Journal of Machine Learning Research 25 (2024) 1-36 Submitted 4/20; Published 9/24

Estimation of Sparse Gaussian Graphical Models with Hidden Clustering Structure

Meixia Lin MEIXIA_LIN@SUTD.EDU.SG
Engineering Systems and Design
Singapore University of Technology and Design
8 Somapah Road, Singapore, 487372

Defeng Sun DEFENG.SUN@POLYU.EDU.HK
Department of Applied Mathematics
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong

Kim-Chuan Toh MATTOHKC@NUS.EDU.SG
Department of Mathematics and Institute of Operations Research and Analytics
National University of Singapore
10 Lower Kent Ridge Road, Singapore, 119076

Chengjing Wang* RENASCENCEWANG@HOTMAIL.COM
School of Mathematics
Southwest Jiaotong University
Chengdu 611756, China

Editor: David Wipf

Abstract

Estimation of Gaussian graphical models is important in natural science when modeling the statistical relationships between variables in the form of a graph. The sparsity and clustering structure of the concentration matrix is enforced to reduce model complexity and describe inherent regularities. We propose a model to estimate the sparse Gaussian graphical models with hidden clustering structure, which also allows additional linear constraints to be imposed on the concentration matrix. We design an efficient two-phase algorithm for solving the proposed model. Specifically, we develop a symmetric Gauss-Seidel based alternating direction method of multipliers (sGS-ADMM) to generate an initial point to warm start the second phase algorithm, which is a proximal augmented Lagrangian method (pALM), to get a solution with high accuracy. Numerical experiments on both synthetic data and real data demonstrate the good performance of our model, as well as the efficiency and robustness of our proposed algorithm.

Keywords: sparse Gaussian graphical model, clustered lasso regularizer, proximal augmented Lagrangian method

1. Introduction

Let $z \in \mathbb{R}^n$ be a random vector following a multivariate Gaussian distribution $\mathcal{N}(0, \Sigma)$ with an unknown covariance matrix Σ. The Gaussian graphical model (Lauritzen, 1996) is

*. Corresponding author

commonly-used to estimate the concentration matrix Σ^{-1} from samples of z, which can be represented by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the vertices $\mathcal{V}$ contain n coordinates and the edges $\mathcal{E} = (e_{ij})_{1\le i<j\le n}$ describe the conditional independence relationships among $z_1, \cdots, z_n$. There is no edge between z_i and z_j if and only if $(\Sigma^{-1})_{ij} = 0$, which means that z_i and z_j are conditionally independent, given all the other variables. To detect nonzero elements in the concentration matrix Σ^{-1}, researchers have proposed sparse Gaussian graphical models (Yuan and Lin, 2007; Banerjee et al., 2008). Given a sample covariance matrix $C \in \mathbb{S}^n_+$, the sparse Gaussian graphical model attempts to estimate the concentration matrix $X^* := \Sigma^{-1}$ by solving the following ℓ_1-regularized log-likelihood minimization problem:

$$\min_{X\succeq 0} \Big\{ \langle C, X\rangle - \log\det(X) + \rho \sum_{i<j} |X_{ij}| \Big\}, \tag{1}$$

where ρ is a given positive parameter, $\langle C, X\rangle$ is the standard trace inner product between C and X, and $X \succeq 0$ means that $X \in \mathbb{S}^n$ is positive semidefinite. We adopt the convention that $\log 0 := -\infty$. The ℓ_1-norm penalty, which is motivated by the lasso idea (Tibshirani, 1996), enforces element-wise sparsity on X. There are many methods for solving the sparse Gaussian graphical model, such as the well-known GLasso algorithm (Friedman et al., 2008), the Newton-CG primal proximal point algorithm (Wang et al., 2010), and QUIC (Hsieh et al., 2014).

In the general setting of concentration matrix estimations with non-Gaussian variables, the model (1) is also called as the regularized log-determinant Bregman divergence model (Kulis et al., 2006; Davis et al., 2007; Dhillon and Tropp, 2008; Ravikumar et al., 2011), whose derivation is based on minimizing the Bregman divergence between the estimated concentration matrix and the true concentration matrix:

$$D(X\|\Sigma^{-1}) := -\log\det X + \log\det\Sigma^{-1} + \langle \Sigma, X - \Sigma^{-1}\rangle,$$

with the unknown true covariance matrix Σ replaced by the sample covariance matrix C. The edge weights learned by the log-determinant Bregman divergence model quantify the similarities between nodes. This is because the trace term can be written as the Laplacian quadratic form (Kalofolias, 2016; Dong et al., 2016; Kumar et al., 2019; Ying et al., 2020), which tends to assign a large weight between nodes if their signal values are similar to each other. Indeed, the similarities represented by the edges in the log-determinant Bregman divergence model can be seen as a generalization of the conditional dependence represented by the edges in the Gaussian graphical model. Furthermore, the model (1) and its variants has been widely-used to study the similarity graph of observations, see Kumar et al. (2020); Ying et al. (2020).

The concentration matrix may have additional structures other than sparsity. For example, Honorio et al. (2009) enforce the local constancy to find connectivities between two close or distant clusters of variables; Højsgaard and Lauritzen (2005, 2008a) propose the restricted concentration models where parameters associated with edges or vertices of the same class are restricted to being identical; Duchi et al. (2012) penalize certain groups of edges together. In all these models, the clustering structure of the edges or vertices is assumed to be known. However, in many real applications like the gene expression in cancer data (Hughes et al., 2000; Yu et al., 2017), the group/cluster information may be unknown in

advance. In order to deal with the unknown group assignments, some researchers construct hierarchical probabilistic models with variational and Bayesian methods to infer the group structure and estimate the concentration matrix. For example, Marlin and Murphy (2009) propose a two stage method, wherein a variational Bayes algorithm is proposed to learn the block structure in the first stage, and then the concentration matrix is estimated by using the block ℓ_1 method in the second stage; Ambroise et al. (2009) and Marlin et al. (2012) introduce latent variables with Laplace distributions as the priori information to indicate group assignments, and then perform an EM algorithm and a variational algorithm to learn the group structure and perform the estimation; Sun et al. (2014) propose a nonparametric Bayesian method which uses Chinese Restaurant Process and Wishart prior to model the group assignments and the concentration matrix, respectively, and adopts Gibbs sampling to estimate the posterior distribution of the group assignment variables; Sun et al. (2015) propose a generative model to describe graphical models on exponential families with soft clusters as well as overlapping blocks by applying an EM algorithm with variational inference. In addition to the hierarchical probabilistic models, Hosseini and Lee (2016) propose a non-convex optimization model to learn the group structure and the concentration matrix jointly, but it needs to know the number of clusters in advance.

Here we aim to propose a convex optimization model to estimate the sparse concentration matrix with hidden clustering structure. Note that in the context of a linear regression model where the regression coefficients are expected to be clustered into groups, the clustered lasso regularizer (Bondell and Reich, 2008; She, 2010; Petry et al., 2011; Lin et al., 2019) has been widely used. We borrow the idea of the regularization term to discover the sparsity and unknown clustering structure in the Gaussian graphical models. Thus we modify the sparse Gaussian graphical model (1) as follows:

$$\min_{X\succeq 0} \Big\{\langle C, X\rangle - \log\det(X) + \rho\sum_{i<j}|X_{ij}| + \lambda\sum_{i<j}\sum_{s<t}|X_{ij}-X_{st}|\Big\}, \tag{2}$$

where $\rho, \lambda > 0$ are given parameters. In the above model, the penalty on the pairwise differences is to force those entries of the concentration matrix associated with the same cluster of the edges to be the same. The idea of clustering the off-diagonal entries of the concentration matrix can be traced back to the work of Højsgaard and Lauritzen in (Højsgaard and Lauritzen, 2005, 2008a,b; Lauritzen and Højsgaard, 2008; Højsgaard, 2008). This line of research focuses on estimating the edge coloring graph via Gaussian graphical models, where the entries of the concentration matrix associated with the edges of the same color are restricted to be identical. As the number of ways of coloring edges in a given graph is enormous, the estimation of the concentration matrix has always been difficult. The additional clustered lasso regularzier on the off-diagonal entries will help us explore the unknown edge coloring structure without the need of brute-force search.

As pointed out in Friedman et al. (2001) and Dahl et al. (2008), the multivariate Gaussian distribution may be known to be Markov with respect to a given undirected network, that is, the sparsity pattern of Σ^{-1} may be a priori known in advance. In some more complicated cases, the conditional independence pattern may be partially known from some prior knowledge of the random variables, as stated in Lu (2010). In addition, in some applications such as the zero mean AR(k) process $\{Y_t\}$ with $Y_t = \sum_{j=1}^{k}\phi_j Y_{t-j} + \epsilon_t$, it is known that the concentration matrix Σ^{-1} is a bandlimited matrix with $(\Sigma^{-1})_{ij} = 0$ if $|i-j| > k$. To deal

with these cases, one can impose additional constraints on X to get the following model:

$$\min_{X\in\mathbb{S}^n}\Big\{\langle C,X\rangle-\log\det(X)+\rho\sum_{i<j}|X_{ij}|+\lambda\sum_{i<j}\sum_{s<t}|X_{ij}-X_{st}|\ \Big|\ X_{ij}=0,(i,j)\in\mathcal{J},X\succeq 0\Big\},\quad(3)$$

where $\mathcal{J}$ is the set of pairs of nodes (i,j) such that z_i and z_j are known to be conditionally independent.

Motivated by the above discussions, in this paper, we consider a more general problem which allows for general linear equality constraints to be imposed on X, that is,

$$\min_{X\in\mathbb{S}^n}\ \Big\{\langle C,X\rangle-\log\det(X)+\underbrace{\rho\sum_{i<j}|X_{ij}|+\lambda\sum_{i<j}\sum_{s<t}|X_{ij}-X_{st}|}_{Q(X)}\ \Big|\ \mathcal{A}X=b,\ X\succeq 0\Big\},\quad(\mathrm{P})$$

where $\mathcal{A}:\mathbb{S}^n\to\mathbb{R}^m$ is a given linear map, $b\in\mathbb{R}^m$ is a given vector, $\rho,\lambda>0$ are given parameters. Without loss of generality, we always assume that $\mathcal{A}:\mathbb{S}^n\to\mathbb{R}^m$ is surjective. Notice that by introducing the general linear constraints in (P), we can treat different linear constraints in a unified manner during the analysis and algorithm design. For example, it enables us to deal with the case when the conditional independence pattern is partially known (Friedman et al., 2001; Dahl et al., 2008; Lu, 2010) as stated in (3). Moreover, in some applications where the covariance matrix is known to have a Toeplitz structure, like the weakly-stationary continuous-time stochastic process, one knows that the entries of the concentration matrix must satisfy some additional constraints (Rodman and Shalom, 1992), for example, the ith row of the concentration matrix is equal to its $(n-i+1)$th column in reverse order for each $i=1,\ldots,n$. Another example comes from the matrix nearness problem (Kulis et al., 2006; Davis et al., 2007; Dhillon and Tropp, 2008), where we may have similarity constraints like $(e_i-e_j)^TX(e_i-e_j)\le u$ for (i,j) in a given index set $\mathcal{S}$ or dissimilarity constraints like $(e_i-e_j)^TX(e_i-e_j)\ge l$ for (i,j) in a given index set $\mathcal{D}$. Furthermore, more complicated linear constraints arising from various scenarios are also possible, such as 1) click-through feedback in unsupervised learning; 2) must-link and cannot-link constraints in semi-supervised learning; 3) points in the same class have "small" distance in supervised learning. Such additional constraints can then be handled by the general linear constraints in (P), with an additional nonnegative slack variable to handle inequality constraints when necessary.

