients for the impulse response function.

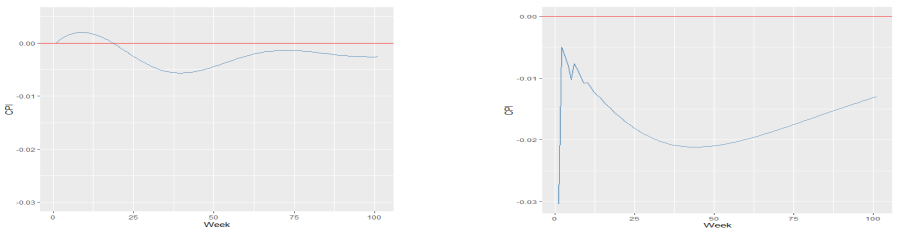
We use RRCF incorporated Algorithm 7 to solve the price puzzle. The price puzzle in a structural autoregression (SVAR) system is known as an inability to explain the positive relationship between an innovation(shock) in the federal funds rate (FFR) and inflation (Bernanke and Blinder, 1992; Sims, 1992; Balke and Emery, 1994; Demiralp et al., 2014). It is a puzzle since an increase in the federal funds is expected to be followed by a decrease in the price level rather than an increase (See Figure 3.5a).

Dallakyan (2020) showed that utilization of the recent DAG techniques to recover the ordering of error terms in VAR mitigates the price puzzle problem. Here, we show that using the sparse VAR approach and RRCF algorithm to recover the DAG structure of error terms leads to the complete disappearance of the price puzzle.

To analyze the price puzzle, we use a relatively rich dataset from Demiralp et al. (2014). Data consist of 12 monthly series for the United States that run from 1959:02 to 2007:06. Data sources and details are provided in Demiralp et al. (2014). In the dataset, monetary policy is represented both by the Federal funds rate (FFR) and two reserve components: (the logarithms of) borrowed reserves (BORRES) and nonborrowed reserves (NBORRES). Financial markets are represented by two monetary aggregates (the logarithms of) M1 and (the non-M1 components of) M, as well as by three interest rates: the own-rate of interest on M2 (M2OWN), the 3-month Treasury bill rate (R3M), and the 10-year Treasury bond rate (R10Y). Prices are represented by (the logarithms of) the consumer price index (CPI) and an index of sensitive commodity prices (COMPRICE). Finally, the real economy is represented by the (logarithm of) industrial production (INDPRO) and the output gap (GAP). Our sample period runs from January 1990 until 2009. The sample period is chosen such that to avoid a policy break (Demiralp et al., 2014).

To introduce sparsity in the VAR estimation, we impose a lasso penalty on the VAR coefficient matrix (Song and Bickel, 2011; Nicholson et al., 2016). For the sparse VAR estimation, we use the `BigVAR` package in `R` (Nicholson et al., 2017), with the number of lags equal to 4. Then, using contemporaneous time restrictions obtained from Algorithm 7, we estimate impulse response functions. For comparison, we include the impulse response function obtained from the PC algorithm (Bessler and Akleman, 1998).

Figure 3.5 plots the responses of Consumer Price Index (CPI) to Federal Fund Rate (FFR) obtained from the PC and RRCF algorithms, respectively. From Figure 3.5a, the prize puzzle is apparent when the response is estimated using the PC algorithm. However, it disappears when the response is estimated using the RRCF algorithm (see Figure 3.5b). The latter result is consistent with the macro-economic literature.



(a) PC

(b) RRCF

Figure 3.5: The response of Consumer Price Index to Federal Fund Rate shock.

3.6 Statistical Properties

In this section, we study the consistency of the RRCF estimator, assuming that the true permutation matrix P is known; i.e., data have known order. Under this assumption, we can omit the dependence of Q_{RRCF} on P and focus only on the consistency of a Cholesky factor estimator in (3.10).

Khare et al. (2019); Yu and Bien (2017) provide consistency of the sparse Cholesky factor estimator for the convex objective function. However, our objective function is *non-convex* and it may possess multiple local optima that are not global. Therefore, the standard statistical techniques are not applicable for establishing consistency.

We establish upper bounds on the Frobenius norm between **any local optimum** of the empirical estimator and the unique minimizer of the population. Even though the non-convex function may possess multiple local optima, our theoretical results guarantee that, from a statistical perspective, all local optima are fundamentally as good as a global optimum. The theoretical analysis relies on the following assumptions:

- A1 *Marginal sub-Gaussian assumption:* The sample matrix $X \in \mathcal{R}^{n \times p}$ has n independent rows with each row drawn from the distribution of a zero-mean random vector $X = (X_1, \ldots, X_p)^t$ with covariance Σ and sub-Gaussian marginals; i.e.,

$$E[\exp(tX_j/\sqrt{\Sigma_{jj}})] \leq \exp(Ct^2)$$

  for all $j = 1, \ldots, p$, $t \leq 0$ and for some constant $C > 0$.

- A2 *Sparsity Assumption:* The true Cholesky factor $L \in \mathcal{R}^{p \times p}$ is the lower triangular matrix with positive diagonal elements and support $\mathcal{S}(L) = \{(i,j), i \neq j | L_{ij} \neq 0\}$. We denote by $s = |S|$ cardinality of the set S.

- A3 *Bounded eigenvalues:* There exist a constant κ such that

$$0 < \kappa^{-1} \leq \lambda_{min}(L) \leq \lambda_{max}(L) \leq \kappa$$

Before providing our main result, we recall that a matrix $\hat{L} \in \mathcal{L}_p$ is a stationary point for Q_{RRCF} if it satisfies (Bertsekas, 2015)

$$\langle \nabla \mathcal{L}_n(\hat{L}) + \nabla \rho(\hat{L}, \lambda), L - \hat{L} \rangle \geq 0, \text{ for } L \in \mathcal{L}_p, \tag{3.30}$$

where $\mathcal{L}_n(L) = \text{tr}(SLL^t) - 2\log|L|$ and $\nabla \rho(\cdot, \cdot)$ is the subgradient.

Theorem 4. *Under Assumptions A1-A3, with tuning parameter λ of scale $\sqrt{\frac{\log p}{n}}$, and $\frac{3}{4\gamma} < (\kappa + 1)^{-2}$, the scaling $(s+p)\log p = o(n)$ is sufficient for any stationary point $\hat{L}$ of the non-convex program Q_{RRCF} to satisfy the following estimation bounds:*

$$\|\hat{L} - L\|_F = \mathcal{O}_p\Big(\sqrt{\frac{(s+p)\log p}{n}}\Big)$$
$$\|\hat{\Sigma}^{-1} - \Sigma^{-1}\|_F = \mathcal{O}_p\Big(\sqrt{\frac{(s+p)\log p}{n}}\Big)$$

The proof is provided in Appendix.

4. CONCLUSION

Cholesky decomposition has diverse applications in the statistical discipline, including least squares, time series, and statistical/machine learning. Motivated by these applications, the Cholesky factor has been the main subject of our research. In this dissertation, we have proposed methods to learn inverse covariance matrix and Bayesian Network exploiting the Cholesky factor.

In Chapter 2, we developed a block coordinate descent algorithm to estimate the inverse covariance matrix from longitudinal data by imposing smoothness assumption on subdiagonals of the Cholesky factor. The algorithm iteratively updates subdiagnals of the Cholesky factor until convergence. Reliance on the Cholesky factor, as the new parameter within a regularized likelihood setup, guarantees: joint convexity of the likelihood function, strict convexity of the likelihood function restricted to each subdiagonal even when $n < p$, and positive-definiteness of the estimated covariance matrix.

In Chapter 3, we eliminate the assumption of known order and learn the Gaussian Bayesian Network through the two-step procedure. In the first step, we impose relaxation to find the permutation matrix, and then for a given ordering, we estimate a (sparse) Cholesky factor by decoupling row-wise. Introduced relaxation avoids the hard combinatorial problem of order estimation and enables learning DAGs without a need to verify expensive acyclicity constraints.

Further Study

As with most research projects, detailed investigation and exploration spawned more ideas with which to approach the problem. For instance, in Section 2.3, we discussed the close relationship between the smoothness of the Cholesky factor subdiagonals and the (inverse) covariance matrix. One of the further extension of our work of Chapter 2 can be an algorithm which directly learns the smoothness of the (inverse) covariance matrix. The other follow-up work that deserves investigation is the statistical properties of our SC estimator.

The statistical properties for the RRCF algorithm, derived in Chapter 3, assume the knowledge of the permutation matrix. Future work can investigate the theoretical properties by eliminating the stringent, fixed permutation matrix assumption.

REFERENCES

Adak, Sudeshna (1998), "Time-dependent spectral analysis of nonstationary time series." *Journal of the American Statistical Association*, 93, 1488–1501.

Anderson, T. (2003), *An Introduction to Multivariate Statistics*. Wiley, New York.

Ansley, Craig F. (1979), "An algorithm for the exact likelihood of a mixed autoregressive-moving average process." *Biometrika*, 66, 59–65.

Aragam, Bryon and Qing Zhou (2015), "Concave penalized estimation of sparse gaussian bayesian networks." *J. Mach. Learn. Res.*, 16, 2273–2328.

Balke, Nathan S. and Kenneth M. Emery (1994), "Understanding the price puzzle." *Economic and Financial Policy Review*, 15–26.

Banerjee, Onureena, Laurent El Ghaoui, and Alexandre d'Aspremont (2008), "Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data." *J. Mach. Learn. Res.*, 9, 485–516.

Bartlett, M. S. (1933), "On the theory of statistical regression." *Proc. R. Soc. Edinburgh*, 53, 260–283.

Barvinok, Alexander I. (2005), "Approximating orthogonal matrices by permutation matrices." *Arxiv preprint arXiv:math/0510612*.

Beck, Amir and Luba Tetruashvili (2013), "On the convergence of block coordinate descent type methods." *SIAM Journal on Optimization*, 23, 2037–2060.

Beinlich, Ingo A., H. J. Suermondt, R. Martin Chavez, and Gregory F. Cooper (1989), "The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks." In *AIME 89*, 247–256, Springer Berlin Heidelberg.

Bernanke, Ben S and Alan S Blinder (1992), "The Federal Funds Rate and the Channels of Monetary Transmission." *American Economic Review*, 82, 901–921.

Bertsekas, Dimitri P. (2015), *Convex Optimization Algorithms*. Athena Scientific.

Bertsekas, D.P. (2016), *Nonlinear Programming*. Athena Scientific.

Bessler, D. and D. Akleman (1998), “Farm prices,retail prices, and directed graphs: Results for pork and beef.” *American journal of Agricultural Economics.*, 1144–1149.

Bickel, Peter J. and Yulia R. Gel (2011), “Banded regularization of autocovariance matrices in application to parameter estimation and forecasting of time series.” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73, 711–728.

Blake, Tayler (2018), *Nonparametric Covariance Estimation with Shrinkage toward Stationary Models*. Ph.D. thesis, The Ohio State University.

Boyd, Stephen and Lieven Vandenberghe (2004), *Convex Optimization*. Cambridge University Press, New York, USA.

Brezinski, Claude (2006), “The life and work of Andrè Cholesky.” *Numerical Algorithms*, 43, 279–288.

Brualdi, Richard A and Peter M Gibson (1977), “Convex polyhedra of doubly stochastic matrices. i. applications of the permanent function.” *Journal of Combinatorial Theory, Series A*, 22, 194 – 230.

Burkard, Rainer, Mauro Dell’Amico, and Silvano Martello (2012), *Assignment Problems. Revised reprint.* SIAM - Society of Industrial and Applied Mathematics.

Cai, Tony, Weidong Liu, and Xi Luo (2011), “A constrained l1 minimization approach to sparse precision matrix estimation.” *Journal of the American Statistical Association*, 106, 594–607.

Chen, Wenyu, Mathias Drton, and Y Samuel Wang (2019), “On causal discovery with an equal-variance assumption.” *Biometrika*, 106, 973–980.

Chern, Jann-Long and Luca Dieci (2000), “Smoothness and periodicity of some matrix decompositions.” *SIAM J. Matrix Analysis Applications*, 22, 772–792.

Chickering, David Maxwell (2002), "Optimal structure identification with greedy search." *J. Mach. Learn. Res.*, 3, 507–554.

Chickering, David Maxwell (2003), "Optimal structure identification with greedy search." *J. Mach. Learn. Res.*, 3, 507–554.

Chickering, David Maxwell, David Heckerman, and Christopher Meek (2004), "Large-sample learning of bayesian networks is np-hard." *J. Mach. Learn. Res.*, 5, 1287–1330.

Chow, C. and C. Liu (1968), "Approximating discrete probability distributions with dependence trees." *IEEE Transactions on Information Theory*, 14, 462–467.

Cooper, Gregory F. and Edward Herskovits (1992), "A bayesian method for the induction of probabilistic networks from data." *Mach. Learn.*, 9, 309–347.

Dahlhaus, R. (1997), "Fitting time series models to nonstationary processes." *Ann. Statist.*, 25, 1–37.

Dahlhaus, Rainer (2012), "Locally stationary processes." *Handbook of Statistics*, 30, 351–413.

Dahlhaus, Rainer and Wolfgang Polonik (2009), "Empirical spectral processes for locally stationary time series." *Bernoulli*, 15, 1–39.

Dai, Ming and Wensheng Guo (2004), "Multivariate spectral analysis using cholesky decomposition." *Biometrika*, 91, 629–643.

Dallakyan, Aramayis (2019), "Sc package." `https://github.com/adallak/SCPackage`.

Dallakyan, Aramayis (2020), "Nonparanormal Structural VAR for Non-Gaussian Data." *Computational Economics*, 0, 1–21.

Dallakyan, Aramayis and Mohsen Pourahmadi (2020), "Fused-lasso regularized cholesky factors of large nonstationary covariance matrices of longitudinal data." *Arxiv preprint arXiv:2007.11168*.

Das, Srinjoy and Dimitris N. Politis (2020), “Predictive inference for locally stationary time series with an application to climate data.” *Journal of the American Statistical Association*, 0, 1–16.

Davis, A Richard, C. M Thomas Lee, and Rodriguez-Yam A Gabriel (2006), “Structural break estimation for nonstationary time series models.” *Journal of the American Statistical Association*, 101, 223–239.

Demiralp, Selva, Kevin Hoover, and Stephen Perez (2014), “Still puzzling: evaluating the price puzzle in an empirically identified structural vector autoregression.” *Empirical Economics*, 46, 701–731.

Demiralp, Selva and Kevin D. Hoover (2003), “Searching for the causal structure of a vector autoregression.” *Oxford Bulletin of Economics and Statistics*, 65, 745–767.

Ding, X. and Z. Zhou (2019), “Globally optimal and adaptive short-term forecast of locally stationary time series and a test for its stability.” *Arxiv preprint arXiv:1912.12937*.

Ding, Xiucai and Z. Zhou (2018), “Estimation and inference for precision matrices of nonstationary time series.” *Arxiv preprint arXiv:1803.01188*.

Fan, Jianqing and Runze Li (2001), “Variable selection via nonconcave penalized likelihood and its oracle properties.” *Journal of the American Statistical Association*, 96, 1348–1360.

Fogel, Fajwel, Rodolphe Jenatton, Francis Bach, and Alexandre D’Aspremont (2013), “Convex relaxations for permutation problems.” In *Advances in Neural Information Processing Systems 26*, 1016–1024.

Foygel, Rina and Mathias Drton (2010), “Extended bayesian information criteria for gaussian graphical models.” In *Advances in Neural Information Processing Systems 23*, 604–612.

Friedman, H. Jerome, J. Trevor Hastie, and Robert Tibshirani (2010), “Applications of the lasso and grouped lasso to the estimation of sparse graphical models.” Available at `http://statweb.stanford.edu/~tibs/ftp/ggraph.pdf`.

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

Friedman, Jerome, Trevor Hastie, Holger Höfling, and Robert Tibshirani (2007), “Pathwise coordinate optimization.” *Ann. Appl. Stat.*, 1, 302–332.

Gabriel, K. R. (1962), “Ante-dependence analysis of an ordered set of variables.” *Ann. Math. Statist.*, 33, 201–212.

Ghoshal, Asish and Jean Honorio (2018), “Learning linear structural equation models in polynomial time and sample complexity.” In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, 1466–1475.

Golub, Gene H. and Charles F. Van Loan (1996), *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA.

Grady, Noella (2009), “Functions of bounded variation.” URL `https://www.whitman.edu/Documents/Academics/Mathematics/grady.pdf`.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2001), *The Elements of Statistical Learning*. Springer Series in Statistics, Springer New York Inc., New York, NY, USA.

Heckerman, David, Dan Geiger, and David M. Chickering (1995), “Learning bayesian networks: The combination of knowledge and statistical data.” *Mach. Learn.*, 20, 197–243.

Hemmecke, Raymond, Silvia Lindner, and Milan Studený (2012), “Characteristic imsets for learning bayesian network structure.” *International Journal of Approximate Reasoning*, 53, 1336 – 1349. Fifth European Workshop on Probabilistic Graphical Models (PGM-2010).

Hodrick, Robert J. and Edward Prescott (1997), “Postwar u.s business cycles: An empirical investigation.” *Journal of Money, Credit and Banking*, 29.

Horn, Roger A. and Charles R. Johnson (2012), *Matrix Analysis*, 2nd edition. Cambridge University Press, New York, NY, USA.

Huang, J, N Liu, M Pourahmadi, and L. Liu (2006), "Covariance matrix selection and estimation via penalised normal likelihood." *Biometrika*, 93, 85–98.

Huang, Z Jianhua, Linxu Liu, and Naiping Liu (2007), "Estimation of large covariance matrices of longitudinal data with basis function approximations." *Journal of Computational and Graphical Statistics*, 16, 189–209.

Inoue, Akihiko and Yukio Kasahara (2006), "Explicit representation of finite predictor coefficients and its applications." *Ann. Statist.*, 34, 973–993.

Jaakkola, Tommi, David Sontag, Amir Globerson, and Marina Meila (2010), "Learning bayesian network structure using lp relaxations." In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, 358–365.

Kalisch, Markus and Peter Bühlmann (2007), "Estimating high-dimensional directed acyclic graphs with the pc-algorithm." *Journal of Machine Learning Research*, 8, 613–636.

Kenward, Michael G. (1987), "A method for comparing profiles of repeated measurements." *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 36, 296–308.

Khare, Kshitij, Sang-Yun Oh, Syed Rahman, and Bala Rajaratnam (2019), "A scalable sparse cholesky based approach for learning high-dimensional covariance matrices in ordered data." *Machine Learning*, 108, 2061–2086.

Khare, Kshitij, Sang-Yun Oh, and Bala Rajaratnam (2015), "A convex pseudolikelihood framework for high dimensional partial correlation estimation with convergence guarantees." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77, 803–825.

Khare, Kshitij and Bala Rajaratnam (2014), "Convergence of cyclic coordinatewise l1 minimization." *arXiv e-prints*. Available at `https://arxiv.org/pdf/1404.5100.pdf`.

Kim, Seung-Jean, Kwangmoo Koh, Stephen P Boyd, and Dimitry M. Gorinevsky (2009), "l1 trend filtering." *SIAM Review*, 51, 339–360.

Kitagawa, G. and W. Gersch (1985), “A smoothness priors time-varying ar coefficient modeling of nonstationary covariance time series.” *IEEE Transactions on Automatic Control*, 30, 48–56.

Koivisto, Mikko (2006), “Advances in exact bayesian structure discovery in bayesian networks.” In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, 241–248, AUAI Press, Arlington, Virginia, USA.

Koller, Daphne and Nir Friedman (2009), *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press.

Levina, Elizaveta, Adam Rothman, and Ji Zhu (2008), “Sparse estimation of large covariance matrices via a nested lasso penalty.” *The Annals of Applied Statistics*, 2, 245–263.

Loh, Po-Ling and Peter Bühlmann (2014), “High-dimensional learning of linear causal networks via inverse covariance estimation.” *J. Mach. Learn. Res.*, 15, 3065–3105.

Loh, Po-Ling and Martin J. Wainwright (2015), “Regularized m-estimators with nonconvexity: Statistical and algorithmic theory for local optima.” *J. Mach. Learn. Res.*, 16, 559–616.

Luo, Z. Q. and P. Tseng (1992), “On the convergence of the coordinate descent method for convex differentiable minimization.” *Journal of Optimization Theory and Applications*, 72, 7–35.

Lütkepohl, Helmut (2007), *New Introduction to Multiple Time Series Analysis*. Springer, New York.

Magnus, Jan R. and H. Neudecker (1986), “Symmetry, 0-1 matrices and jacobians: A review.” *Econometric Theory*, 2, 157–190.

Marcus, Marvin and Henrik Minc (1962), “Some results on doubly stochastic matrices.” *Proceedings of the American Mathematical Society*, 13, 571–579.

McMurry, Timothy L. and Dimitris N. Politis (2010), “Banded and tapered estimates for autocovariance matrices and the linear process bootstrap.” *Journal of Time Series Analysis*, 31, 471–482.

McMurry, Timothy L. and Dimitris N. Politis (2015), “High-dimensional autocovariance matrices and optimal linear prediction.” *Electron. J. Statist.*, 9, 753–788.

Meinshausen, Nicolai and Peter Buhlmann (2006), “High-dimensional graphs and variable selection with the lasso.” *Ann. Statist.*, 34, 1436–1462.

Needham, Chris J, James R Bradford, Andrew J Bulpitt, and David R Westhead (2007), “A primer on learning in bayesian networks for computational biology.” *PLOS Computational Biology*, 3, 1–8.

Neil, Martin, Norman Fenton, and Manesh Tailor (2005), “Using bayesian networks to model expected and unexpected operational losses.” *Risk Analysis*, 25, 963–972.

Nicholson, W., D. Matteson, and J. Bien (2017), “BigVAR: Tools for Modeling Sparse High-Dimensional Multivariate Time Series.” *ArXiv e-prints*.

Nicholson, William B., Jacob Bien, and David S. Matteson (2016), “Hierarchical vector autoregression.” *Arxiv preprint arXiv:1412.5250v2*.

Olkin, Ingram (1985), “Estimating a cholesky decomposition.” *Linear Algebra and its Applications*, 67, 201 – 205.

Pearl, Judea (2009), *Causality: Models, Reasoning and Inference*, 2nd edition. Cambridge University Press, USA.

Peng, Jie, Pei Wang, Nengfeng Zhou, and Ji Zhu (2009), “Partial correlation estimation by joint sparse regression models.” *Journal of the American Statistical Association*, 104, 735–746.

Peters, J. and P. Bühlmann (2013), “Identifiability of Gaussian structural equation models with equal error variances.” *Biometrika*, 101, 219–228.

Peters, Jonas, Dominik Janzing, and Bernhard Schlkopf (2017), *Elements of Causal Inference: Foundations and Learning Algorithms*. The MIT Press.

Pourahmadi, M. (2001), *Foundations of time series analysis and prediction theory*. John Wiley & Sons, Ltd.

Pourahmadi, Mohsen (1999), “Joint mean-covariance models with applications to longitudinal data: Unconstrained parameterisation.” *Biometrika*, 86, 677–690.

Pourahmadi, Mohsen (2013), *High-Dimensional Covariance Estimation*. John Wiley & Sons, Ltd.

R Core Team (2019), *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, URL `http://www.R-project.org/`.

Rao, T. Subba (1970), “The fitting of non-stationary time-series models with time-dependent parameters.” *Journal of the Royal Statistical Society. Series B (Methodological)*, 32, 312–322.

Robinson, R. W. (1977), “Counting unlabeled acyclic digraphs.” In *Combinatorial Mathematics V* (Charles H. C. Little, ed.), 28–43, Springer Berlin Heidelberg, Berlin, Heidelberg.

Rojas, Cristian R. and Bo Wahlberg (2014), “On change point detection using the fused lasso method.” *arXiv: Statistics Theory*.

Rosen, Ori and David S. Stoffer (2007), “Automatic estimation of multivariate spectra via smoothing splines.” *Biometrika*, 94, 335–345.

Rothman, Adam J., Peter J. Bickel, Elizaveta Levina, and Ji Zhu (2008), “Sparse permutation invariant covariance estimation.” *Electron. J. Statist.*, 2, 494–515.

Rothman, J. Adam, Elizaveta Levina, and Ji Zhu (2010), “A new approach to cholesky-based covariance regularization in high dimensions.” *Biometrika*, 97, 539–550.