Solving the problem (P) with a large n is a challenging task due to the combined effects of the $n\times n$ positive semidefinite variable, and the complicated regularization term, together with the linear constraints. At a first glance, it would appear to be extremely expensive to evaluate the second part of $Q(X)$ as it involves approximately $n^4/8$ terms, thus it becomes unthinkable to even compute the objective function value of (P) for the case when n is large. Fortunately, as we shall see later, the symmetric nature of the summation allows us to carry out the evaluation of the regularization term in $O(n^2\log n)$ operations. This reduction in the computation cost makes it possible to solve the problem (P) for large n.

Our contributions in this paper can be summarized in three parts.

1 Firstly, we propose the convex optimization model (P) to estimate the sparse Gaussian graphical model with hidden clustering structure, which also allows additional linear constraints to be imposed on the concentration matrix.

2 Secondly, we design an efficient two-phase algorithm for solving the dual of (P). We develop a symmetric Gauss-Seidel based alternating direction method of multipliers (sGS-ADMM) to generate an initial point to warm start the second phase algorithm, which is a proximal augmented Lagrangian method (pALM), to get a solution with high accuracy. For solving the pALM subproblems, we use the semismooth Newton method (SSN) where the sparsity and clustering structure is carefully analysed and exploited in the underlying generalized Jacobians to reduce the computational cost in each semismooth Newton iteration.

3 Thirdly, we conduct comprehensive numerical experiments on both synthetic data and real data to demonstrate the performance of our model comparing with the sparse Gaussian graphical model (Yuan and Lin, 2007; Banerjee et al., 2008) and Graphical models with overlapping blocks (GRAB) (Hosseini and Lee, 2016), as well as the efficiency and robustness of our proposed algorithm comparing with the standalone sGS-ADMM proposed in Section 3.1 and the alternating linearization method proposed in Scheinberg et al. (2010). The numerical results show that our model can rather successfully estimate the concentration matrix as well as uncovering its clustering structure.

The remaining parts of the paper are organized as follows. In Section 2, we state the problem setup, derive the asymptotic property of the proposed estimator and discuss some useful results associated with the problem (P). In Section 3, we describe the proposed two-phase algorithm for solving our model. In Section 4, we present the numerical results on both synthetic data and real data. Finally, in Section 5, we make some concluding remarks.

Notation. Throughout the paper, we use $\mathrm{diag}(X)$ to denote a vector in $\mathbb{R}^n$ consisting of the diagonal entries of a matrix $X \in \mathbb{R}^{n\times n}$, and $\mathrm{Diag}(x)$ to denote a diagonal matrix in $\mathbb{R}^{n\times n}$ whose diagonal is given by a vector $x \in \mathbb{R}^n$. For $Z \in \mathbb{R}^{m\times n}$, $\|Z\|$ denotes its Frobenius norm, and Z_i denotes its i-th row. For any symmetric matrix Y, $\sigma_{\min}(Y)$ denotes the smallest eigenvalue of Y.

2. Problem Setup and Preliminaries

In this section, we set up the problem, analyse the asymptotic property of the proposed estimator, and then derive some useful results of the regularization term $Q(\cdot)$ and the function $\log\det(\cdot)$, respectively. Note that these properties play an important role in the algorithm design which will be presented later.

2.1 Asymptotic Property of the Proposed Estimator

In this subsection, we derive the asymptotic properties of the proposed (unconstrained) estimator (2), which are analogous to those for the lasso (Fu and Knight, 2000) and the sparse Gaussian graphical model (Yuan and Lin, 2007). For simplicity, we consider the case when the dimension n is fixed, and the sample size $p \to \infty$. The following theorem establishes some properties of the proposed estimator (2).

Theorem 1 *Let $z \in \mathbb{R}^n$ be a random vector following a multivariate Gaussian distribution $\mathcal{N}(0,\Sigma)$ with an invertible covariance matrix Σ. Set $X^* = \Sigma^{-1}$. If the sequences $\{\rho_p\}$ and $\{\lambda_p\}$ satisfy $\sqrt{p}\rho_p \to \rho_0$ and $\sqrt{p}\lambda_p \to \lambda_0$ as $p \to \infty$, the estimator $\widehat{X}_p$, that is, the optimal*

solution to (2) *with the parameters* (ρ_p, λ_p)*, satisfies*

$$\sqrt{p}(\widehat{X}_p - X^*) \to \mathop{\arg\min}_{V\in\mathbb{S}^n} \Upsilon(V)$$

in distribution as $p \to \infty$*, where*

$$\Upsilon(V) = \frac{1}{2}\langle \Sigma V, V\Sigma\rangle + \langle V, W\rangle + \rho_0 \sum_{i<j}\Big\{V_{ij}\mathrm{sign}(X^*_{ij})\mathbf{1}(X^*_{ij}\neq 0) + |V_{ij}|\mathbf{1}(X^*_{ij}=0)\Big\}$$
$$+ \lambda_0 \sum_{i<j}\sum_{s<t}\Big\{(V_{ij}-V_{st})\mathrm{sign}(X^*_{ij}-X^*_{st})\mathbf{1}(X^*_{ij}\neq X^*_{st}) + |V_{ij}-V_{st}|\mathbf{1}(X^*_{ij}=X^*_{st})\Big\},$$

in which $\mathbf{1}(\text{Event B}) = 1$ *if event B happens, and* $\mathbf{1}(\text{Event B}) = 0$ *otherwise. Here* W *is a random matrix in* $\mathbb{S}^n$ *such that its vectorized form* $\mathrm{vec}(W)$ *satisfies* $\mathrm{vec}(W) \sim \mathcal{N}(0, \Lambda)$ *with* $\Lambda_{(i,j),(s,t)} = \mathrm{cov}(W_{ij}, W_{st}) = \mathrm{cov}(z_i z_j, z_s z_t)$*, for any* $i, j, s, t = 1, \cdots, n$*.*

Proof For any $p = 1, 2, \cdots$, define the function $\Upsilon_p : \mathbb{S}^n \to \mathbb{R}$ as

$$\Upsilon_p(V) = \left\langle C, X^* + \frac{V}{\sqrt{p}}\right\rangle - \log\det\left(X^* + \frac{V}{\sqrt{p}}\right) + \rho_p \sum_{i<j}\left|X^*_{ij} + \frac{V_{ij}}{\sqrt{p}}\right|$$
$$+\lambda_p\sum_{i<j}\sum_{s<t}\left|X^*_{ij} + \frac{V_{ij}}{\sqrt{p}} - X^*_{st} - \frac{V_{st}}{\sqrt{p}}\right| - \langle C, X^*\rangle + \log\det(X^*) - \rho_p\sum_{i<j}|X^*_{ij}| - \lambda_p\sum_{i<j}\sum_{s<t}|X^*_{ij} - X^*_{st}|.$$

Then the unique minimizer V_p^* of the above convex function satisfies $\widehat{X}_p = X^* + \frac{V_p^*}{\sqrt{p}}$. Therefore, we have $V_p^* = \sqrt{p}(\widehat{X}_p - X^*)$ minimizes the function $\Upsilon_p(V)$. Note that

$$\left\langle C, X^* + \frac{V}{\sqrt{p}}\right\rangle - \langle C, X^*\rangle = \left\langle C, \frac{V}{\sqrt{p}}\right\rangle = \frac{1}{\sqrt{p}}\langle \Sigma, V\rangle + \frac{1}{\sqrt{p}}\langle C - \Sigma, V\rangle,$$

and

$$-\log\det\left(X^* + \frac{V}{\sqrt{p}}\right) + \log\det(X^*) = -\log\det\left(I + \frac{\Sigma^{1/2}V\Sigma^{1/2}}{\sqrt{p}}\right) = -\frac{1}{\sqrt{p}}\langle\Sigma, V\rangle + \frac{1}{2p}\langle \Sigma V, V\Sigma\rangle + o(\frac{1}{p}),$$

where the last equality follows from the fact that

$$\log\det\left(I + \frac{\Sigma^{1/2}V\Sigma^{1/2}}{\sqrt{p}}\right) = \sum_{i=1}^n \log\left(1 + \frac{\sigma_i(\Sigma^{1/2}V\Sigma^{1/2})}{\sqrt{p}}\right)$$
$$= \sum_{i=1}^n \frac{\sigma_i(\Sigma^{1/2}V\Sigma^{1/2})}{\sqrt{p}} - \sum_{i=1}^n \frac{\sigma_i^2(\Sigma^{1/2}V\Sigma^{1/2})}{2p} + o(\frac{1}{p})$$
$$= \frac{1}{\sqrt{p}}\mathrm{tr}(\Sigma^{1/2}V\Sigma^{1/2}) - \frac{1}{2p}\mathrm{tr}(\Sigma^{1/2}V\Sigma V\Sigma^{1/2}) + o(\frac{1}{p}) = \frac{1}{\sqrt{p}}\langle\Sigma, V\rangle - \frac{1}{2p}\langle\Sigma V, V\Sigma\rangle + o(\frac{1}{p}),$$

where $\{\sigma_i(X)\}_{i=1}^n$ denotes the eigenvalues of $X \in \mathbb{S}^n$. Moreover, when $p \to \infty$, we have

$$p\left(\rho_p\sum_{i<j}\left|X^*_{ij} + \frac{V_{ij}}{\sqrt{p}}\right| - \rho_p\sum_{i<j}|X^*_{ij}|\right) \to \rho_0\sum_{i<j}\Big\{V_{ij}\mathrm{sign}(X^*_{ij})\mathbf{1}(X^*_{ij}\neq 0) + |V_{ij}|\mathbf{1}(X^*_{ij}=0)\Big\},$$

and

$$p\left(\lambda_p\sum_{i<j}\sum_{s<t}\left|X^*_{ij}+\frac{V_{ij}}{\sqrt{p}}-X^*_{st}-\frac{V_{st}}{\sqrt{p}}\right|-\lambda_p\sum_{i<j}\sum_{s<t}|X^*_{ij}-X^*_{st}|\right)$$
$$\to\lambda_0\sum_{i<j}\sum_{s<t}\Big\{(V_{ij}-V_{st})\mathrm{sign}(X^*_{ij}-X^*_{st})\mathbf{1}(X^*_{ij}\neq X^*_{st})+|V_{ij}-V_{st}|\mathbf{1}(X^*_{ij}=X^*_{st})\Big\}.$$