Rudin, Leonid I., Stanley Osher, and Emad Fatemi (1992), “Nonlinear total variation based noise removal algorithms.” *Physica D: Nonlinear Phenomena*, 60, 259 – 268.

Shojaie, Ali and George Michailidis (2010), “Penalized likelihood methods for estimation of sparse high-dimensional directed acyclic graphs.” *Biometrika*, 97, 519–538.

Silander, Tomi and Petri Myllymäki (2006), “A simple approach for finding the globally optimal bayesian network structure.” In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, 445–452, AUAI Press, Arlington, Virginia, USA.

Sims, Christopher A. (1992), “Interpreting the Macroeconomic Time Series Facts: The Effects of Monetary Policy.” Technical report.

Song, S. and P. J. Bickel (2011), “Large vector auto regressions.” *Arxiv preprint arXiv:1106.3915*.

Spirtes, Peter and Clark Glymour (1991), “An algorithm for fast recovery of sparse causal graphs.” *Social Science Computer Review*, 9, 62–72.

Studený, Milan and David Haws (2014), “Learning bayesian network structure: Towards the essential graph by integer linear programming tools.” *International Journal of Approximate Reasoning*, 55, 1043 – 1071. Special issue on the sixth European Workshop on Probabilistic Graphical Models.

Swanson, Norman R. and Clive W. J. Granger (1997), “Impulse response functions based on a causal approach to residual orthogonalization in vector autoregressions.” *Journal of the American Statistical Association*, 92, 357–367.

Tao, Terence (2016), *Analysis II*, third edition edition. Singapore : Springer.

Teyssier, Marc and Daphne Koller (2005), “Ordering-based search: A simple and effective algorithm for learning bayesian networks.” In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, 584–590, AUAI Press, Arlington, Virginia, USA.

Tibshirani, Robert (1996), “Regression shrinkage and selection via the lasso.” *Journal of the Royal Statistical Society. Series B (Methodological)*, 58, 267–288.

Tibshirani, Robert, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight (2005), “Sparsity and smoothness via the fused lasso.” *Journal of the Royal Statistical Society*, 67, 91–108.

Tibshirani, Ryan J. and Jonathan Taylor (2011), “The solution path of the generalized lasso.” *Ann. Statist.*, 39, 1335–1371.

Tsamardinos, Ioannis, Laura Brown, and Constantin Aliferis (2006), “The max-min hill-climbing bayesian network structure learning algorithm.” *Machine Learning*, 65, 31–78.

Tseng, P. (2001), "Convergence of a block coordinate descent method for nondifferentiable minimization." *Journal of Optimization Theory and Applications*, 109, 475–494.

van de Geer, Sara and Peter Bühlmann (2013), "ℓ_0-penalized maximum likelihood for sparse directed acyclic graphs." *Ann. Statist.*, 41, 536–567.

Whittaker, J. (1990), *Graphical models in applied multivariate statistics*. John Wiley & Sons, Ltd.

Wolstenholme, R. J. and Andrew T. Walden (2016), "A sampling strategy for projecting to permutations in the graph matching problem." *Arxiv preprint arXiv:1604.04235*.

Wu, Wei Biao and Mohsen Pourahmadi (2003), "Nonparametric estimation of large covariance matrices of longitudinal data." *Biometrika*, 90, 831–844.

Wu, Wei Biao and Mohsen Pourahmadi (2009), "Banding sample autocovariance matrices of stationary processes." *Statistica Sinica*, 19, 1755–1768.

Ye, Q., A. Amini, and Q. Zhou (2020), "Optimizing regularized cholesky score for order-based learning of bayesian networks." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–1.

Yu, Guo and Jacob Bien (2017), "Learning local dependence in ordered data." *Journal of Machine Learning Research*, 18, 1–60.

Yuan, Ming and Yi Lin (2006), "Model selection and estimation in regression with grouped variables." *Journal of the Royal Statistical Society Series B*, 68, 49–67.

Zaslavskiy, M., F. Bach, and J. Vert (2009), "A path following algorithm for the graph matching problem." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31, 2227–2242.

Zhang, Cun-Hui (2010), "Nearly unbiased variable selection under minimax concave penalty." *Ann. Statist.*, 38, 894–942.

Zheng, Xun, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing (2018), "Dags with no tears: Continuous optimization for structure learning." In *NeurIPS*.

Zheng, Xun, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric Xing (2020), "Learning sparse nonparametric dags." volume 108 of *Proceedings of Machine Learning Research*, 3414–3425, PMLR.

Ziegler, Günter M. (1995), *Lectures on polytopes*. Springer-Verlag, New York.

Zimmerman, Dale L. and Vicente A. Nunez-Anton (2010), *Antedependence Models for Longitudinal Data*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability, Taylor & Francis, New York.

APPENDIX A

CHAPTER 2 SUPPLEMENTARY MATERIALS

A.1 Proofs

Proof of Lemma 6

(a): We use the selection matrices K_i which are $(p-i) \times p^2$ submatrices of the $p^2 \times p^2$ identity matrix with row indices in I_j such that $L^{[i]} = K_i V$. Then it is evident that $V = \sum_{i=0}^{p-1} K_i' L^{[i]}$ and a compatible partition of B leads to

$$\begin{aligned} tr(LSL') = V'BV &= \sum_{i=0}^{p-1} (L^{[i]})' K_i B \sum_{j=0}^{p-1} K_j' L^{[j]} \\ &= \sum_{i=0}^{p-1} \sum_{j=0}^{p-1} (L^{[i]})' K_i B K_j' L^{[j]} = \sum_{i=0}^{p-1} \sum_{j=0}^{p-1} (L^{[i]})' B_{ij} L^{[j]}. \end{aligned} \tag{A.1}$$

Note that the submatrix $B_{ii} = \text{diag}(S_{1,1}, S_{2,2}, \dots, S_{p-i,p-i})$ is diagonal with positive entries, and the $(p-i) \times (p-j)$ matrix B_{ij} has nonzero values in the $((1+j-i+k), (1+k)),\ 0 \le k \le \min(p-1+i-j, p-i-1)$ entries, which correspond to the diagonal of the submatrix $S[(1+j-i):(p-i), 1:(p-j)]$ of S.

(b): From rewriting (A.1)as

$$\begin{aligned} tr(LSL') &= \sum_{i=0}^{p-1} \Big[(L^{[i]})' K_i B K_i' L^{[i]} + \sum_{i \neq j} (L^{[i]})' K_i B K_j' L^{[j]} \Big], \\ &= \sum_{i=0}^{p-1} \Big[(L^{[i]})' B_{ii} L^{[i]} + (L^{[i]})' \sum_{i \neq j} B_{ij} L^{[j]} \Big] \end{aligned} \tag{A.2}$$

the desired result follows from substituting into the objective function (2.4) and noting that $|L| = \sum_{j=1}^{p} \log L_j^{[0]}$.

(c): Since B_{ii} is positive definite, then $Q_i(\cdot)$ as the sum of strictly convex and convex functions, is strictly convex (Boyd and Vandenberghe, 2004).

Proof of Lemma 2

(a): The derivative with respect to x of the quadratic form in (2.9) is

$$-2\sum_{i=1}^{p}\frac{1}{x_i}e_i + 2C_0x + 2y_0 = 0, \tag{A.3}$$

where e_i is the $p-$vector with ith element equal to 1 and 0 otherwise. By construction, the C_0 matrix is diagonal and the first element of y_0 is 0, so that the first identity in (A.3) is

$$-\frac{1}{x_1} + (C_0)_{1,1}x_1 = 0 \Rightarrow x_1 = 1/\sqrt{(C_0)_{1,1}}.$$

Similarly, for the rows, $i = 2, \ldots, p$ we have

$$-\frac{1}{x_i} + (C_0)_{i,i}x_i + (y_0)_i = 0,$$

where its non-negative solution is as given in (2.12).

(b): In (2.10), C_i is a diagonal matrix with positive entries, setting $\tilde{y}_i = -C_i^{-1/2}y_i$ and completing the square, then finding x_i^* is equivalent to solving a generalized lasso problem:

$$\min_x \left\{\|C_i^{1/2}x - \tilde{y}_i\|_2^2 + \lambda\|Dx\|_1\right\}, \tag{A.4}$$

which has a unique solution Tibshirani and Taylor (2011).

(c):

(1): Proof is similar to the transformation in part (b).

(2): Setting the derivative of $h_i(x|y_i)$ to zero and solving for x_i gives

$$x_i^* = -\frac{1}{2}(C_i + \lambda(D'D))^{-1}y_i.$$

The matrix inverse can be computed in $O(p-i)$ flops Golub and Van Loan (1996), since C_i is diagonal and $D'D$ is a tridiagonal matrix. Here $p-i$ is the length of the vector $x_i, i = 1, \ldots, p-1$.

(d): Proof of the lemma is similar to Friedman et al. (2007, Proposition 1), thus omitted.

Proof of Theorem 1

(a): Recall that $L^{-[0]} = [(L^{[1]})', \ldots, (L^{[p-1]})']'$ where $L^{[i]}$ is the vector of ith subdiagonal. To make a change of variables in terms of difference of successive subdiagonal terms, define $\theta = [(\theta^1)', \ldots, (\theta^{p-1})']'$, where $\theta_1^j = L_1^{[j]}, \theta_i^j = L_i^{[j]} - L_{i-1}^{[j]}$, for each $1 \leq j \leq p-1,\ i = 2, \ldots, p-j$. Then, we have $L^{-[0]} = A\theta$ where $A \in R^{\binom{p}{2}\times\binom{p}{2}}$ is a block diagonal matrix where the ith $(1 \leq i \leq p-1)$ block is a $(p-i)\times(p-i)$ lower triangular matrix with ones as the nonzero entries. Substituting for $L^{-[0]}$ in $Q(L)$, we get

$$Q(L) = (L^{[0]})'B_{00}L^{[0]} + 2(L^{[0]})'B_{0\,-0}A\theta + \theta'A'B_{-0\,-0}A\theta - 2\sum_{i=1}^{p}\log L_i^{[0]} + \lambda\sum_{j=1}^{p-1}\sum_{i=2}^{p-j}|\theta_i^j|, \quad \text{(A.5)}$$

where B_{-0-0} is the submatrix that selects the rows and columns of B with indices in $\{I_{-0}, I_{-0}\}$. Next, we rewrite

$$Q(L) = x'Mx - \sum_{i=1}^{p}\log x_i + \sum_{j\in C}|x_i|, \quad \text{(A.6)}$$

where $x = [L^0, \theta]'$, $M = \tilde{A}'\tilde{B}\tilde{A}$,

$$\tilde{A} = \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0}' & A \end{bmatrix}, \tilde{B} = \begin{bmatrix} B_{00} & B_{0\,-0} \\ (B_{0\,-0})' & B_{-0\,-0} \end{bmatrix},$$

and the set $C = \{i|x_i = \theta_k^j,\ 1 \leq j \leq p-1,\ 2 \leq k \leq p-j\}$ corresponds to the indices of the difference terms in θ.

The matrix $\tilde{B}$ is positive semi-definite, since it is a submatrix of the positive semi-definite matrix B obtained by selecting specific rows and columns . Therefore, from Horn and Johnson (2012, Observation 7.1.8) the matrix M is positive semi-definite and can be written as $M = E'E$ (Horn and Johnson, 2012, Chapter 7) which establishes the equivalency of $Q(L)$ and (2.15). Since the diagonal elements of the sample covariance matrix is assumed to be positive, then B and hence E do not have 0 columns.

We note that (A.6) is not a fully form of (2.14), since the ℓ_1 penalty reformulation involves $p - i - 1$ of the $p - i$ components in each subdiagonal $L^{[i]}$. However, with the transformation similar to Rojas and Wahlberg (2014, Lemma 2.4) easily formulate (A.6) as (2.14).

(b): We show the convergence of iterates produced by Algorithm 1 to a global minimum by invoking Khare and Rajaratnam (2014, Theorem 2.2).

From Part (a) of the Theorem, there exist matrix E with no 0 columns such that (2.15 holds and Lemma 3 shows an existence of an uniform lower bound for $Q(L)$. Thus, to show convergence, it suffices to show that the assumption (A5)* Khare and Rajaratnam (2014, page 6) is satisfied or the level set of $Q(L)$, $\{L|Q(L) \leq Q(L^{(0)})\}$ is bounded. The latter property follows from the coercive property of the $Q(L)$ established in Lemma 3, since the level sets of coercive function are bounded (Bertsekas, 2016).

Proof of Lemma 3

In objective function $Q(L)$, $L \in \mathcal{L}_p$ and the eigenvalues of a lower triangular matrix are its diagonal elements, then from the well-known inequality $\log x \leq x - 1,\ x > 0$ it follows that

$$\sum_{j=1}^{p} \log L_j^{[0]} \leq \sum_{j=1}^{p} (L_j^{[0]} - 1) \leq (L^{[0]})' \mathbf{1}_p.$$

Thus

$$
\begin{aligned}
Q(L) &\geq (L^{[0]})' B_{00} L^{[0]} + 2(L^{[0]})' B_{0\,-0} L^{-[0]} + (L^{-[0]})' B_{-0\,-0} L^{-[0]} - 2(L^{[0]})' \mathbf{1}_p \\
&\overset{(*)}{=} \| B_{-0\,-0}^{1/2} L^{-[0]} + B_{-0\,-0}^{-1/2} B_{-0\,0} L^{[0]} \|_2^2 + (L^{[0]})' (B_{00} - B_{0\,-0} B_{-0\,-0}^{-1} B_{-0\,0}) L^{[0]} - 2(L^{[0]})' \mathbf{1}_p \\
&\overset{(**)}{\geq} \| B_{-0\,-0}^{1/2} L^{-[0]} + B_{-0\,-0}^{-1/2} B_{-0\,0} L^{[0]} \|_2^2 + \| K^{1/2} L^{[0]} - K^{-1/2} \mathbf{1}_p \|_2^2 - \mathbf{1}_p' K \mathbf{1}_p \\
&\overset{(***)}{\geq} (\| B_{-0-0}^{1/2} L^{-[0]} \| - \| -B_{-0-0}^{-1/2} B_{-00} L^{[0]} \|)^2 + (\| K^{1/2} L^{[0]} \| - \| K^{-1/2} \mathbf{1}_p \|)^2 - \mathbf{1}_p' K \mathbf{1}_p \\
&\geq -\mathbf{1}_p' K \mathbf{1}_p > -\infty,
\end{aligned}
$$

where the equality in (*) follows from adding and subtracting $\| B_{-0\,-0}^{-1/2} B_{-0\,0} L^0 \|_2^2$ to complete the square and writing $(L^{[0]})' B_{0\,-0} L^{-[0]} = (L^{[0]})' B_{0\,-0} B_{-0\,-0}^{-1/2} B_{-0\,-0}^{1/2} L^{-[0]}$. The inequality in (**) follows by completing the middle term as square and noting that $K = B_{00} - B_{0\,-0} B_{-0\,-0}^{-1} B_{-0\,0}$ is positive semi-definite (the Schur complement of the positive semi-definite matrix $\tilde{B}$) and (***) is based on the triangle inequality $\|x\| - \|y\| \leq \|x - y\|$.

It follows from (**) and (***) that $Q(L) \to \infty$ as any subdiagonal $\|L^{[j]}\| \to \infty$, and that if any diagonal element $L_j^{[0]} = 0$ then $Q(L) \to \infty$. Therefore, any global minimum of $Q(L)$ has a strictly positive values for $L^{[0]}$ and hence any global minimum of $Q(L)$ over the open set $\mathcal{L}_p$ lies in $\mathcal{L}_p$. Here, $\mathcal{L}_p$ is open in the set of all lower triangular matrices. Moreover, from the discussion above the function $Q(L)$ is coercive, i.e. if $\|[L^{[0]}, L^{-[0]}]\| \to \infty$, then $Q(L) \to \infty$.

Convergence of ℓ_1-trend filtering and HP

The convergence proof of trend filtering follows the same steps as described in previous section. For this case, the change of variables occurs by taking $A \in R^{\binom{p}{2} \times \binom{p}{2}}$ as a block diagonal matrix where the ith $(1 \leq i \leq p-1)$ block is a $(p-i) \times (p-i)$ lower triangular matrix with the sequence $1, \ldots, p-j$ as a nonzero elements in jth column (Kim et al., 2009, Section 3.2). The rest of the proof is similar to Appendix A.1, thus omitted.

For the convergence of HP, we note that for this case $Q(L)$ is convex differentiable function and the existing literature can be used to show convergence. For example see Luo and Tseng (1992).

Proof of Lemma 4

We use ideas similar to the Friedman et al. (2010); Cai et al. (2011); Khare et al. (2019). We start by considering two cases

Case 1 ($n \geq p$) Each iteration of SC Algorithm sweeps over diagonal and subdiagonal elements. Thus, update of the diagonal consists of estimating y and then computing diagonal using Lemma 2. From the discussion provided before the Theorem 6, recall that matrix B_{ii} is diagonal and B_{ij} has $p - j + 1$ nonzero elements located in separate columns, for $0 \leq i, j \leq p, \ i \neq j$. Thus the complexity of computing y_0 in SC algorithm is

$$\sum_{j=1}^{p-1}(p - j) = p(p - 1) - \frac{p(p - 1)}{2} \approx O(p^2)$$

From the Lemma 2, the computational cost of estimating diagonal is $O(p)$. Therefore the cost of diagonal update can be done in $p(p + 1)/2$ steps.

The update of each subdiagonal consist of computing $y_i, \ 1 \leq i \leq p - 1$ and estimating the subdiagonal in SC algorithm. Thus, the cost of estimating y_i is

$$p + \sum_{j=0}^{i-1}(p - j) + \sum_{j=i+1}^{p-1}(p - j) = \frac{p(p - 1)}{2} - p - i^2$$

and since each iteration sweeps over $p - 1$ subdiagonals we have

$$\sum_{i=1}^{p-1} \frac{p(p - 1)}{2} - p - i^2 \approx O(p^3).$$

Case 2 ($n < p$) We use similar technique as in Khare et al. (2019, Lemma C.1). Note that, since $S = YY'/n$, where $Y \in R^{p\times n}$ matrix, then $B = S \otimes I_p = (Y \otimes I_p)(Y \otimes I_p)'/n = AA'$, where $A = (Y \otimes I_p)/\sqrt{n}$. Moreover $B_{jk} = A_{j\cdot}A'_{\cdot k}$, where $A_{j\cdot}$ is submatrix whose rows were selected from index I_j. Recall $V = vec(L)$ and let $r(V) = A'V \in R^{pn}$, which takes $O(np^2)$ iterations, due to sparsity structure of A. Given initial value V^0, we evaluate $r(V^0) = A'V^0$ and keep truck of

A^tV^{current}. If V and $\tilde{V}$ differ only in one block coordinate k, then

$$(A'V)_j = \sum_{j=0}^{p-1} A_{\cdot j}\tilde{L}^{[j]} = \sum_{j=0}^{p-1} A_{\cdot j}L^{[j]} + A_{\cdot k}(\tilde{L}^{[k]} - L^{[k]}), \tag{A.7}$$

for $1 \leq j \leq np$. Therefore it takes $O(np)$ computations to update $A'V$ to $A'\tilde{V}$. Hence, after each block update in SC algorithm, it will take $O(np)$ computations to update r to its current value. Thus, the computation of y_i can be transformed into

$$\sum_{j\neq i} B_{ij}(L^{[j]}) = \sum_{j=1}^{p} B_{ij}L^{[j]} - B_{ii}L^{[i]} = A_{i\cdot}\sum_{j=1}^{p} A'_{\cdot j}L^{[j]} - B_{ii}L^{[i]}, \tag{A.8}$$

for $0 \leq i \leq p-1$. It follows the update of k'th block in (A.7), consequently in (A.8) takes $O(np)$ computations. Hence one iteration will take $O(np^2)$ computations.

Proof of Theorem 2

(a): From (2.1), for any two elements in ith subdiagonal u, v

$$|L^{[i]}(u) - L^{[i]}(v)| = |\frac{T^{[i]}(u)}{\sigma(u)} - \frac{T^{[i]}(v)}{\sigma(v)}| = |\frac{T^{[i]}(u) - T^{[i]}(v)}{\sigma(v)} + \frac{T^{[i]}(u)\Delta_{vu}(\sigma)}{\sigma(u)\sigma(v)}|,$$

where $\Delta_{vu}(\sigma) = \sigma(v) - \sigma(u)$. The simple algebra shows that

$$|L^{[i]}(u) - L^{[i]}(v)| \leq \frac{1}{c}|\Delta_{uv}(T^{[i]})| + \frac{1}{c^2}|T^{[i]}(u)||\Delta_{vu}(\sigma)| \tag{A.9}$$

(b): The bounded total variation of $L^{[i]}$ follows from the fact that it is product of two functions of bounded total variation (Grady, 2009, Theorem 2.4) and (2.17) follows from summing (A.9) over $u, v \in [0,1]$

$$\begin{aligned}|L^{[i]}(u) - L^{[i]}(v)| &= |\frac{T^{[i]}(u)}{\sigma(u)} - \frac{T^{[i]}(v)}{\sigma(v)}| = |\frac{T^{[i]}(u)}{\sigma(u)} - \frac{T^{[i]}(u)}{\sigma(v)} + \frac{T^{[i]}(u)}{\sigma(v)} - \frac{T^{[i]}(v)}{\sigma(v)}| \\ &\leq \frac{|T^{[i]}(u)||\sigma(v) - \sigma(u)|}{\sigma(u)\sigma(v)} + \frac{|T^{[i]}(v) - T^{[i]}(u)|}{\sigma(v)}\end{aligned}$$

(c): We say that the matrix $A \in R^{p\times p}$ belongs to the class TV$(R^{p\times p})$ if its diagonal and subdiagonals are functions of bounded variation. The following notation introduced in Golub and Van Loan (1996, Chapter 1.2.8) simplifies the discussion of the proof. For $L \in R^{p\times p}$ we introduce the matrix $D(L,i) \in R^{p\times p}$, which has the same ith sub(sup)diagonal as L and 0 elsewhere. Clearly, if $A \in TV(R^{p\times p})$ then $D(A,i) \in TV(R^{p\times p}),\ 0 \leq i \leq p-1$. For the lower triangular matrix L we have

$$L = \underset{D(L,0)}{\begin{pmatrix} L_{11} & 0 & \dots & 0 \\ 0 & L_{22} & \vdots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \dots & 0 & L_{pp} \end{pmatrix}} + \underset{D(L,1)}{\begin{pmatrix} 0 & \dots & \dots & 0 \\ L_{21} & 0 & \vdots & \vdots \\ \vdots & \ddots & \vdots & 0 \\ 0 & 0 & L_{p,p-1} & 0 \end{pmatrix}} + \cdots + \underset{D(L,p-1)}{\begin{pmatrix} 0 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ L_{p1} & 0 & \dots & 0 \end{pmatrix}}$$

and

$$\Omega = L'L = (D(L,p-1) + \cdots + D(L,0))'(D(L,p-1) + \cdots + D(L,0)).$$

From the structure of $D(L,i)$'s it can be seen that the ith subdiagonal of Σ can be written as the sum of the ith subdiagonals of the following matrix products

$$\Omega^i = \sum_{j=0}^{p-i-1} ((D(L,j)'D(L,j+i))^i, \tag{A.10}$$

where from the position of degenerate values, the matrix product $D(L,j)'D(L,j+i)$ has nonzero values on the ith subdiagonal and zero elsewhere. Moreover, nonzero values in the ith subdiagonal of $(D(L,j)'D(L,j+i))^i = (L^{[i]}_{(p-j-i-1):(p-1)})'L^{[j+i]}$. Now, since the product of two functions of total bounded variation are of bounded variation and after adding and subtracting corresponding terms as in the proof of Lemma 2, we get

$$TV(((D(L,j)'D(L,j+i))^i) \leq m_j K_{j+i} + m_{j+i} K_j$$

and the result follows from (A.10).

Now we show the converse, i.e. if $\Omega \in \text{TV}(R^{p\times p})$ then there exist a unique $L \in \text{TV}(T^{p\times p})$ and $\Omega = L'L$. The proof uses similar strategy proposed in (Chern and Dieci, 2000, Lemma 2.8). Before introducing the main argument, we state the following lemma, which will be used in the proof.