Therefore, combining the above results, we have

$$p\Upsilon_p(V)=\sqrt{p}\langle C-\Sigma,V\rangle+\frac{1}{2}\langle\Sigma V,V\Sigma\rangle+o(1)+p\left(\rho_p\sum_{i<j}\left|X^*_{ij}+\frac{V_{ij}}{\sqrt{p}}\right|-\rho_p\sum_{i<j}|X^*_{ij}|\right)$$
$$+p\left(\lambda_p\sum_{i<j}\sum_{s<t}\left|X^*_{ij}+\frac{V_{ij}}{\sqrt{p}}-X^*_{st}-\frac{V_{st}}{\sqrt{p}}\right|-\lambda_p\sum_{i<j}\sum_{s<t}|X^*_{ij}-X^*_{st}|\right).$$

By noting that $W_p:=\sqrt{p}(C-\Sigma)\to\mathcal{N}(0,\Lambda)$, we have $p\Upsilon_p(V)\to\Upsilon(V)$ in distribution as $p\to\infty$. Furthermore, since $\Upsilon(\cdot)$, $\Upsilon_p(\cdot)$ are convex and $\Upsilon(\cdot)$ admits a unique minimizer V^*_p, it follows from Geyer (1994) that

$$V^*_p=\sqrt{p}(\widehat{X}_p-X^*)=\mathop{\arg\min}_{V\in\mathbb{S}^n}p\Upsilon_p(V)\to\mathop{\arg\min}_{V\in\mathbb{S}^n}\Upsilon(V),$$

in distribution as $p\to\infty$. This completes the proof. ■

Before studying the regularization term $Q(\cdot)$ and the function $\log\det(\cdot)$, we give the definition of the proximal mapping and the Moreau envelope. Let $f:\mathcal{H}\to\mathbb{R}$ be a given closed convex function, where $\mathcal{H}$ is a finite dimensional real Euclidean space equipped with an inner product $\langle\cdot,\cdot\rangle$ and its induced norm $\|\cdot\|$. The Fenchel conjugate f^* of the function f is then defined as

$$f^*(u):=\sup_{x\in\mathcal{H}}\{\langle x,u\rangle-f(x)\},\quad u\in\mathcal{H}.\tag{4}$$

The Moreau envelope of f at $x\in\mathcal{H}$ is defined as

$$\mathrm{E}_f(x):=\min_{y\in\mathcal{H}}\ \Big\{\frac{1}{2}\|y-x\|^2+f(y)\Big\},$$

whose corresponding unique minimizer, which is called the proximal mapping of f at x, is denoted as $\mathrm{Prox}_f(x)$. It is proved in Moreau (1965); Rockafellar (1976) that $\mathrm{Prox}_f(\cdot)$ is globally Lipschitz continuous with modulus 1 and $\mathrm{E}_f(\cdot)$ is finite-valued, convex and continuously differentiable with

$$\nabla\mathrm{E}_f(x)=x-\mathrm{Prox}_f(x).$$

The Moreau identity states that for any $t>0$, it holds that

$$\mathrm{Prox}_{tf}(x)+t\mathrm{Prox}_{f^*/t}(x/t)=x,\quad\forall x\in\mathcal{H},$$

where f^* is the Fenchel conjugate of f defined in (4).

2.2 Duality and Optimality Conditions

The minimization form for the dual of (P) is given by

$$\begin{aligned} \min_{y\in\mathbb{R}^m, Z\in\mathbb{S}^n, S\in\mathbb{S}^n} \quad & -\langle b, y\rangle - \log\det(Z) + Q^*(-S) - n \\ \text{s.t.} \quad & C - \mathcal{A}^*y - Z - S = 0, \quad Z \succeq 0, \end{aligned} \tag{D}$$

where $\mathcal{A}^* : \mathbb{R}^m \to \mathbb{S}^n$ is the adjoint map of $\mathcal{A}$, Q^* is the Fenchel conjugate of Q. Note that the linear maps $\mathcal{A}$ and $\mathcal{A}^*$ can be expressed as

$$\mathcal{A}X = [\langle A_1, X\rangle, \cdots \langle A_m, X\rangle]^T, \quad \mathcal{A}^*y = \sum_{k=1}^m y_k A_k,$$

where $A_1, A_2, \cdots, A_m$ are given matrices in $\mathbb{S}^n$.

Remark 2 *For better illustration of the linear maps $\mathcal{A}$ and $\mathcal{A}^*$, we give a simple example where the conditional independence pattern is partially known, that is, the feasible set of* (P) *is*

$$\mathcal{F}_{\mathcal{J}} := \{X \in \mathbb{S}^n \mid X_{ij} = 0,\ (i,j) \in \mathcal{J},\ X \succeq 0\},$$

where the set of pairs of nodes $\mathcal{J} := \{(i_k, j_k)\}_{k=1}^m$. Without loss of generality, we assume $i_k < j_k$ for all $k = 1, \cdots, m$. Define the linear map $\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m$ as $\mathcal{A}X = [X_{i_1 j_1}, \cdots, X_{i_m j_m}]^T$. Then we have $\mathcal{F}_{\mathcal{J}} = \{X \in \mathbb{S}^n \mid \mathcal{A}X = 0,\ X \succeq 0\}$. By the definition of the adjoint map, we know that the linear map $\mathcal{A}^* : \mathbb{R}^m \to \mathbb{S}^n$ is given as follows: for any $y \in \mathbb{R}^m$, $\mathcal{A}^*y \in \mathbb{S}^n$ satisfies $(\mathcal{A}^*y)_{i_k j_k} = (\mathcal{A}^*y)_{j_k i_k} = y_k/2$, $k = 1, \cdots, m$ and $(\mathcal{A}^*y)_{ij} = 0$ otherwise. In addition, it can be seen that $\mathcal{A}\mathcal{A}^* = I_m/2$. This example indicates that when the conditional independence pattern is partially known, the general linear constraints will not increase the difficulty in our computation, but help us to unify the mathematical notations during the analysis and the algorithm design.

The Karush-Kuhn-Tucker (KKT) conditions associated with (P) and (D) are given as

$$\begin{cases} C - \mathcal{A}^*y - Z - S = 0, \\ XZ = I_n,\ Z \succeq 0,\ X \succeq 0, \\ 0 \in \partial Q(X) + S, \\ \mathcal{A}X = b. \end{cases} \tag{5}$$

Throughout this paper, we make the following blanket assumption.

Assumption 1 *The problem* (P) *admits an optimal solution X^*.*

Under Assumption 1, we can see that X^* must be the unique minimizer of (P), since the objective function of (P) is strictly convex with respect to X. In addition, the assumption also implies that the constraint qualification holds, that is, there exists $X_0 \in \mathbb{S}^n_{++}$ such that $\mathcal{A}X_0 = b$. According to Rockafellar (1997, Corollary 28.2.2 and Corollary 28.3.1), we have that the set

$$\Omega(X^*) := \{(y, Z, S) \in \mathbb{R}^m \times \mathbb{S}^n \times \mathbb{S}^n \mid (X^*, y, Z, S) \text{ satisfies the KKT system (5)}\}$$

is nonempty. Moreover, any $(y, Z, S) \in \Omega(X^*)$ is an optimal solution to (D).

2.3 Properties of the Regularization Term $Q(\cdot)$

Let $\mathcal{B} : \mathbb{S}^n \to \mathbb{R}^{\bar{n}}$ be the linear map such that $\mathcal{B}X$ is the vector obtained from $X \in \mathbb{S}^n$ by concatenating the columns of the strictly upper triangular part of X sequentially into a vector of dimension $\bar{n} := n(n-1)/2$. The adjoint $\mathcal{B}^* : \mathbb{R}^{\bar{n}} \to \mathbb{S}^n$ is such that $\mathcal{B}^*x$ is the operation of first putting the entries of the vector $x \in \mathbb{R}^{\bar{n}}$ into the strictly upper triangular part of an $n \times n$ matrix X, and then symmetrizing it. That is, for any $X \in \mathbb{S}^n$,

$$\mathcal{B}X = [X_{12}, X_{13}, X_{23}, \cdots, X_{1n}, \cdots, X_{n-1,n}]^T \in \mathbb{R}^{\bar{n}},$$

and for any $x \in \mathbb{R}^{\bar{n}}$,

$$\mathcal{B}^*x = \frac{1}{2}\begin{pmatrix} 0 & x_1 & x_2 & \cdots & x_{\bar{n}-n+2} \\ x_1 & 0 & x_3 & \cdots & x_{\bar{n}-n+3} \\ x_2 & x_3 & 0 & \cdots & x_{\bar{n}-n+4} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{\bar{n}-2n+4} & x_{\bar{n}-2n+5} & x_{\bar{n}-2n+6} & \cdots & x_{\bar{n}} \\ x_{\bar{n}-n+2} & x_{\bar{n}-n+3} & x_{\bar{n}-n+4} & \cdots & 0 \end{pmatrix} \in \mathbb{S}^n.$$

Denote

$$q(x) = \rho\|x\|_1 + \lambda p(x), \quad p(x) = \sum_{1 \le k < l \le \bar{n}} |x_k - x_l|, \quad \forall x \in \mathbb{R}^{\bar{n}}.$$

Based on the linear map $\mathcal{B}$, we could see that

$$Q(X) = q(\mathcal{B}X), \quad \forall X \in \mathbb{S}^n.$$

Note that the function $q(\cdot)$ is the clustered lasso regularizer in the context of the linear regression models, which has been studied in Bondell and Reich (2008); She (2010); Petry et al. (2011); Lin et al. (2019).

Before going into details of the properties of the regularization term $Q(\cdot)$, we first present the properties of the clustered lasso regularizer $q(\cdot)$. The following proposition provides an explicit formula of $\mathrm{Prox}_q(\cdot)$ as well as the formula of $q^*(\cdot)$. Note that the proposition enables us to compute $\mathrm{Prox}_q(y)$ in $O(\bar{n}\log(\bar{n}))$ operations. The results in the parts (a) and (b) follow from Lin et al. (2019) and the proof of the part (c) is given in the appendix.