Lemma 11. *If lower triangular matrices $G, M \in TV(R^{p\times p})$ then $A = GM \in TV(R^{p\times p})$*

Proof. Using the matrix notation $(D(L, \cdots)$ introduced in part(a), it can be shown that the ith subdiagonal of the matrix product $A = GM$ can be written as

$$A^j = (GM)^j = \sum_{i=0}^{j} (D(G,i)D(M,j-i))^j$$

and the result follows by recalling that the product of the functions of bounded variation is of bounded variation. □

The main argument consist in writing $\Omega = \begin{bmatrix} \hat{\Omega} & b \\ b' & \omega_{pp}^2 \end{bmatrix}$ and let $G_1 = \begin{bmatrix} I_{p-1} & b/\omega_{pp} \\ 0 & \omega_{pp} \end{bmatrix}$. From the construction of G_1 and $\Omega \in \text{TV}(R^{p\times p})$, it is easy to see that $G_1 \in \text{TV}(R^{p\times p})$ and $G_1^{-1}\Omega G_1^{-t} = \begin{bmatrix} \Omega_1 & 0 \\ 0 & 1 \end{bmatrix}$, where $\Omega_1 = \hat{\Omega} - bb'/\omega_{pp}^2$. Clearly, $\Omega_1 \in TV(R^{p-1\times p-1})$ since $\hat{\Omega} \in TV(R^{p-1\times p-1})$ by construction and $TV(\Omega_1^{[i]}) = TV(\hat{\Omega}^{[i]} - (bb')^i/\omega_{pp}^2) \leq TV(\hat{\Omega}^{[i]}) < \infty$. By repeating this procedure and using Lemma 11, result follows.

A.2 Tuning Parameter Selection

We use BIC-like measure and cross-validation to choose the tuning parameter λ. In particular, the tuning parameter λ is determined by choosing the minimum of BIC-like measure and CV over the grid. BIC is defined as:

$$BIC(\lambda) = ntr(\hat{L}'\hat{L}S) - n \log\det(\hat{L}'\hat{L}) + \log n \times E,$$

where E denoted the degrees of freedom, n and S are the sample size and covariance matrix, respectively. For example for the sparse fused lasso, E corresponds to number of nonzero fused groups in $\hat{L}$ (Tibshirani and Taylor, 2011).

For $K-$fold cross-validation, we randomly split the full dataset $\mathcal{D}$ into K subsets of about the same size, denoted by $\mathcal{D}^{\nu},\ \nu = 1, \ldots, K$. For each ν, $\mathcal{D} - \mathcal{D}^{\nu}$ is used to estimate the parameters and $\mathcal{D}^{\nu}$ to validate. The performance of the model is measured using the log-likelihood. We choose the tuning parameter λ as a minimum of the $K-$fold cross-validated log-likelihood criterion over the grid.

$$CV(\lambda) = \frac{1}{K} \sum_{\nu=1}^{K} \Big(d_{\nu} \log \det(\hat{L}'_{-\nu} \hat{L}_{-\nu})^{-1}) + \sum_{I_{\nu}} y_i' \hat{L}'_{-\nu} \hat{L}_{-\nu} y_i \Big),$$

where $\hat{L}_{-\nu}$ is the estimated Cholesky factor using the data set $\mathcal{D} - \mathcal{D}^{\nu}$, I_{ν} is the index set of the data in $\mathcal{D}$, d_{ν} is the size of I_{ν}, and y_i is the ith observation of the dataset $\mathcal{D}$.

A.3 Additional Simulation

In this section we provide additional simulation results. Two different cases are considered. In the first case, matrix T is a full lower triangular matrix and each subdiagonal is generated from the first subdiagonal of one of the Cases (A-D) by eliminating the corresponding excessive elements. In the second case, matrix T follows nonhierarchical structure, in a sense described in Yu and Bien (2017). That is, in a full lower triangular matrix T, we enforce first and last $p/3$ subdiagonals admit nonzero values, drawn from uniform [0.1, 0.2] and positive/negative signs are then assigned with probability 0.5. The rest of $p/3$ subdiagonals admit zero value. See Figure A.3 for an illustration. For the latter case, Sparse SC have been used for the estimation. That is we use two tuning parameters λ_1 to control sparsity and λ_2 smoothness, respectively.

For both cases, we consider settings when $p = 50, 150$ and $n = 100$, however because of the space limitation only $p = 150$ is reported. Each possible setting is repeated over 20 simulated datasets. The tuning parameters were chosen using cross-validation. Moreover, for the second case we compare the results from sparse SC (HP, Fused, Trend) estimator with CSCS and HSC using a receiver operating characteristic curve, or ROC curve.

We start by providing results for the first case. The Figure A.1 plots the first four estimated subdiagonals of the full lower triangular matrix T, using SC estimator. From the figure, the first two subdiagonals correspond to the Case B, the third to the Case D and the fourth to the Case C, respectively. Visually, the SC-Fused captures the step function the best for the first subdiagonal. However, all three estimators failed to capture the stepwise linear structure of the second subdiagonal, but there is a significant improvement of SC estimator to capture the wiggliness of the Markov process in the third subdiagonal (SC-HP being the best) and smooth, slow time-varying structure of the fourth subdiagonal (SC-Trend being the best). Next we report the performance of three estimators using Frobenius and Infinity norm. Figure A.2 plots the results. Overall, for matrix T, SC-Trend filtering provides the lowest Frobenius and Infinity norm followed by SC-Fused.

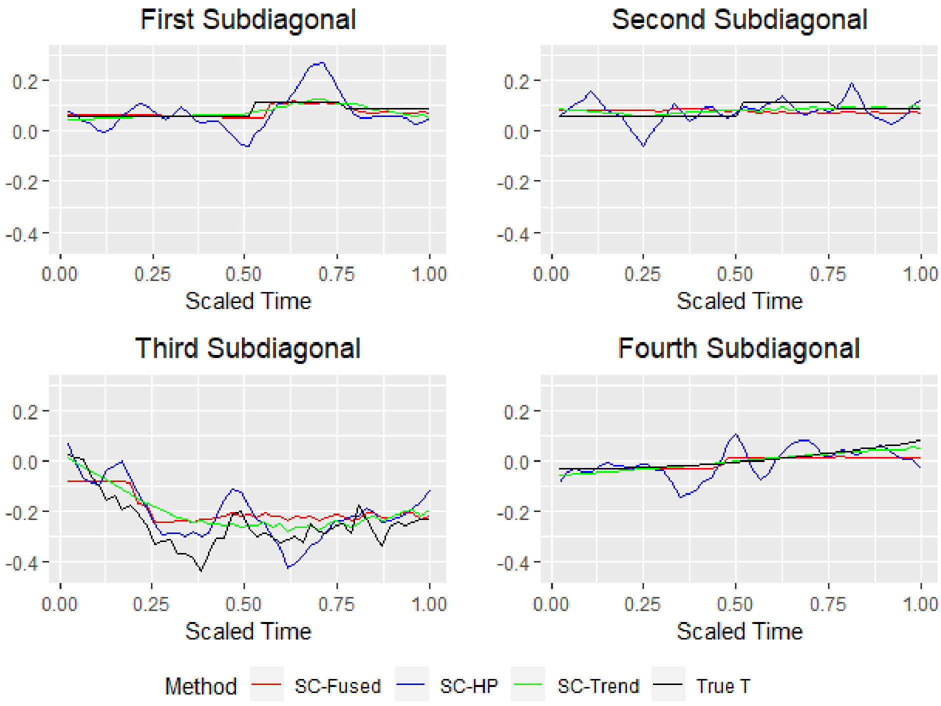


Figure A.1: Estimated first four subdiagonals ($p = 150$).

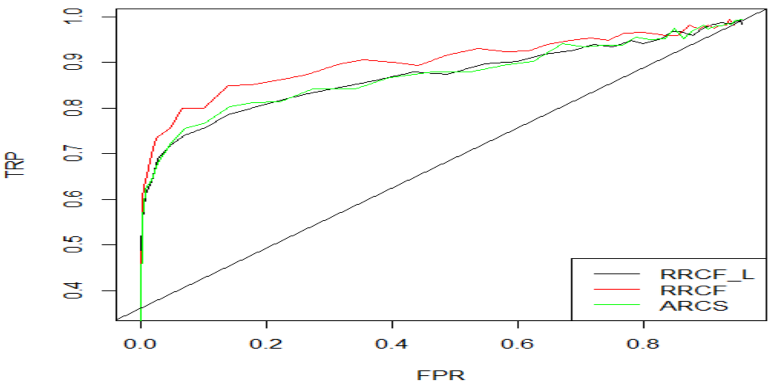


Figure A.2: ROC curve for $p = 150$.

Remark 1. *Relying on the result above, one can learn the lower triangular matrix $L(T)$ by considering the penalty form as an additional parameter to tune for each subdiagonal.*

Now, we compare the performance of the SC with the CSCS and HSC on the support recovery, when the structure is non-hierarchical. Comparison is implemented using ROC curves. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various penalty parameter settings (Friedman et al., 2010). Here, the ROC curve is obtained by varying around 60 possible values for the penalty parameter λ_1. For the SC-Fused, Trend and HP, the smoothing tuning parameter λ_2 is obtained from the cross-validation by fixing λ_1 in a given value. In applications, FPR is usually controlled to be sufficiently small, thus following Khare et al. (2019), the focus is on comparing portion of ROC curves for which FPR is less than 0.15. The comparison of ROC curves is implemented using Area-under-the-curve (AUC) (Friedman et al., 2010).

Table A.1 reports the mean and the standard deviation (over 20 simulations) for the AUCs for SC (HP, Fused and Trend), CSCS and HSC when $p = 150$ and $n = 100$. The best result is given in bold.

From the table above, it can be seen that HSC provides the best result. However, Figure A.3, which captures snapshot of the graphical comparison of the estimated matrix L for the five esti-

Table A.1: Mean and Standard Deviation of area-under-the-curve (AUC) for 20 simulations for p = 150.

Method	Mean	Std. Dev
SC-HP	0.068	0.019
SC- Fused	0.121	0.023
SC- Trend	0.104	0.015
CSCS	0.058	0.007
HSC	**0.137**	0.025

mators, sheds more lights into characteristics of estimators. As can be seen, even though HSC provides the highest AUC for FPR less than 0.15, it fails to capture the zero gap between subdiagonals of matrix L compare, for example, with SC-Fused, which provides the second best result in the Table A.1.

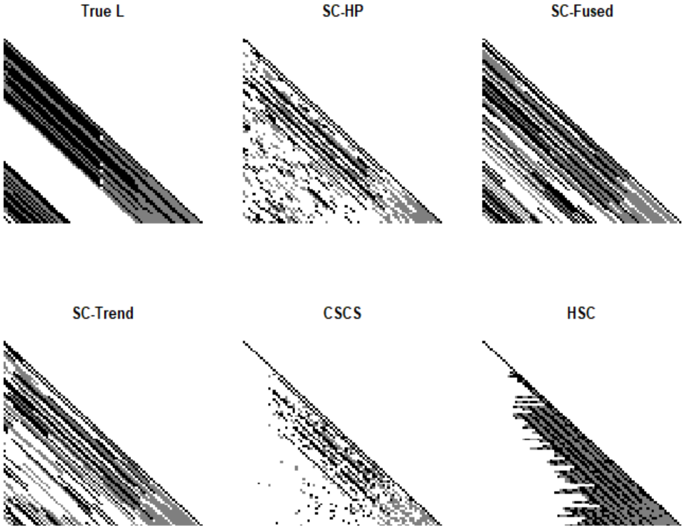


Figure A.3: Comparison of snapshots for the simulated example for $p = 150$.

Additional Visualization of Subdiagonals

In this section, we provide an addition visualization of the estimated first subdiagonals for cases A-D using CV criteria. In particular, we calculate the mean and standard deviation (sd) for each element of a lower triangular matrix over 20 simulated repetitions. Then the estimated mean and mean $\pm 2 * \text{sd}$ were plotted for each penalty function (SC-Fused, SC-HP, SC-Trend) and Case A-D, resulting to 3×3 panel plot. Figure A.4 illustrated the result. The columns and rows correspond to the penalty form and cases, respectively. In each plot, the red thick line corresponds to the mean and the gray line to mean $\pm$ 2sd, the true line is depicted in black.

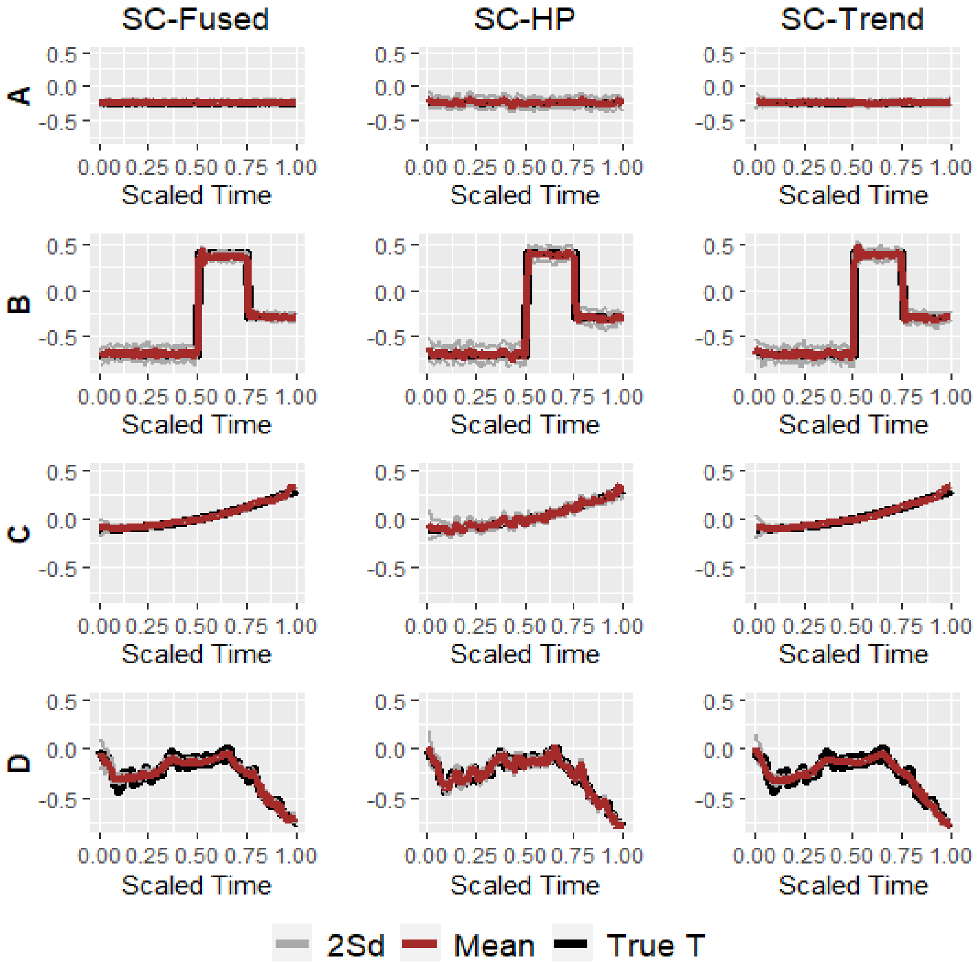


Figure A.4: Plot of the mean of the estimated first subdiagonal and $\pm 2sd$ for each penalty and case.

Illustration of Bias

Figure A.5 illustrates the plot of bias and variance for the simulated example.

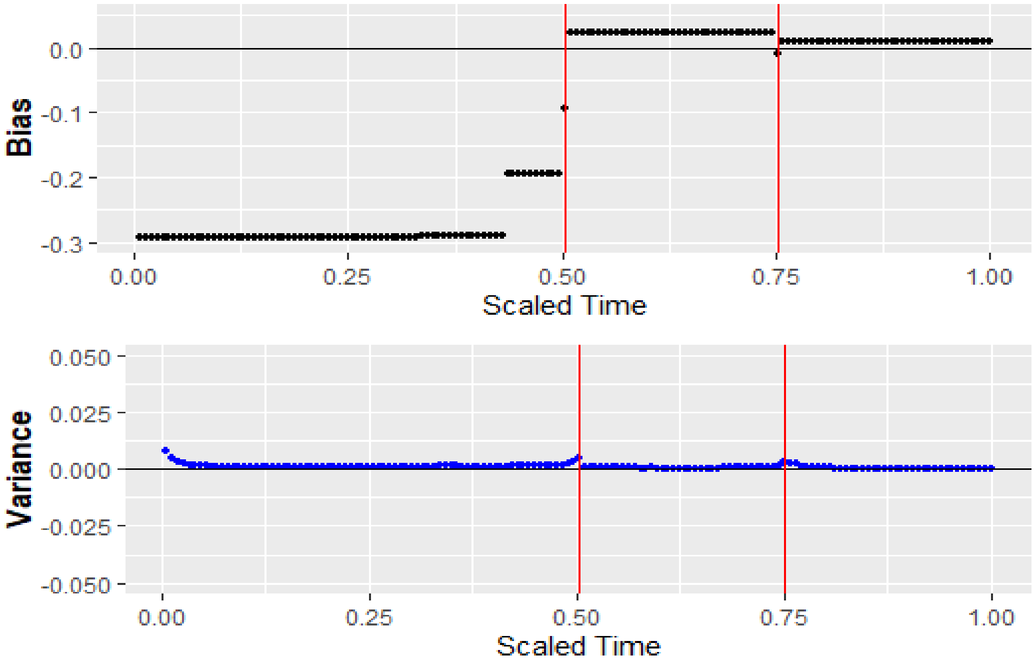


Figure A.5: The plot of bias and variance for the simulated example.

A.4 Cattle data: Additional Analysis

Figure A.6 provides the plot of the first two subdiagonals using eight estimators described in Section 2.4.5.

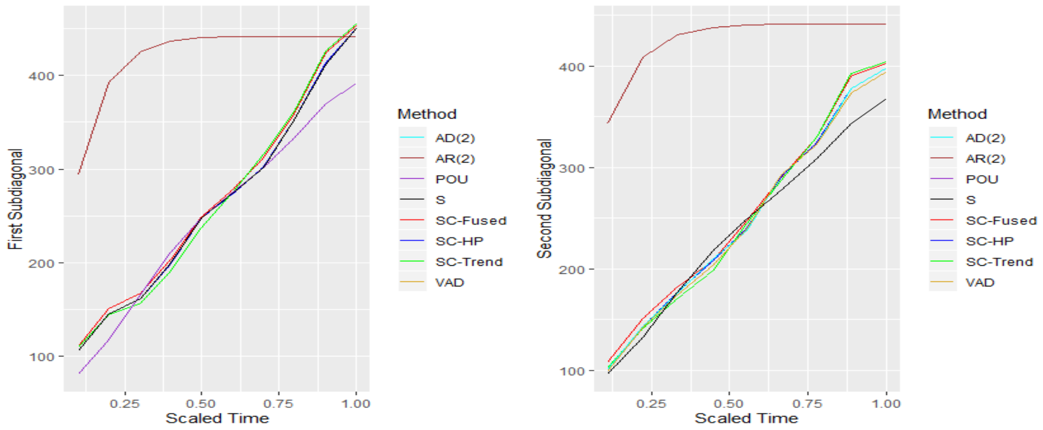


Figure A.6: Plots of estimated first and second subdiagonals of the covariance matrix for various estimation methods.

A.5 Example of SC Penalty Form

In this section we give details of the SC penalty form based on a toy example. We consider $p = 4$, i.e 4×4 lower triangular matrix L.

$$L = \begin{bmatrix} L_{1,1} & 0 & 0 & 0 \\ L_{2,1} & L_{2,2} & 0 & 0 \\ L_{3,1} & L_{3,2} & L_{3,3} & 0 \\ L_{4,1} & L_{4,2} & L_{4,3} & L_{4,4} \end{bmatrix}$$

Here $L^{[0]} = [L^{[0]}_1, L^{[0]}_2, L^{[0]}_3, L^{[0]}_4]' = [L_{1,1}, L_{2,2}, L_{3,3}, L_{4,4}]'$, $L^{[1]} = [L^{[1]}_1, L^{[1]}_2, L^{[1]}_3]' = [L_{2,1}, L_{3,2}, L_{4,3}]'$, $L^{[2]} = [L^{[2]}_1, L^{[2]}_2]' = [L_{3,1}, L_{4,2}]'$ and $L^{[3]} = L_{4,1}$. The penalty for the subdiagonal $L^{[1]}$ is

$$P_{\nabla}(L^{[1]}) = \sum_{j=2}^{4-1} |L^{[1]}_j - L^{[1]}_{j-1}| = \sum_{j=2}^{4-1} |L_{j+1,j} - L_{j,j-1}|$$

The penalties for the second $L^{[2]}$ and third $L^{[3]}$ subdiagonal can be written similarly.

APPENDIX B

CHAPTER 3 SUPPLEMENTARY MATERIALS

B.1 Proofs

Proof of Lemma 5

For the proof of only if side; i.e., P is a permutation matrix or $P = J/p$, the equality holds trivially from the definition of the Frobenius norm (Horn and Johnson, 2012, Chapter 5.6).

For the if part, it is known that the maximum spectral radius and the spectral norm $\|P\|_2$ of the doubly stochastic matrix $P \in \mathcal{D}_p$ are equal 1 (Horn and Johnson, 2012, Chapter 8.7). From the matrix norm inequality (Horn and Johnson, 2012, Corollary 5.4.5)

$$1 = \|P\|_2^2 \leq \|P\|_F^2 = \sum_{j=1}^{p} \sigma_j^2(P) \leq p\|P\|_2^2 \leq p, \tag{B.1}$$

where $\sigma_j(\cdot)$ is the jth singular value. Thus, $\|P\|_F = \|P\|_2 = 1$ if and only if it is a matrix of rank one; i.e., $P = J/p$ from the Marcus and Minc (1962, Theorem 4).

On the other hand, it is easy to see that $\|P\|_F^2 = p$ equality holds if and only if $\sigma_j(p) = 1,\ j = 1, \ldots, p$, that is $P \in \mathcal{P}_p$, and any doubly stochastic matrix with Frobenius norm $\sqrt{p}$ is a permutation matrix.

Proof of Corollary 1

We write for $P \in \mathcal{D}_p$

$$tr(LPSP^tL^t) = vec(P)^t(L^tL \otimes S)vec(P) \geq \lambda_1(L^tL \otimes S)\|P\|_F^2,$$

where $\lambda_1(A)$ is the smallest eigenvalue of the symmetric matrix A and the minimum value achieved when $P = J/p$ from the Lemma 5.

Proof of Lemma 6

(a): The Hessian of the objective function can be found by noting that

$$\frac{1}{2}tr(LPSP^tL^t) - \frac{1}{2}\mu\|P\|_F^2 = \frac{1}{2}vec(P)^t(L^tL \otimes S)vec(P) - \frac{1}{2}\mu vec(P)^t vec(P)$$

where $vec(\cdot)$ is the usual matrix vectorization operator. Thus the Hessian is

$$L^tL \otimes S - \mu\mathbf{I},$$

and the result easily follows from the definition of convexity (Boyd and Vandenberghe, 2004).

(b): We first show that the transformation (3.15) is equivalent to (3.14). Following Fogel et al. (2013), we write

$$\|TP\|_F^2 = tr(P^tT^tTP) = tr(P^tP - \mathbf{1}\mathbf{1}^t/p) = \|P\|_F^2 - 1,$$

where we use idempotent property of the projection matrix T. Thus, (3.15) has the same objective function as (3.14) up to a constant. To show convexity we look on the Hessian of the objective function. Note that

$$\frac{1}{2}tr(LPSP^tL^t) - \frac{1}{2}\mu\|TP\|_F^2 = \frac{1}{2}vec(P)^t(L^tL \otimes S)vec(P) - \frac{1}{2}\mu vec(P)^t(\mathbf{I} \otimes T)vec(P),$$

where $vec(\cdot)$ is the usual matrix vectorization operator. Thus the Hessian is

$$L^tL \otimes S - \mu\mathbf{I} \otimes T \tag{B.2}$$

and under $\mu \leq \lambda_2(S)\lambda_1(L^tL)$ convexity holds.

(b): Similarly, from (B.2) if $\mu > \lambda_m(S)\lambda_m(L^tL)$ then the objective function of (3.15) is concave. Thus, from the Horn and Johnson (2012, Corrollary 8.7.4) the minimum of concave function over the set of doubly stochastic matrices is attained at a permutation matrix and the proof follows.