Proposition 3 *Denote the convex set $\mathcal{D} = \{x \in \mathbb{R}^{\bar{n}} \mid Bx \ge 0\}$, where the matrix $B \in \mathbb{R}^{(\bar{n}-1)\times\bar{n}}$ is defined as $Bx = [x_1 - x_2, x_2 - x_3, \cdots, x_{\bar{n}-1} - x_{\bar{n}}]^T \in \mathbb{R}^{\bar{n}-1}$ for any $x \in \mathbb{R}^{\bar{n}}$. Define $w \in \mathbb{R}^{\bar{n}}$ as $w_k = \bar{n} - 2k + 1$, $k = 1, \ldots, \bar{n}$. For any $y \in \mathbb{R}^{\bar{n}}$, let $P_y \in \mathbb{R}^{\bar{n}\times\bar{n}}$ be a permutation matrix such that $P_y y$ is sorted in a non-increasing order. Then for any $y \in \mathbb{R}^{\bar{n}}$, the following statements hold.*

(a) The computational cost of evaluating $p(y)$ can be reduced from $O(\bar{n}^2)$ to $O(\bar{n}\log\bar{n})$ as

$$p(y) = \sum_{1 \le k < l \le \bar{n}} |y_k - y_l| = \langle w, P_y y \rangle. \tag{6}$$

(b) The proximal mapping of q at y can be computed as

$$\mathrm{Prox}_q(y) = \mathrm{Prox}_{\rho\|\cdot\|_1}(\mathrm{Prox}_{\lambda p}(y)) = \mathrm{sign}(\mathrm{Prox}_{\lambda p}(y)) \circ \max(|\mathrm{Prox}_{\lambda p}(y)| - \rho, 0), \quad (7)$$

where $\mathrm{sign}(\cdot)$, $|\cdot|$ *and* $\max(\cdot,\cdot)$ *are taken component-wise, and*

$$\mathrm{Prox}_{\lambda p}(y) = P_y^T \Pi_{\mathcal{D}}(P_y y - \lambda w).$$

Here $\Pi_{\mathcal{D}}(\cdot)$ *(the metric projection onto* $\mathcal{D}$*) can be computed by the pool-adjacent-violators algorithm (Best and Chakravarti, 1990) in* $O(\bar{n})$ *operations.*

(c) Moreover, for any $u \in \mathbb{R}^{\bar{n}}$, the Fenchel conjugate q^* at u is

$$q^*(u) = \begin{cases} 0, & \text{if } \sum_{i=1}^{k}((P_u u - \lambda w)_i - \rho) \le 0, \sum_{i=k}^{n}((P_u u - \lambda w)_i + \rho) \ge 0, \forall k = 1, \cdots, \bar{n}, \\ +\infty, & \text{otherwise.} \end{cases} \quad (8)$$

In order to design second-order algorithms for solving the problems involving the clustered lasso regularizer $q(\cdot)$, we need the generalized Jacobian of $\mathrm{Prox}_q(\cdot)$. However, the Clarke generalized Jacobian $\partial \mathrm{Prox}_q(\cdot)$ is not easily computable. Fortunately, we can define the following multifunction $\mathcal{M}$ which could be viewed as the generalized Jacobian of $\mathrm{Prox}_q(\cdot)$. In particular, following the idea in Lin et al. (2019), we define the multifunction $\mathcal{M} : \mathbb{R}^{\bar{n}} \rightrightarrows \mathbb{R}^{\bar{n}\times\bar{n}}$ by

$$\mathcal{M}(y) = \Big\{ M \in \mathbb{S}^{\bar{n}} \mid M = \Theta P_y^T \widehat{Q} P_y, \Theta \in \partial_B \mathrm{Prox}_{\rho\|\cdot\|_1}(\mathrm{Prox}_{\lambda p}(y)), \widehat{Q} \in \partial_{\mathrm{HS}} \Pi_{\mathcal{D}}(P_y y - \lambda w) \Big\}, \quad (9)$$

where $\partial_B \mathrm{Prox}_{\rho\|\cdot\|_1}(\cdot)$ is the B-subdifferential of $\mathrm{Prox}_{\rho\|\cdot\|_1}(\cdot)$ and $\partial_{\mathrm{HS}}\Pi_{\mathcal{D}}(\cdot)$ is the HS-Jacobian (Han and Sun, 1997) of $\Pi_{\mathcal{D}}(\cdot)$. In the implementation of our proposed algorithm later, we need an explicitly computable element in $\mathcal{M}(y)$ for any given $y \in \mathbb{R}^{\bar{n}}$, which is provided in the following proposition.

Proposition 4 *For any $y \in \mathbb{R}^{\bar{n}}$, define $\Sigma_y := \mathrm{Diag}(\sigma) \in \mathbb{R}^{(\bar{n}-1)\times(\bar{n}-1)}$ with $\sigma_i = 1$ if $B_i \Pi_{\mathcal{D}}(P_y y - \lambda w) = 0$ and $\sigma_i = 0$ otherwise. Also define $\Theta_y := \mathrm{Diag}(\theta) \in \mathbb{R}^{\bar{n}\times\bar{n}}$ with $\theta_i = 0$ if $|\mathrm{Prox}_{\lambda p}(y)|_i \le \rho$ and $\theta_i = 1$ otherwise. Then it holds that*

$$W_y := \Theta_y P_y^T (I_{\bar{n}} - B^T (\Sigma_y B B^T \Sigma_y)^\dagger B) P_y \in \mathcal{M}(y),$$

where $(\cdot)^\dagger$ denotes the pseduoinverse.

Remark 5 *Further details on the computation of W_y given in the previous proposition could be found in Lin et al. (2019, Proposition 2.8).*

Based on the results associated with the clustered lasso regularizer $q(\cdot)$, we could study the properties of the function $Q(\cdot)$ in (P), which are summarized in the following proposition.

Proposition 6 *For any $Y \in \mathbb{S}^n$, the following statements hold.*

(a) The computational cost of evaluating $Q(Y)$ can be reduced from $O(n^4)$ to $O(n^2 \log n)$ as

$$Q(Y) = q(\mathcal{B}Y) = \rho\|\mathcal{B}Y\|_1 + \lambda\langle w, P_{\mathcal{B}Y}(\mathcal{B}Y)\rangle.$$

(b) The Fenchel conjugate Q^* at Y could be computed as

$$Q^*(Y) = \begin{cases} q^*(2\mathcal{B}Y), & \text{if } \mathrm{diag}(Y) = 0, \\ +\infty, & \text{otherwise,} \end{cases}$$

where the formula of $q^*(\cdot)$ could be found in (8).

(c) The proximal mapping $\mathrm{Prox}_Q(Y)$ could be computed as

$$\mathrm{Prox}_Q(Y) = \mathrm{Diag}\big(\mathrm{diag}(Y)\big) + \mathcal{B}^*\mathrm{Prox}_q(2\mathcal{B}Y), \tag{10}$$

where the explicit formula of $\mathrm{Prox}_q(\cdot)$ could be found in (7).

Proof (a) The reformulation of the clustered lasso regularizer as a weighted ordered-lasso regularizer in (6) enables us to evaluate the function value $Q(Y)$ in $O(n^2 \log n)$ operations instead of $O(n^4)$ operations.
(b) By the definition of $Q^*(\cdot)$, we can see that

$$\begin{aligned} Q^*(Y) &= \sup_{X\in\mathbb{S}^n} \Big\{\langle X, Y\rangle - Q(X)\Big\} \\ &= \sup_{X\in\mathbb{S}^n} \Big\{\sum_{i=1}^n X_{ii}Y_{ii} + 2\langle \mathcal{B}X, \mathcal{B}Y\rangle - q(\mathcal{B}X)\Big\} = \begin{cases} q^*(2\mathcal{B}Y), & \text{if } \mathrm{diag}(Y) = 0, \\ +\infty, & \text{otherwise.} \end{cases} \end{aligned}$$

(c) According to the definition of $\mathrm{Prox}_Q(\cdot)$, it can be seen that

$$\begin{aligned} \mathrm{Prox}_Q(Y) &= \arg\min_{X\in\mathbb{S}^n} \Big\{\frac{1}{2}\|\mathrm{diag}(X) - \mathrm{diag}(Y)\|^2 + \|\mathcal{B}X - \mathcal{B}Y\|^2 + q(\mathcal{B}X)\Big\} \\ &= \mathrm{Diag}\big(\mathrm{diag}(Y)\big) + \mathcal{B}^*\mathrm{Prox}_q(2\mathcal{B}Y). \end{aligned}$$

This completes the proof. ■

To design second-order algorithms, we need the generalized Jacobian of $\mathrm{Prox}_Q(\cdot)$. Note that according to Hiriart-Urruty et al. (1984, Example 2.5), for any $Y \in \mathbb{S}^n$, it holds that

$$\partial\mathrm{Prox}_Q(Y)[H] = \mathrm{Diag}\big(\mathrm{diag}(H)\big) + 2\mathcal{B}^*\partial\mathrm{Prox}_q(2\mathcal{B}Y)[\mathcal{B}H], \quad \forall H \in \mathbb{S}^n, \tag{11}$$

where $\partial\mathrm{Prox}_Q(Y)$ is the Clarke generalized Jacobian of $\mathrm{Prox}_Q(\cdot)$ at Y, $\partial\mathrm{Prox}_q(2\mathcal{B}Y)$ is the Clarke generalized Jacobian of $\mathrm{Prox}_q(\cdot)$ at $2\mathcal{B}Y$. As already mentioned, though the Clarke generalized Jacobian $\partial\mathrm{Prox}_q(\cdot)$ is not easily computable, we can use the multifunction $\mathcal{M}$ defined in (9) as the proxy of the Clarke generalized Jacobian of $\mathrm{Prox}_q(\cdot)$. Inspired by the relationship in (11), we provide a computable surrogate of $\partial\mathrm{Prox}_Q(\cdot)$ in the following proposition.

Proposition 7 *For any $Y \in \mathbb{S}^n$, define the generalized Jacobian $\hat{\partial}\mathrm{Prox}_Q(Y) : \mathbb{S}^n \rightrightarrows \mathbb{S}^n$ as follows: $\mathcal{P} \in \hat{\partial}\mathrm{Prox}_Q(Y)$ if and only if there exists $M \in \mathcal{M}(2\mathcal{B}Y)$ such that*

$$\mathcal{P}H = \mathrm{Diag}\big(\mathrm{diag}(H)\big) + 2\mathcal{B}^* M[\mathcal{B}H], \quad \forall H \in \mathbb{S}^n.$$

Then the multifunction $\hat{\partial}\mathrm{Prox}_Q(\cdot)$ is nonempty, compact, and upper-semicontinuous. In addition, $\mathrm{Prox}_Q(\cdot)$ is strongly semismooth with respect to $\hat{\partial}\mathrm{Prox}_Q(\cdot)$.

Proof According to Lin et al. (2019, Theorem 2.10), we know that the multifunction $\mathcal{M}$ defined in (9) is nonempty, compact, and upper-semicontinuous. Thus the multifunction $\hat{\partial}\mathrm{Prox}_Q(\cdot)$ is also nonempty, compact, and upper-semicontinuous. In addition, we can see that $\mathrm{Prox}_Q(\cdot)$ is directionally differentiable. Given $Y \in \mathbb{S}^n$, for any $\Delta Y \in \mathbb{S}^n$ with $\|\Delta Y\|$ sufficiently small, by the strong semismoothness of $\mathrm{Prox}_q(\cdot)$ with respect to $\mathcal{M}$ (Lin et al., 2019, Theorem 2.10), we have that

$$\mathrm{Prox}_q(2\mathcal{B}Y + 2\mathcal{B}\Delta Y) - \mathrm{Prox}_q(2\mathcal{B}Y) - M[2\mathcal{B}\Delta Y] = 0, \quad \forall M \in \mathcal{M}(2\mathcal{B}Y + 2\mathcal{B}\Delta Y).$$

Therefore, for any $\mathcal{H} \in \hat{\partial}\mathrm{Prox}_Q(Y + \Delta Y)$ with $\Delta Y \in \mathbb{S}^n$ sufficiently small, it holds that

$$\begin{aligned}
&\mathrm{Prox}_Q(Y + \Delta Y) - \mathrm{Prox}_Q(Y) - \mathcal{H}[\Delta Y] \\
&= \mathrm{Diag}(\mathrm{diag}(\Delta Y)) + \mathcal{B}^*\Big(\mathrm{Prox}_q(2\mathcal{B}Y + 2\mathcal{B}\Delta Y) - \mathrm{Prox}_q(2\mathcal{B}Y)\Big) - \mathcal{H}[\Delta Y].
\end{aligned}$$

By the definition of $\hat{\partial}\mathrm{Prox}_Q(\cdot)$, there must exist $M \in \mathcal{M}(2\mathcal{B}Y + 2\mathcal{B}\Delta Y)$ such that

$$\mathcal{H}[\Delta Y] = \mathrm{Diag}(\mathrm{diag}(\Delta Y)) + 2\mathcal{B}^* M[\mathcal{B}\Delta Y].$$

Thus, for any $\Delta Y \in \mathbb{S}^n$ with $\|\Delta Y\|$ sufficiently small we have

$$\begin{aligned}
&\mathrm{Prox}_Q(Y + \Delta Y) - \mathrm{Prox}_Q(Y) - \mathcal{H}[\Delta Y] \\
&= \mathcal{B}^*\Big(\mathrm{Prox}_q(2\mathcal{B}Y + 2\mathcal{B}\Delta Y) - \mathrm{Prox}_q(2\mathcal{B}Y) - M[2\mathcal{B}\Delta Y]\Big) = 0,
\end{aligned}$$

which means that $\mathrm{Prox}_Q(\cdot)$ is strongly semismooth with respect to $\hat{\partial}\mathrm{Prox}_Q(\cdot)$. ■

2.4 Properties of the $\log\det(\cdot)$ Function

The following proposition states the computation of the proximal mapping of $-\log\det(\cdot)$ and the corresponding Jacobian, which is obtained from Wang et al. (2010, Lemma 2.1). For simplicity, we denote $r(X) := -\log\det(X)$ for any $X \succeq 0$.

Proposition 8 *For any given $X \in \mathbb{S}^n$, with its eigenvalue decomposition $X = P\mathrm{Diag}(d)P^T$, where d is the vector of eigenvalues and the columns of P are the corresponding orthonormal set of eigenvectors. We assume that $d_1 \geq \cdots \geq d_t > 0 \geq d_{t+1} \geq \cdots d_n$. Given $\mu > 0$ and the scaler function $\phi_\mu^+(x) := (\sqrt{x^2 + 4\mu} + x)/2$ for all $x \in \mathbb{R}$, we define its matrix counterpart:*

$$\phi_\mu^+(X) := P\mathrm{Diag}(\phi_\mu^+(d))P^T, \tag{12}$$

where $\phi_\mu^+(d) \in \mathbb{R}^n$ is such that its i-th component is given by $\phi_\mu^+(d_i)$.

(a) The proximal mapping of $\mu r(\cdot)$ can be computed as

$$\mathrm{Prox}_{\mu r}(X) = \phi_\mu^+(X). \tag{13}$$

(b) ϕ_μ^+ is continuously differentiable and its Fréchet derivative $(\phi_\mu^+)'(X)$ at X is given by

$$(\phi_\mu^+)'(X)[H] = P(\Omega \circ (P^T H P))P^T, \quad \forall H \in \mathbb{S}^n,$$

where $\Omega \in \mathbb{S}^n$ is defined by

$$\Omega_{ij} = \frac{\phi_\mu^+(d_i) + \phi_\mu^+(d_j)}{\sqrt{d_i^2 + 4\mu} + \sqrt{d_j^2 + 4\mu}}, \quad i, j = 1, \cdots, n.$$

3. A Two-phase Algorithm

In this section, we propose a two-phase algorithm to solve the problem (P) based on the augmented Lagrangian function of (D). In Phase I, we design a symmetric Gauss-Seidel based alternating direction method of multipliers to solve the problem to a moderate level of accuracy. In Phase II, we employ a proximal augmented Lagrangian method with its subproblems solved by the SSN method to get a solution with high accuracy. Note that the sGS-ADMM not only can be used to generate a good initial point to warm start the pALM, it can also be used alone to solve the problem. But as a first-order method, the sGS-ADMM may not be efficient enough in some cases to solve a problem to high accuracy. Thus in the second phase, we switch to the superlinearly convergent pALM to compute an accurate solution.

3.1 Phase I: sGS-ADMM

A natural way to solve the problem (D) is the popular alternating direction method of multipliers (ADMM), but as shown via a counterexample in Chen et al. (2016), the directly extended sequential Gauss-Seidel-type multi-block ADMM may not be convergent even with a small step length. Thus, in this paper, we employ a more powerful symmetric Gauss-Seidel-type multi-block ADMM, that is, the sGS-ADMM to solve (D). As is shown in Chen et al. (2017), the sGS-ADMM is not only guaranteed to converge theoretically, in practice it also performs better than the possibly nonconvergent directly extended multi-block ADMM.

The Lagrangian function associated with (D) is given by

$$l(y, Z, S; X) := -\langle b, y\rangle - \log\det(Z) + Q^*(-S) + \delta_{\mathbb{S}^n_+}(Z) - n - \langle C - \mathcal{A}^* y - Z - S, X\rangle,$$
$$\forall\, (y, Z, S, X) \in \mathbb{R}^m \times \mathbb{S}^n \times \mathbb{S}^n \times \mathbb{S}^n.$$

The associated augmented Lagrangian function is

$$\mathcal{L}_\sigma(y, Z, S; X) := l(y, Z, S; X) + \frac{\sigma}{2}\|C - \mathcal{A}^* y - Z - S\|^2, \tag{14}$$

where $\sigma > 0$ is a given parameter. Based on the augmented Lagrangian function (14), we design the sGS-ADMM for solving (D). To be specific, we update Z and (y, S) alternatively

as in the commonly used 2-block ADMM, but with the key difference of applying the sGS iteration technique (Li et al., 2018) to the second block. The template for the algorithm is given as follows:

$$\begin{cases} Z^{k+1} = \arg\min \mathcal{L}_\sigma(y^k, Z, S^k; X^k), \\ \overline{y}^{k+1} = \arg\min \mathcal{L}_\sigma(y, Z^{k+1}, S^k; X^k), \\ S^{k+1} = \arg\min \mathcal{L}_\sigma(\overline{y}^{k+1}, Z^{k+1}, S; X^k), \\ y^{k+1} = \arg\min \mathcal{L}_\sigma(y, Z^{k+1}, S^{k+1}; X^k), \\ X^{k+1} = X^k - \tau\sigma(C - \mathcal{A}^* y^{k+1} - Z^{k+1} - S^{k+1}), \end{cases}$$

where $\tau \in (0, (1+\sqrt{5})/2)$ is a given step length that is typically set to be 1.618. The implementation of updating each variable can be given as follows.

Updating of Z. Given $\widehat{y}, \widehat{S}, \widehat{X}$, the unique minimizer of $\mathcal{L}_\sigma(\widehat{y}, Z, \widehat{S}; \widehat{X})$ can be obtained by

$$\overline{Z} = \underset{Z \succeq 0}{\arg\min} \left\{ \frac{\sigma}{2} \left\| Z + \frac{1}{\sigma}\widehat{M} \right\|^2 - \log\det(Z) \right\} = \frac{1}{\sigma}\phi_\sigma^+(-\widehat{M}) = \frac{1}{\sigma}(\phi_\sigma^+(\widehat{M}) - \widehat{M}),$$

where $\widehat{M} = \widehat{X} - \sigma(C - \mathcal{A}^*\widehat{y} - \widehat{S})$.

Algorithm 1 : sGS-ADMM

Input: Given $C \in \mathbb{S}^n$, $b \in \mathbb{R}^m$, linear map $\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m$, parameters $\rho, \lambda > 0$. Set the maximum iteration number as `Maxiter`. Choose $X^0 \in \mathbb{S}^n_{++}$, $S^0 \in \mathbb{S}^n$, $y^0 \in \mathbb{R}^m$, $\sigma > 0$, $\tau \in (0, (1+\sqrt{5})/2)$, and set $k = 0$.

1: Compute

$$Z^{k+1} = (\phi_\sigma^+(M^k) - M^k)/\sigma, \quad M^k = X^k - \sigma(C - \mathcal{A}^* y^k - S^k),$$

where the definition of $\phi_\sigma^+(\cdot)$ could be found in (12).

2: Compute

$$\begin{cases} \overline{y}^{k+1} = & (\mathcal{A}\mathcal{A}^*)^{-1}\big(\mathcal{A}(C - S^k - Z^{k+1} - X^k/\sigma) + b/\sigma\big), \\ S^{k+1} = & -V^k + \mathrm{Prox}_Q(V^k), \quad V^k = -(C - \mathcal{A}^*\overline{y}^{k+1} - Z^{k+1} - X^k/\sigma), \\ y^{k+1} = & (\mathcal{A}\mathcal{A}^*)^{-1}\big(\mathcal{A}(C - S^{k+1} - Z^{k+1} - X^k/\sigma) + b/\sigma\big), \end{cases}$$

where the explicit formula of $\mathrm{Prox}_Q(\cdot)$ could be seen in (10).

3: Compute

$$X^{k+1} = X^k - \tau\sigma(C - \mathcal{A}^* y^{k+1} - S^{k+1} - Z^{k+1}).$$

4: If $k =$ `Maxiter` or the stopping criterion (17) is satisfied, terminate; otherwise $k \leftarrow k+1$, go to Step 1.