Proof of Lemma 7

We start by assuming contradiction. The proof exploits strategy introduced in Rothman et al. (2008). Let for $\tilde{P} \in \mathcal{D}_p$ belonging to a Birkhoff polytope

$$\begin{aligned} Q(\tilde{P}) &= tr(L\tilde{P}S\tilde{P}^tL^t) - \frac{1}{2}\mu\|\tilde{P}\|_F^2 - \frac{1}{2}tr(LPSP^tL^t) - \frac{1}{2}\mu\|P\|_F^2 \\ &= tr(L(\tilde{P}-P)S(\tilde{P}-P)^tL^t) - \frac{1}{2}\mu(\|\tilde{P}\|_F^2 - \|P\|_F^2), \end{aligned} \tag{B.3}$$

where $P \in \mathcal{P}_p$ is a true permutation matrix, $\|P\|_F^2 = p$ from Lemma 5. Our estimate $\hat{P}$ minimizes $Q(\tilde{P})$, or equivalently $\hat{\Delta} = \hat{P} - P$ minimizes $G(\Delta) = Q(P + \Delta)$, where $\Delta = \tilde{P} - P$. Under convexity condition in Lemma 6(a), $G(\Delta)$ is a convex function and $G(\hat{\Delta}) \leq G(0) = 0$.

Next we introduce the set

$$\Theta_n = \{\Delta : \|\Delta\|_F = r_n\},$$

where $r_n \to 0$. Now if we show that $\inf\{G(\Delta) : \Delta \in \Theta_n\} > 0$ then $\hat{\Delta} \in \Theta_n$ and $\|\hat{\Delta}\|_F \leq r_n$.

We write

$$G(\Delta) = I + II, \tag{B.4}$$

where $I = tr(L\Delta S\Delta^tL^t)$ and $II = \frac{\mu}{2}(\|P\|_F^2 - \|P+\Delta\|_F^2)$.

For the part I,

$$I = tr(L^tL\Delta S\Delta^t) = vec(\Delta)^t(L^tL \otimes S)vec(\Delta) \geq \lambda_1(L^tL \otimes S)\|\Delta\|_F^2 \tag{B.5}$$

To find lower bound for II, we denote by $C = \{(i,j) : P_{ij} = 1\}$ non-zero entries of the permutation matrix P. We also note that the cardinality $|C| = p$. Thus,

$$\|P+\Delta\|_F^2 = \sum_{(i,j)\in C}|1+\Delta_{ij}|^2 + \sum_{(m,n)\notin C}|\Delta_{mn}|^2 = p + 2\sum_{(i,j)\in C}|\Delta_{ij}| + \|\Delta\|_F^2$$

and

$$\|P + \Delta\|_F^2 - \|P\|_F^2 = 2 \sum_{(i,j)\in C} |\Delta_{ij}| + \|\Delta\|_F^2 \leq 2p \max_{i,j} |\Delta_{ij}| + \|\Delta\|_F^2 \leq 2p + \|\Delta\|_F^2, \tag{B.6}$$

where we used the norm inequality $\|A\|_{max} \leq \|A\|_F$ and $\max_{i,j}|\Delta_{ij}| \leq 1$. It follows from (B.5) and B.6 the lower bound for (B.4) is

$$\begin{aligned} G(\Delta) &\geq \lambda_1(L^t L \otimes S)\|\Delta\|_F^2 - 2p\mu - \mu\|\Delta\|_F^2 \\ &= \|\Delta\|_F^2(\lambda_1(L^t L \otimes S) - \mu - \frac{2p}{r_n^2}\mu) \end{aligned} \tag{B.7}$$

Thus, $G(\Delta) > 0$ condition holds if and only if

$$\lambda_1(L^t L \otimes S) - \mu - \frac{2p}{r_n^2}\mu > 0,$$

from which follows that

$$\mu < \frac{\lambda_1(L^t L \otimes S)}{1 + \frac{2p}{r_n^2}}$$

and from $r_n \to 0$, it follows $\mu \to 0$, which contradicts the initial statement.

Proof of Lemma 8

The result follows from the Bertsekas (2015, Proposition 6.1.2), since the gradient of the objeciton function of (3.15) is Lipschitz

$$\|L^t L P_1 S - \mu T P_1 - L^t L P_2 S + \mu T P_2\| \leq (\|L^t L\|\|S\| + \mu\|T\|)\|P_1 - P_2\|$$

and the convergence to global minimum follows from Lemma 6.

Proof of Lemma 9

We start by writing for $1 \leq j \leq k-1$

$$h_{k,A,\lambda,\gamma} = x_j^2 A_{jj} + 2x_j(\sum_{l \neq j} A_{lj} x_l) + \rho(|x_j|, \lambda, \gamma) + C_j, \tag{B.8}$$

where C_j includes terms independent of x_j. Taking derivative with respect to x_j and noting that the subdifferential

$$\rho^{`}(|x_j|, \lambda, \gamma) = \begin{cases} \lambda s - \frac{x_j}{\gamma} & |x_j| < \gamma\lambda \\ 0 & |x_j| \geq \gamma\lambda \end{cases} \tag{B.9}$$

Here, the subgradient $s = sgn(x_j)$ if $x_j \neq 0$ and take values in $[-1, 1]$ otherwise. Thus it follows

$$x_j^* = \frac{S_\lambda(-2\sum_{l \neq j} A_{lj} x_l)}{2A_{jj} - 1/\gamma}$$

Similarly,

$$h_{k,A,\lambda,\gamma} = x_k^2 A_{kk} + 2x_k(\sum_{l \neq k} A_{lj} x_l) - 2\log x_k + C_k,$$

where C_k includes terms independent of x_k and after taking derivatives with respect to x_k

$$\frac{-2}{x_k} + 2x_k A_{kk} + 2\sum_{l \neq k} A_{lk} x_l = 0 \iff x_k^2 A_{kk} + \sum_{l \neq k} A_{lk} x_l x_k - 1 = 0,$$

and (3.27) follows after retaining the positive root of the above quadratic equation.

Proof of Lemma 10

(a): From (3.26)

$$h_{k,A,\lambda,\gamma}(x) \geq 2x_k - 2,$$

where we used that A is positive semidefinite and $x^t A x \geq 0$, $\rho(|x_i|, \lambda, \gamma) \geq 0$ and for $y > 0$, $\log y \leq 1 - y$.

(b): We denote by $D_u h$ and $D_u^2 h$ the derivative and the second derivative of h in the direction of u. Thus, the proof follows from (B.8) and (B.9) by writing

$$\min\{D^2_{\beta_i^-} h_{k,A,\lambda,\gamma}(\beta), D^2_{\beta_i^+} h_{k,A,\lambda,\gamma}(\beta)\} \geq 2A_{ii} - \frac{1}{\gamma}$$

(c): Note that

$$\inf_{L \in \mathcal{L}_p} Q_{RRCF}(L) \geq -2p > -\infty$$

directly follows from the part (b) of the proof. Moreover, from (3.24) if $|\beta^i_j| \to \infty$ or $\beta^i_i = 0$, then $Q_{RRCF} \to \infty$. Therefore, any local minimum of Q_{RRCF} over the open set $\mathcal{L}_p$ lies in $\mathcal{L}_p$.

Proof of Theorem 3

For the proof of Theorem 3, we exploit Tseng (2001, Theorem 5.1), where the author established sufficient conditions for the convergence of cyclic coordinate descent algorithms to coordinate-wise minima. The strict convexity of (3.23) with respect to each coordinate direction and lower boundedness established in Lemma 10 satisfy the required conditions in Theorem 5.1. Thus, convergence to a coordinate-wise minimum point is guaranteed. Moreover, since all directional derivatives exist, every coordinate-wise minimum is also a local minimum.

Proof of Theorem 4

We start by showing that $\mathcal{L}_n$ satisfies RSC conditions. Recall that the differentiable function $\mathcal{L}_n : \mathcal{R}^{p \times p} \to \mathcal{R}$ satisfies RSC condition if:

$$\langle \nabla \mathcal{L}_n(L + \Delta) - \nabla \mathcal{L}_n(L), \Delta \rangle \geq \begin{cases} \alpha_1 \|\Delta\|_F^2 - \tau_1 \frac{\log p}{n} \|\Delta\|_1^2, & \forall \|\Delta\|_F \leq 1 \qquad \text{(B.10)} \\ \alpha_2 \|\Delta\|_F - \tau_2 \sqrt{\frac{\log p}{n}} \|\Delta\|_2, & \forall \|\Delta\|_F \geq 1\,, \qquad \text{(B.11)} \end{cases}$$

where the α_j's are strictly positive constants and the τ_j's are nonnegative constants. From Loh and Wainwright (2015, Lemma 4), under conditions of Theorem 4, if (B.10) holds then (B.11) holds.

Thus, we concentrate only on showing that (B.10) holds for $\|\Delta\|_F \leq 1$. Recall that

$$\mathcal{L}_n(L) = \mathrm{tr}(SL^tL) - 2\log|L| \tag{B.12}$$

Lemma 12. *The cost function (B.12) satisfies RSC condition (4) with $\alpha_1 = (\kappa+1)^{-2}$ and $\tau_1 = 0$; i.e.,*

$$\langle \nabla\mathcal{L}_n(L+\Delta) - \nabla\mathcal{L}_n(L), \Delta\rangle \geq (\kappa+1)^{-2}\|\Delta\|_F^2,\ \forall\|\Delta\|_F \leq 1 \tag{B.13}$$

The proof is provided in Appendix B.1.

From the penalty conditions listed in Appendix B.2, $\rho_\mu(L,\lambda) = \rho(L,\lambda) + \frac{\mu}{2}\|L\|_F^2$ is convex, where in case of MCP $\mu = 1/\gamma$. Thus,

$$\rho_\mu(L,\lambda) - \rho_\mu(\hat{L},\lambda) \geq \langle \nabla\rho_\mu(\hat{L},\lambda), L-\hat{L}\rangle = \langle \nabla\rho(\hat{L},\lambda) + \mu\hat{L}, L-\hat{L}\rangle,$$

which implies that

$$\langle \nabla\rho(\hat{L},\lambda), L-\hat{L}\rangle \leq \rho(L,\lambda) - \rho(\hat{L},\lambda) + \frac{\mu}{2}\|\hat{L}-L\|_F^2 \tag{B.14}$$

From stationarity condition (3.30)

$$\langle \nabla\mathcal{L}_n(\hat{L}), L-\hat{L}\rangle \geq -\langle \nabla\rho(\hat{L},\lambda), L-\hat{L}\rangle$$

and combining above result with (B.13)

$$\begin{aligned}
(1+\kappa)^{-2}\|\Delta\|_F^2 &\leq \langle \mathcal{L}_n(\hat{L}), \Delta\rangle - \langle \nabla\mathcal{L}_n(L), \Delta\rangle \\
&\leq \langle \nabla\rho(\hat{L},\lambda), L-\hat{L}\rangle - \langle \nabla\mathcal{L}_n(L), \Delta\rangle \\
&\leq \rho(L,\lambda) - \rho(\hat{L},\lambda) + \frac{\mu}{2}\|\hat{L}-L\|_F^2 - \langle \nabla\mathcal{L}_n(L), \Delta\rangle
\end{aligned}$$

After rearrangement and H´older inequality

$$\Big((1+\kappa)^{-2}) - \frac{\mu}{2}\Big)\|\Delta\|_F^2 \le \rho(L,\lambda) - \rho(\hat{L},\lambda) + \|\nabla\mathcal{L}_n(L)\|_\infty\|\Delta\|_1$$

From Loh and Wainwright (2015, Lemma 4)

$$\lambda\|\Delta\|_1 \le \rho(\Delta,\lambda) + \frac{\mu}{2}\|\Delta\|_F^2$$

and from Yu and Bien (2017, Lemma 15) under the assumed scaling of λ

$$\|\nabla\mathcal{L}_n(L)\|_\infty \le \frac{\lambda}{2}$$

with probability going to 1. Combining above two results and using subadditive property; i.e., $\rho(\Delta,\lambda) \le \rho(L,\lambda) + \rho(\hat{L},\lambda)$:

$$\begin{aligned}\Big((1+\kappa)^{-2} - \frac{\mu}{2}\Big)\|\Delta\|_F^2 &\le \rho(L,\lambda) - \rho(\hat{L},\lambda) + \frac{\lambda}{2}\|\hat{\Delta}\|_1 \\ &\le \rho(L,\lambda) - \rho(\hat{L},\lambda) + \frac{\rho(\Delta,\lambda)}{2} + \frac{\mu}{4}\|\Delta\|_F^2 \\ &\le \rho(L,\lambda) - \rho(\hat{L},\lambda) + \frac{\rho(L,\lambda) + \rho(\hat{L},\lambda)}{2} + \frac{\mu}{4}\|\Delta\|_F^2\end{aligned}$$