Updating of y. Given $\widehat{Z}, \widehat{S}, \widehat{X}$, the unique minimizer of $\mathcal{L}_\sigma(y, \widehat{Z}, \widehat{S}; \widehat{X})$ can be obtained by solving the linear system as

$$\overline{y} = \underset{y\in\mathbb{R}^m}{\arg\min} \Big\{ -\langle b, y\rangle + \frac{\sigma}{2}\Big\| C - \mathcal{A}^* y - \widehat{Z} - \widehat{S} - \frac{1}{\sigma}\widehat{X}\Big\|^2 \Big\} = (\mathcal{A}\mathcal{A}^*)^{-1}\big(\mathcal{A}(C - \widehat{S} - \widehat{Z} - \frac{1}{\sigma}\widehat{X}) + \frac{1}{\sigma}b\big).$$

Updating of S. Given $\widehat{y}, \widehat{Z}, \widehat{X}$, the unique minimizer of $\mathcal{L}_\sigma(\widehat{y}, \widehat{Z}, S; \widehat{X})$ could be given as

$$\overline{S} = \underset{S\in\mathbb{S}^n}{\arg\min} \Big\{ Q^*(-S) + \frac{\sigma}{2}\|S + \widehat{V}\|^2 \Big\} = -\mathrm{Prox}_{Q^*}(\widehat{V}) = -\widehat{V} + \mathrm{Prox}_Q(\widehat{V}),$$

where $\widehat{V} = -(C - \mathcal{A}^*\widehat{y} - \widehat{Z} - \widehat{X}/\sigma)$.

The whole sGS-ADMM for solving (D) is summarized in Algorithm 1. The convergence result of this algorithm can be obtained from Chen et al. (2017, Theorem 5.1) as stated in the following theorem.

Theorem 9 *Let $\{(y^k, Z^k, S^k, X^k)\}$ be the sequence generated by Algorithm 1. Then the sequence $\{(y^k, Z^k, S^k)\}$ converges to an optimal solution of* (D) *and $\{X^k\}$ converges to the optimal solution X^* of* (P).

3.2 Phase II: pALM

The augmented Lagrangian method (ALM) is a widely used method for solving the convex optimization problem in the literature. It has the important property of possessing superlinear convergence guarantee.

We write the dual problem (D) in the following unconstrained form

$$\min_{y\in\mathbb{R}^m, S\in\mathbb{S}^n} \Big\{ f(y, S) = -\langle b, y\rangle - \log\det(C - \mathcal{A}^* y - S) + Q^*(-S) + \delta_{\mathbb{S}^n_+}(C - \mathcal{A}^* y - S) - n \Big\}. \quad (\mathrm{D}')$$

Denote

$$\widetilde{f}(y, S, Z, V) = -\langle b, y\rangle - \log\det(C - \mathcal{A}^* y - S - Z) + Q^*(-S + V) + \delta_{\mathbb{S}^n_+}(C - \mathcal{A}^* y - S - Z) - n.$$

Then by Rockafellar and Wets (2009, Example 11.46), the Lagrangian function associated with (D′) is

$$\begin{aligned}\widetilde{l}(y, S; X, U) &= \inf_{Z\in\mathbb{S}^n, V\in\mathbb{S}^n} \Big\{ \widetilde{f}(y, S, Z, V) - \langle Z, X\rangle - \langle U, V\rangle \Big\} \\ &= -\langle b, y\rangle - \langle C - \mathcal{A}^* y - S, X\rangle + \log\det X - \delta_{\mathbb{S}^n_+}(X) - \langle U, S\rangle - Q(U).\end{aligned}$$

By Rockafellar and Wets (2009, Example 11.57), the corresponding augmented Lagrangian function is

$$\begin{aligned}\widetilde{L}_\sigma(y, S; X, U) &= \sup_{\widetilde{X}\in\mathbb{S}^n, \widetilde{U}\in\mathbb{S}^n} \Big\{ \widetilde{l}(y, S; \widetilde{X}, \widetilde{U}) - \frac{1}{2\sigma}\|X - \widetilde{X}\|^2 - \frac{1}{2\sigma}\|U - \widetilde{U}\|^2 \Big\} \\ &= -\langle b, y\rangle - \frac{1}{\sigma}\mathrm{E}_{\sigma r}(M(y, S)) + \frac{1}{2\sigma}\|M(y, S)\|^2 - \frac{1}{2\sigma}\|X\|^2 - \frac{1}{\sigma}\mathrm{E}_{\sigma Q}(U - \sigma S) + \frac{1}{2\sigma}\|U - \sigma S\|^2 - \frac{1}{2\sigma}\|U\|^2,\end{aligned}$$

where $M(y, S) = X - \sigma(C - \mathcal{A}^* y - S)$, $\sigma > 0$ is a given parameter.

Based on the above notations, we describe the pALM for solving (D′) as follows.

Algorithm 2 : pALM

Input: Given $C \in \mathbb{S}^n$, $b \in \mathbb{R}^m$, linear map $\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m$, parameters $\rho, \lambda > 0$. Denote the approximate solution obtained from Phase I as $(X^0, y^0, S^0) \in \mathbb{S}^n_{++} \times \mathbb{R}^m \times \mathbb{S}^n$. Define $U^0 = X^0$. Let $\tau > 0$, $0 < \sigma_0 < \sigma_\infty \le \infty$, and set $k = 0$.

1: Compute

$$(y^{k+1}, S^{k+1}) \approx \underset{y \in \mathbb{R}^m, S \in \mathbb{S}^n}{\arg\min} \Big\{ \Psi_k(y, S) = \widetilde{L}_\sigma(y, S; X^k, U^k) + \frac{\tau}{2\sigma_k}(\|y - y^k\|^2 + \|S - S^k\|^2) \Big\}. \quad (15)$$

2: Compute

$$X^{k+1} = \mathrm{Prox}_{\sigma_k r}(X^k - \sigma(C - \mathcal{A}^* y^{k+1} - S^{k+1})),$$
$$U^{k+1} = \mathrm{Prox}_{\sigma_k Q}(U^k - \sigma S^{k+1}),$$

where the formulae of $\mathrm{Prox}_{\sigma_k r}(\cdot)$ and $\mathrm{Prox}_{\sigma_k Q}(\cdot)$ could be found in (13) and (10), respectively.

3: If the stopping criterion (17) is satisfied, terminate; otherwise update $\sigma_{k+1} \uparrow \sigma_\infty$, $k \leftarrow k + 1$, go to Step 1.

3.2.1 Convergence result of the pALM

The global convergence and global linear-rate convergence of the pALM are provided in this subsection. To establish the convergence result, we define the maximal monotone operator

$$\mathcal{T}_{\widetilde{l}}(y, S, X, U) := \Big\{ (y', S', X', U') \mid (y', S', -X', -U') \in \partial \widetilde{l}(y, S; X, U) \Big\},$$

and its inverse operator

$$\mathcal{T}_{\widetilde{l}}^{-1}(y', S', X', U') := \arg\min_{y,S} \max_{X,U} \Big\{ \widetilde{l}(y, S; X, U) - \langle y', y \rangle - \langle S', S \rangle + \langle X', X \rangle + \langle U', U \rangle \Big\}.$$

As we note in the pALM, we need to specify the stopping criterion of computing the approximate solution (y^{k+1}, S^{k+1}) in (15). Denote the operator

$$\Lambda = \mathrm{Diag}(\tau I_m, \tau \mathcal{I}_n, \mathcal{I}_n, \mathcal{I}_n),$$

where $\mathcal{I}_n$ is the identity operator over $\mathbb{S}^n$. We use the following stopping criteria for solving (15):

$$\|\nabla \Psi_k(y^{k+1}, S^{k+1})\| \le \frac{\min\{\sqrt{\tau}, 1\}}{\sigma_k} \varepsilon_k, \quad (A)$$

$$\|\nabla \Psi_k(y^{k+1}, S^{k+1})\| \le \frac{\min\{\sqrt{\tau}, 1\}}{\sigma_k} \delta_k \|(y^{k+1}, S^{k+1}, X^{k+1}, U^{k+1}) - (y^k, S^k, X^k, U^k)\|_\Lambda, \quad (B)$$

where $\{\varepsilon_k\}$ and $\{\delta_k\}$ are summable nonnegative sequences satisfying $\delta_k < 1$ for all k.

Based on the above preparation, we could present the convergence result of the pALM in the following theorem, which is an application of Li et al. (2020, Theorem 1 and Theorem 2).

Theorem 10 *Let $\{(y^k, S^k, X^k, U^k)\}$ be the sequence generated by Algorithm 2 with the stopping criterion* (A).

(a) Then $\{(y^k, S^k, X^k, U^k)\}$ is bounded, $\{(y^k, S^k)\}$ converges to an optimal solution of (D′)*, and both $\{X^k\}$ and $\{U^k\}$ converge to the optimal solution X^* of* (P).

(b) Assume that for $\zeta := \mathrm{dist}_\Lambda((y^0, S^0, X^0, U^0), \mathcal{T}_{\widetilde{l}}^{-1}(0)) + \sum_{k=0}^{\infty} \varepsilon_k$, there exists $\kappa > 0$ such that

$$\mathrm{dist}_\Lambda((y, S, X, U), \mathcal{T}_{\widetilde{l}}^{-1}(0)) \le \kappa \mathrm{dist}(0, \mathcal{T}_{\widetilde{l}}(y, S, X, U)),$$

for all (y, S, X, U) satisfying $\mathrm{dist}_\Lambda((y, S, X, U), \mathcal{T}_{\widetilde{l}}^{-1}(0)) \le \zeta$. Suppose the stopping criteria (B) *is also satisfied. Then for $k \ge 0$, it holds that*

$$\mathrm{dist}_\Lambda((y^{k+1}, S^{k+1}, X^{k+1}, U^{k+1}), \mathcal{T}_{\widetilde{l}}^{-1}(0)) \le \mu_k \mathrm{dist}_\Lambda((y^k, S^k, X^k, U^k), \mathcal{T}_{\widetilde{l}}^{-1}(0)),$$

where

$$\mu_k = \frac{\delta_k + (1+\delta_k)\kappa \max\{\tau, 1\}/\sqrt{\sigma_k^2 + \kappa^2 \max\{\tau^2, 1\}}}{1 - \delta_k} \to \mu_\infty := \frac{\kappa \max\{\tau, 1\}}{\sqrt{\sigma_\infty^2 + \kappa^2 \max\{\tau^2, 1\}}}.$$

3.2.2 An SSN method for solving the pALM subproblems

As one can see, the main task in the pALM is to solve the subproblem (15) in an efficient way. Note that given $(\widetilde{y}, \widetilde{S}, \widetilde{X}, \widetilde{U})$, the subproblem (15) takes the form of

$$\min_{y \in \mathbb{R}^m, S \in \mathbb{S}^n} \Big\{ \Psi(y, S) := \widetilde{L}_\sigma(y, S; \widetilde{X}, \widetilde{U}) + \frac{\tau}{2\sigma}(\|y - \widetilde{y}\|^2 + \|S - \widetilde{S}\|^2) \Big\}.$$

Since $\Psi(\cdot, \cdot)$ is a strongly convex, continuously differentiable function on $\mathbb{R}^m \times \mathbb{S}^n$, the above minimization problem has a unique optimal solution, denoted as $(\overline{y}, \overline{S})$, which can be computed by solving the nonsmooth optimality condition:

$$\nabla \Psi(y, S) = \begin{pmatrix} -b + \mathcal{A}\mathrm{Prox}_{\sigma r}(\widetilde{M}(y, S)) + \frac{\tau}{\sigma}(y - \widetilde{y}) \\ \mathrm{Prox}_{\sigma r}(\widetilde{M}(y, S)) - \mathrm{Prox}_{\sigma Q}(\widetilde{U} - \sigma S) + \frac{\tau}{\sigma}(S - \widetilde{S}) \end{pmatrix} = 0, \tag{16}$$

where $\widetilde{M}(y, S) = \widetilde{X} - \sigma(C - \mathcal{A}^* y - S)$.

For any $(y, S) \in \mathbb{R}^m \times \mathbb{S}^n$, define the generalized Hessian $\hat{\partial}^2 \Psi(y, S) : \mathbb{R}^m \times \mathbb{S}^n \rightrightarrows \mathbb{R}^m \times \mathbb{S}^n$ as follows: $\mathcal{Q} \in \hat{\partial}^2 \Psi(y, S)$ if and only if there exists $\mathcal{P} \in \hat{\partial} \mathrm{Prox}_{\sigma Q}(\widetilde{U} - \sigma S)$ such that

$$\mathcal{Q} \begin{pmatrix} d_y \\ d_S \end{pmatrix} = \sigma \begin{pmatrix} \mathcal{A} \\ \mathcal{I}_n \end{pmatrix} (\phi_\sigma^+(\widetilde{M}(y, S)))'(\mathcal{A}^* d_y + d_S) + \sigma \begin{pmatrix} 0 \\ \mathcal{P} d_S \end{pmatrix} + \frac{\tau}{\sigma} \begin{pmatrix} d_y \\ d_S \end{pmatrix}, \ \forall (d_y, d_S) \in \mathbb{R}^m \times \mathbb{S}^n.$$

We can treat $\hat{\partial}^2 \Psi(y, S)$ as a surrogate of the Clarke generalized Jacobian of $\nabla \Psi(\cdot, \cdot)$ at (y, S), according to the analysis of the regularization term $Q(\cdot)$ and the function $r(\cdot)$ in Section 2. In addition, we have that $\nabla \Psi(\cdot, \cdot)$ is strongly semismooth with respect to $\hat{\partial}^2 \Psi(\cdot, \cdot)$. Thus we could apply the SSN method to solve (16), which has the following template.

Algorithm 3 : SSN

Input: Given $(\widetilde{y}, \widetilde{S}, \widetilde{X}, \widetilde{U}) \in \mathbb{R}^m \times \mathbb{S}^n \times \mathbb{S}^n \times \mathbb{S}^n$, choose $y^0 \in \mathbb{R}^m$, $S^0 \in \mathbb{S}^n$, $\beta \in (0,1]$, $\eta \in (0,1)$, and $\zeta \in (0, \frac{1}{2}), \delta \in (0,1)$, and set $j = 0$.

1: Choose $\mathcal{H}_j \in \hat{\partial}\mathrm{Prox}_{\sigma Q}(\widetilde{U} - \sigma S^j)$, use the conjugate gradient method (CG) to solve the linear system

$$\sigma \begin{pmatrix} \mathcal{A} \\ \mathcal{I}_n \end{pmatrix} (\phi_\sigma^+(\widetilde{M}(y^j, S^j)))'(\mathcal{A}^* d_y^j + d_S^j) + \sigma \begin{pmatrix} 0 \\ \mathcal{H}_j d_S^j \end{pmatrix} + \frac{\tau}{\sigma} \begin{pmatrix} d_y^j \\ d_S^j \end{pmatrix} = -\nabla \Psi(y^j, S^j)$$

to obtain d_y^j and d_S^j such that the residual is no larger than $\min\{\eta, \|\nabla \Psi(y^j, S^j)\|^{1+\beta}\}$.

2: Set $\alpha_j = \delta^{m_j}$, where m_j is the first nonnegative integer m for which

$$\Psi(y^j + \delta^m d_y^j, S^j + \delta^m d_S^j) \le \Psi(y^j, S^j) + \zeta \delta^m \left\langle \nabla \Psi(y^j, S^j), \begin{pmatrix} d_y^j \\ d_S^j \end{pmatrix} \right\rangle.$$

3: Set $y^{j+1} = y^j + \alpha_j d_y^j$, $S^{j+1} = S^j + \alpha_j d_S^j$, $j \leftarrow j + 1$, go to Step 1.

Since the operator $\hat{\partial}^2 \Psi(\cdot, \cdot)$ is positive definite, we can obtain the following superlinear or even quadratic convergence result of the SSN method from Zhao et al. (2010).

Theorem 11 *Let $\{(y^j, S^j)\}$ be the sequence generated by Algorithm 3, then $\{(y^j, S^j)\}$ converges to the unique minimizer $(\overline{y}, \overline{S})$ of $\Psi(\cdot, \cdot)$ and*

$$\|(y^{j+1}, S^{j+1}) - (\overline{y}, \overline{S})\| = O(\|(y^j, S^j) - (\overline{y}, \overline{S})\|^{1+\beta}),$$

where $\beta \in (0,1]$ is from the algorithm.

4. Numerical Experiments

In this section, we present some numerical experiments on both synthetic and real data to demonstrate the performance of our proposed model and the efficiency of our two-phase algorithm. The experiments are mainly in three aspects:

- comparing our model with the sparse Gaussian graphical model (Yuan and Lin, 2007; Banerjee et al., 2008) and GRAB (Hosseini and Lee, 2016) on the performance of estimating concentration matrices, in the sense of Fscore and ROC curve.
- comparing our proposed two-phase algorithm for solving the model (P) with the standalone sGS-ADMM proposed in Section 3.1 and the alternating linearization method proposed in Scheinberg et al. (2010), in the sense of computation time;
- giving an illustration on real data to show how our proposed model could learn similarities among items and detect some meaningful clusters.

All experiments are implemented in Matlab R2022b on a windows workstation (16-core, Intel Xeon Gold 6244 @ 3.60GHz, 128 G RAM).

4.1 Stopping Criteria of the Two-phase Algorithm

In this paper, we use the relative KKT residual to measure the quality of the obtained solution. That is, we stop the algorithm when

$$\max\{R_P, R_D, R_C\} < \texttt{Tol}, \tag{17}$$

with $\texttt{Tol} = 10^{-6}$ as the default, where

$$R_P = \frac{\|\mathcal{A}X - b\|}{1 + \|b\|},\ R_D = \frac{\|C - \mathcal{A}^* y - S - Z\|}{1 + \|C\|},\ R_C = \max\Big\{\frac{\|XZ - I_n\|}{1 + \|X\| + \|Z\|}, \frac{\|X - \mathrm{Prox}_Q(X - S)\|}{1 + \|X\| + \|S\|}\Big\}.$$

Furthermore, we also provide the duality gap for reference, which is

$$R_G := \frac{|\texttt{pobj} - \texttt{dobj}|}{1 + |\texttt{pobj}| + |\texttt{dobj}|},$$

where `pobj` and `dobj` are the primal and dual objective function values given by

$$\texttt{pobj} = \langle C, X\rangle - \log\det(X) + Q(X), \quad \texttt{dobj} = \langle b, y\rangle + \log\det(Z) + n.$$

As a side note, in Phase II, the variable Z could be constructed according to the derivation of the Lagrangian function as $(\phi_\sigma^+(M) - M)/\sigma$, where $M = X - \sigma(C - \mathcal{A}^* y - S)$.

In our two-phase algorithm, we fix the iteration number of the sGS-ADMM in Phase I to be 200 in consideration of the trade-off between the computation time and the effect of warm start, and then run Phase II until the stopping criterion (17) is satisfied.

4.2 Experiments on Synthetic Data

In this subsection, we conduct experiments on edge coloring models, autoregressive models and modular graph models, to demonstrate the performance of our proposed estimator and the two-phase algorithm. Specifically, we will test the performance of different forms of our estimator: the unconstrained one (2), the one with constraints from a given zero pattern (3), and the one with general linear constraints (P).

4.2.1 Data construction

Given a true concentration matrix $\Sigma^{-1} \in \mathbb{S}^n$, we first generate p samples $\{z^{(i)}\}_{i=1}^p$ with $z^{(i)} \sim \mathcal{N}(0, \Sigma)$, then construct the sample covariance matrix C as

$$C = \frac{1}{p}\sum_{i=1}^p (z^{(i)} - \bar{z})(z^{(i)} - \bar{z})^T, \quad \bar{z} = \frac{1}{p}\sum_{i=1}^p z^{(i)}.$$

We consider three kinds of concentration matrices with sparsity and clustering structure as follows.

(1) (Edge coloring models) We generate a sparse edge coloring graph $\Sigma^{-1} \in \mathbb{S}^n$ as in Lauritzen and Højsgaard (2008); Højsgaard and Lauritzen (2008a) with n_G clusters of the coordinates, which is taken as a block matrix, where the probability of having a nonzero (i, j)-block is 0.3 for $1 \le i \ne j \le n_G$, and 1 for $i = j = 1, \cdots, n_G$. Within each

nonzero block, the entries are drawn i.i.d. from the Gaussian distribution $\mathcal{N}(\mu, 1)$, where μ is uniformly chosen from $[-1, 1]$. To ensure the positive definiteness of Σ^{-1}, we apply the same procedure in d'Aspremont et al. (2008); Wang et al. (2010) to compute

$$\Sigma^{-1} = \Sigma^{-1} + \max\{-1.2\sigma_{\min}(\Sigma^{-1})), 0.001\}I_n.$$

(2) (Autoregressive models) Consider an AR(k) process $Y_t = \sum_{j=1}^{k} \phi_j Y_{t-j} + \epsilon_t$, where ϵ_t's are i.i.d. such that $\mathbb{E}[\epsilon_t] = 0$ and $\mathrm{Var}[\epsilon_t] = 1$. We generate each element of $\phi \in \mathbb{R}^k$ randomly from a standard Gaussian distribution, then scale ϕ such that $\|\phi\|_2 = 0.9 < 1$ to ensure the stationarity of the process. Consider $\epsilon_0 = Y_0$, then we have $\epsilon = LY$, where $\epsilon = [\epsilon_0, \cdots, \epsilon_{n-1}]^T$, $Y = [Y_0, \cdots, Y_{n-1}]^T$ and L is a lower triangular matrix with its diagonal elements being 1 and jth off diagonal elements being $-\phi_j$ for $j = 1, \cdots, k$. Denote the covariance matrix of Y as Σ. Then the fact that $\mathrm{Var}(\epsilon) = L\Sigma L^T = I_n$ implies that $\Sigma^{-1} = L^T L$.