After rearranging and using $3/4\mu \le (1+\kappa)^{-2}$

$$0 \le \Big((1+\kappa)^{-2} - \frac{3}{4}\mu\Big)\|\Delta\|_F^2 \le 3\rho(L,\lambda) - \rho(\hat{L},\lambda) \tag{B.15}$$

From (B.15) and Loh and Wainwright (2015, Lemma 5) follows

$$\rho(L,\lambda) - \rho(L,\lambda) \le 2\lambda\|\Delta_S\| - \lambda\|\Delta_{S^c}\| \Rightarrow \|\Delta_{S^c}\|_1 \le 3\|\Delta_S\|_1$$

Thus,

$$\Big(2(1+\kappa)^{-2}-\frac{3}{2}\mu\Big)\|\Delta\|_F^2\leq 3\lambda\|\Delta_S\|_2-\lambda\|\Delta_{S^c}\|_1\leq 3\lambda\|\Delta_S\|_1\leq 3\lambda\sqrt{p+s}\|\Delta\|_F,$$

from which we conclude that

$$\|\Delta\|_F\leq \frac{6\lambda\sqrt{p+s}}{4(1+\kappa)^{-2}-3\mu}, \tag{B.16}$$

and the result follows from the chosen scaling of λ.

For the precision matrix bound, from page 45 of Yu and Bien (2017) we note that

$$\hat{L}^t\hat{L}-L^tL=(\hat{L}-L)^t(\hat{L}-L)+(\hat{L}-L)^TL+L^t(\hat{L}-L)$$

and

$$\|L^t(\hat{L}-L)\|_F\leq |||L|||_2\|\hat{L}-L\|_F$$

From submultiplicativity property of matrix norm

$$\|(\hat{L}-L)^t(\hat{L}-L)\|_F\leq \|(\hat{L}-L)\|_F^2$$

Therefore,

$$\|\hat{L}^t\hat{L}-L^tL\|_F\leq (\|\hat{L}-L\|_F+2|||L|||_2)\|\hat{L}-L\|_F \tag{B.17}$$

Proof of Lemma 12

The following facts will be useful in the proof.

Fact 1

1. $(K_{pp})^{-1}=K_{pp}$
2. $\lambda_{max}(K_{pp})=1$
3. $tr(ABCD)=vec(D^t)(C^t\otimes A)vec(B)$

4. $\lambda_{max}(A \otimes B) = \lambda_{max}(A)\lambda_{max}(B)$

where K_{pp} is the commutation matrix such that $vec(L) = K_{pp}vec(L^t)$. The proof of the facts can be found in Magnus and Neudecker (1986, Section 4).

To show the RSC condition, we rely on the directional derivatives (for example see Tao (2016, Section 6.3)). In particular, if we denote by $D_\Delta \mathcal{L}_n(L)$ the directional derivative with respect to the direction Δ, then from Tao (2016, Lemma 6.3.5) :

$$\langle \nabla \mathcal{L}_n(L), \Delta \rangle = D_\Delta \mathcal{L}_n(L) = 2\text{tr}[(SL^t - L^{-1})\Delta] \tag{B.18}$$

Similarly

$$\langle \nabla \mathcal{L}_n(L+\Delta), \Delta \rangle = D_\Delta \mathcal{L}_n(L+\Delta) = 2\text{tr}[(S(L+\Delta)^t - (L+\Delta)^{-1})\Delta] \tag{B.19}$$

From Woodbury identity (Horn and Johnson, 2012)

$$(L+\Delta)^{-1} = L^{-1} - L^{-1}\Delta(L+\Delta)^{-1}$$

Plugging back into (B.19) and after some algebra

$$\langle \nabla \mathcal{L}_n(L+\Delta), \Delta \rangle = 2\text{tr}[(S(L+\Delta)^t - (L+\Delta)^{-1})\Delta] + 2\text{tr}[S\Delta^t\Delta + L^{-1}\Delta(L+\Delta)^{-1}\Delta] \tag{B.20}$$

Thus, from (B.18) and (B.20)

$$\begin{aligned}
\langle \nabla \mathcal{L}_n(L+\Delta) - \nabla \mathcal{L}_n(L), \Delta \rangle &= 2\text{tr}[\Delta^t S\Delta + L^{-1}\Delta(L+\Delta)^{-1}\Delta] \\
&\geq \text{vec}(\Delta)^t K_{pp}((L+\Delta)^{-t} \otimes L^{-1})\text{vec}(\Delta) \\
&= \text{vec}(\Delta)^t[((L+\Delta)^t \otimes L)K_{pp}^{-1}]^{-1}\text{vec}(\Delta) \\
&\geq \lambda_{min}([((L+\Delta)^t \otimes L)K_{pp}^{-1}]^{-1})\|\Delta\|_F^2,
\end{aligned} \tag{B.21}$$

where for the first inequality we used the fact that S is positive semi-definite and the second equality follows from the Fact 1. Now, since

$$
\begin{aligned}
\lambda_{min}([((L+\Delta)^t \otimes L)K_{pp}^{-1}]^{-1}) &= \lambda_{max}^{-1}[((L+\Delta)^t \otimes L)K_{pp}^{-1}] \\
&\geq \lambda_{max}^{-1}(K_{pp}^{-1})\lambda_{max}^{-1}(L)\lambda_{max}^{-1}(L+\Delta) \geq (\kappa+1)^{-2},
\end{aligned}
\tag{B.22}
$$

where the first inequality follows from the submultiplicativity property of the norm and lower-triangularity of the L and Δ. The second inequality follows from the triangular property, the fact that $\|\Delta\|_2 \leq \|\Delta\|_F \leq 1$ and, properties of the K_{pp} stated in Fact 1. After plugging (B.22) into (B.21), the result follows.

B.2 Algorithms and Related Derivations

Conditions

The penalty function $\rho(\cdot, \lambda)$ satisfies the following conditions:

- The function $\rho(\cdot, \lambda)$ satisfies $\rho(0, \lambda) = 0$ and is symmetric around zero.
- On the non negative real line, the function $\rho(\cdot, \lambda)$ is nondecreasing.
- For $t > 0$, the function $t \to \rho(\cdot, \lambda)/t$ is nonincreasing in t.
- The function $\rho(\cdot, \lambda)$ is differentiable for all $t \neq 0$ and subdifferentiable at $t = 0$, with $\lim_{t \to 0^+} \rho'(t, \lambda) = \lambda C$.
- There exists $\mu > 0$ such that $\rho_\mu(t, \lambda) = \rho(t, \lambda) + \frac{\mu}{2}t^2$ is convex.

Algorithm and derivation of the closed form solution for itarates in (3.19)

For each iteration k, we use the following notations:

$$
f(P^{(k)}) = \frac{1}{2}\|P^{(k)} - P_0\|,\ f_*(u^{(k)}, v^{(k)}, U^{(k)}) = -\frac{1}{2}\|u^{(k)}\mathbf{1}^t + \mathbf{1}(v^{(k)})^t - U^{(k)}\|_F^2 - tr((U^{(k)})^t P_0)
$$

.

Algorithm 8 Projection on doubly stochastic matrices

```
*input*:
$k_{max}, \epsilon \leftarrow$ *max. number of iteration and stopping criteria*
$U^{(0)} \in R_{+}^{p \times p}, u^{(0)}, v^{(0)} \in R^{p} \leftarrow$ *Initial dual variables*
*converged = FALSE*
*while converged == False  and* $k < k_{max}$:
    $U^{(k)} \leftarrow \max\{0, u^{(k-1)}\mathbf{1}^t + \mathbf{1}(v^{(k-1)})^t - P_0\}$
    $u^{(k)} \leftarrow \frac{1}{p}(P_0\mathbf{1} - ((v^{(k-1)})^t\mathbf{1} + 1)\mathbf{1} + U^{(k)}\mathbf{1})$
    $v^{(k)} \leftarrow \frac{1}{p}(P_0^t\mathbf{1} - ((u^{(k)})^t\mathbf{1} + 1)\mathbf{1} + (U^{(k)})^t\mathbf{1})$
    *Update primal variable:* $P^{(k)} = P_0 - u^{(k)}\mathbf{1}^t - \mathbf{1}(v^{(k)})^t + U^{(k)}$
    *If* $|f(P^{(k)}) - f_*(u^{(k)}, v^{(k)}, U^{(k)})| < \epsilon$
        *converged = TRUE*
    *else*
        $k = k + 1$
*Output*: *Doubly Stochastic Matrix* $P$
```

The **convergence of Algorithm 8 is guaranteed** since the objective function is differentiable and strictly concave in each block component when all other block components are held fixed (Bertsekas, 2015, Proposition 6.5.2).

Dual function derivation

The Lagrangian of (3.17) is (Bertsekas, 2015)

$$\begin{aligned}\mathcal{L}(P, u, v, U) = \frac{1}{2}\|P - P_0\|_F^2 &+ u^t(P\mathbf{1} - \mathbf{1}) \\ &+ v^t(P^t\mathbf{1} - \mathbf{1}) - tr(U^tP)\end{aligned}$$

The dual objective function is defined as

$$\mathcal{L}_*(u, v, U) = \inf_P \mathcal{L}(P, u, v, U) \tag{B.23}$$

Thus,

$$P = P_0 - u\mathbf{1}^t - \mathbf{1}v^t + U$$

Plugging this back into (B.23)

$$\begin{aligned}\mathcal{L}_*(u, v, U) = \frac{1}{2}\|u\mathbf{1}^t + \mathbf{1}v^t - U\|_F^2 &+ u^t(P_0\mathbf{1} - \mathbf{1}) + v^t(P_0^t\mathbf{1} - \mathbf{1}) - tr(U^t P_0) \\ &- tr(u^t u\mathbf{1}^t\,\mathbf{1} + u^t\,\mathbf{1}\,v^t\,\mathbf{1} + u^t U\,\mathbf{1} + v^t\,\mathbf{1}\,u^t\,\mathbf{1} + v^t v\,\mathbf{1}^t\,\mathbf{1} + v^t u^t\,\mathbf{1} \\ &- U^t u\,\mathbf{1}^t - U^t u\,\mathbf{1}^t - U^t\,\mathbf{1}\,v^t - U^t U)\end{aligned}$$

Then the (3.19) follows by noting that the expression in the trace function is equal to $\|u\mathbf{1}^t - \mathbf{1}v^t + U\|_F^2$. Now taking derivative with respect to u, v, U, the corresponding expressions for iterations in the Algorithm 8 follows.

B.3 Tuning Parameter Selection

We use extended BIC (eBIC) criterion (Foygel and Drton, 2010) for the tuning parameters $\theta = (\lambda, \gamma)$ selection in Algorithm 2. Ideally, we want to tune the parameters for each update of $\hat{L}^{(k)}$ in line 6, however, when the convergence takes more iteration this approach is computationally costly. In practice, the tuning parameters are selected before starting the iterations (Ye et al., 2020), hence a particular scoring function is fixed throughout the algorithm.

The eBIC criterion takes the form

$$\text{BIC}_{\gamma_{BIC}}(\mathcal{S}(L)) = -2\mathcal{L}_n(\hat{L}) + s\log n + 4s\gamma_{bic}\log p,$$

where $\mathcal{S}(L)$ is the support of matrix L, $s = |\mathcal{S}(L)|$, $\mathcal{L}_n(\hat{L})$ is the maximized log-likelihood function and $\gamma_{BIC} \in [0, 1]$. A larger value of γ_{BIC} results to a stronger penalization of L, and the case $\gamma_{BIC} = 0$ corresponds to the classical BIC. In general, the value of γ_{eBIC} is unknown, but relying on simulation results, authors suggest γ_{eBIC} as a candidate value.