(3) (Modular graph models) We generate a modular graph via the procedure in Egilmez et al. (2017) with n vertices and n_G modules, where the vertex attachment probabilities across and within modules are 0.01 and 0.3. The edge weights are randomly selected based on a uniform distribution from $[0.1, 3]$. Take Σ^{-1} as the Laplacian matrix of the graph.

4.2.2 Experiments on edge coloring problems

We compare three methods for estimating edge coloring graphs: our proposed unconstrained estimator (2) with $\lambda = \frac{\rho}{n^2}$, the sparse Gaussian graphical model (1) and GRAB (Hosseini and Lee, 2016) (with the true number of clusters as a prior). Note that here we test the performance of our proposed unconstrained estimator with only one tuning parameter ρ by fixing the ratio of ρ and λ to fairly compare different models. From the experimental results, we will see that this estimator with a fixed ratio on the two parameters already provide better performance than the other two estimators. In practice, one can further improve the performance of our estimator by tuning ρ and λ simultaneously.

Note that in each method, we always have one tuning parameter ρ, which will be selected by grid search. Specifically, for our estimator, we select ρ in the range of 5×10^{-5} to 5×10^{-3} with 20 equally divided grid points; for the sparse Gaussian graphical model, we select ρ in the range of 5×10^{-5} to 3×10^{-3} with 20 equally divided grid points; and for GRAB, we select ρ in the range of 8×10^{-5} to 2×10^{-3} with 20 equally divided grid points.

Consider the edge coloring graphs with $n = 100$, $n_G = 10$. All simulation experiments involve 100 independent trials. In each trial, we draw $p \in \{0.5n, n, 5n, 10n, 20n, 50n\}$ independent samples from the underlying edge coloring model, which are used to compute the sample covariance matrix and the concentration matrix estimators. Here we use

$$\text{Fscore} := \frac{2\text{tp}}{2\text{tp} + \text{fp} + \text{fn}},$$

to measure the estimation accuracy, where tp, fp, and fn denote the number of true positive, false positive, and false negative edges between the truth Σ^{-1} and the estimator $\widehat{X}$, respectively.

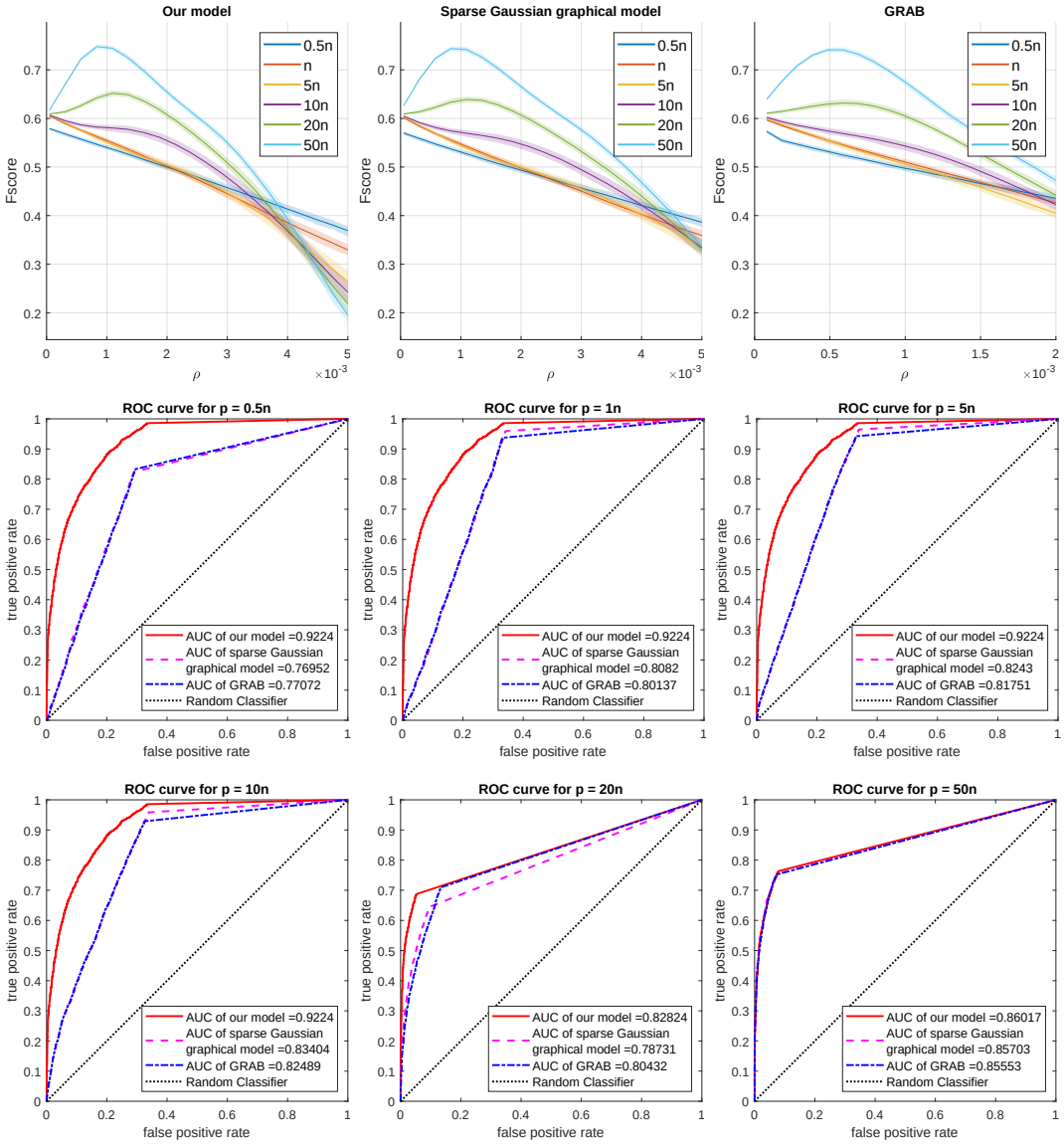


Figure 1: Experimental results on edge coloring models. First three figures shows Fscore v.s. the parameter ρ of different models and different sample sizes over 100 independent trials. Last six figures show the ROC curve of three estimators for each sample size in the first trial.

Figure 1 shows the results on edge coloring problems, wherein the first three figures show Fscore versus the parameter ρ of different models and different sample sizes. Lines in the figures represent averages, and shaded areas capture the tubes between the empirical 10% and 90% quantiles across all 100 trials. We can see that three models show similar best Fscore, while our model performs slightly better than the other two. In order to better compare the three estimators, we further use the ROC curve to investigate the accuracy of each estimator, which is selected as the one that maximizes the Fscore over all ρ for that method. The results for different samples sizes in the first trial are shown in the last six

figures of Figure 1. We find that the AUC (Area Under the ROC Curve) of our estimator is always much larger than that of the other two estimators, especially in the low-sample cases. This further demonstrates the superior performance of our proposed estimator.

4.2.3 Experiments on autogressive models

Consider the problem of estimating the concentration matrices of AR(10) models when $n = 100$ and $p \in \{0.5n, n, 5n, 10n\}$. We compare the performance of four estimators: our unconstrained estimator (2) with $\lambda = \frac{\rho}{n^2}$; our estimator (3) with $\lambda = \frac{\rho}{n^2}$ and the prior knowledge of AR(k) structure, where $k = 10$ or $k = 20$; and the sparse Gaussian graphical model (1). Specifically, our estimator (3) with $\lambda = \frac{\rho}{n^2}$ and the prior knowledge of AR(k) structure takes the form as:

$$\min_{X \in \mathbb{S}^n_{++}} \Big\{ \langle C, X \rangle - \log\det(X) + \rho \sum_{i<j} |X_{ij}| + \frac{\rho}{n^2} \sum_{i<j} \sum_{s<t} |X_{ij} - X_{st}| \;\Big|\; X_{ij} = 0 \text{ if } |i-j| > k \Big\}.$$

Each estimator has one parameter ρ, which we select from 4×10^{-5} to 0.04 (resp. 0.01 to 0.2) with 20 equally divided grid points for the first three estimators (resp. the last one).

Figure 2 shows the experimental results of estimating the concentration matrices of AR(10) models, wherein the first four figures show Fscore versus the parameter ρ of different estimators and different sample sizes. It reveals that the estimator obtained by our model with constraints ($k = 10$) dominates the other three estimators, as it imposes the exact prior knowledge. In addition, the superior performance of our model with constraints ($k = 20$) provides the numerical evidence that, in practice, it would be a good choice to randomly guess a relatively large and reasonable value of k when imposing the AR(k) structure without the prior knowledge of k. As for the remaining two estimators, it can be seen that our model without constraints provides slightly better Fscore than the sparse Gaussian graphical model. We further use the ROC curve to investigate the accuracy of each estimator, which is selected as the one that maximizes the Fscore over all ρ for different methods. The results in the first trial are shown in the last four plots of Figure 2. We can see that the two estimators with prior knowledge both give excellent performance and have AUC greater than 94%. Moreover, the performance of our model without constraints is also much better than that of the sparse Gaussian graphical model, which demonstrates that the ℓ_1 penalty imposed on the pairwise differences indeed helps in estimating the concentration matrix for autogressive models.

4.2.4 Experiments on modular graph recovery

In this part, we consider the problem of modular graph recovery with $n = 200$, $n_G = 10$ and the sample size $p \in \{0.5n, n, 5n, 10n\}$. We compare three estimators: our unconstrained estimator (2) with $\lambda = \frac{\rho}{n^2}$, the sparse Gaussian graphical model (1), and our estimator with $\lambda = \frac{\rho}{n^2}$ and linear constraints:

$$\begin{aligned} \min_{X \in \mathbb{S}^n_+} \; & \Big\{ \langle C, X \rangle - \log\det(X) + \rho \sum_{i<j} |X_{ij}| + \frac{\rho}{n^2} \sum_{i<j} \sum_{s<t} |X_{ij} - X_{st}| \Big\} \qquad (18) \\ \text{s.t.} \quad & (e_i - e_j)^T X (e_i - e_j) \geq l, \quad (i,j) \in \mathcal{D}, \end{aligned}$$

where $l = 0.7 \max_{1\le i,j\le n}\{(\Sigma^{-1})_{ii} + (\Sigma^{-1})_{jj} - 2(\Sigma^{-1})_{ij}\}$, the dissimilarity constraint set $\mathcal{D}$ is generated by randomly choosing a subset of $\widehat{\mathcal{D}} := \{(i,j) \mid (\Sigma^{-1})_{ii} + (\Sigma^{-1})_{jj} - 2(\Sigma^{-1})_{ij} \ge l,\ i,j = 1,\cdots,n\}$ with $|\mathcal{D}| = \lfloor 0.5|\widehat{\mathcal{D}}| \rfloor$.

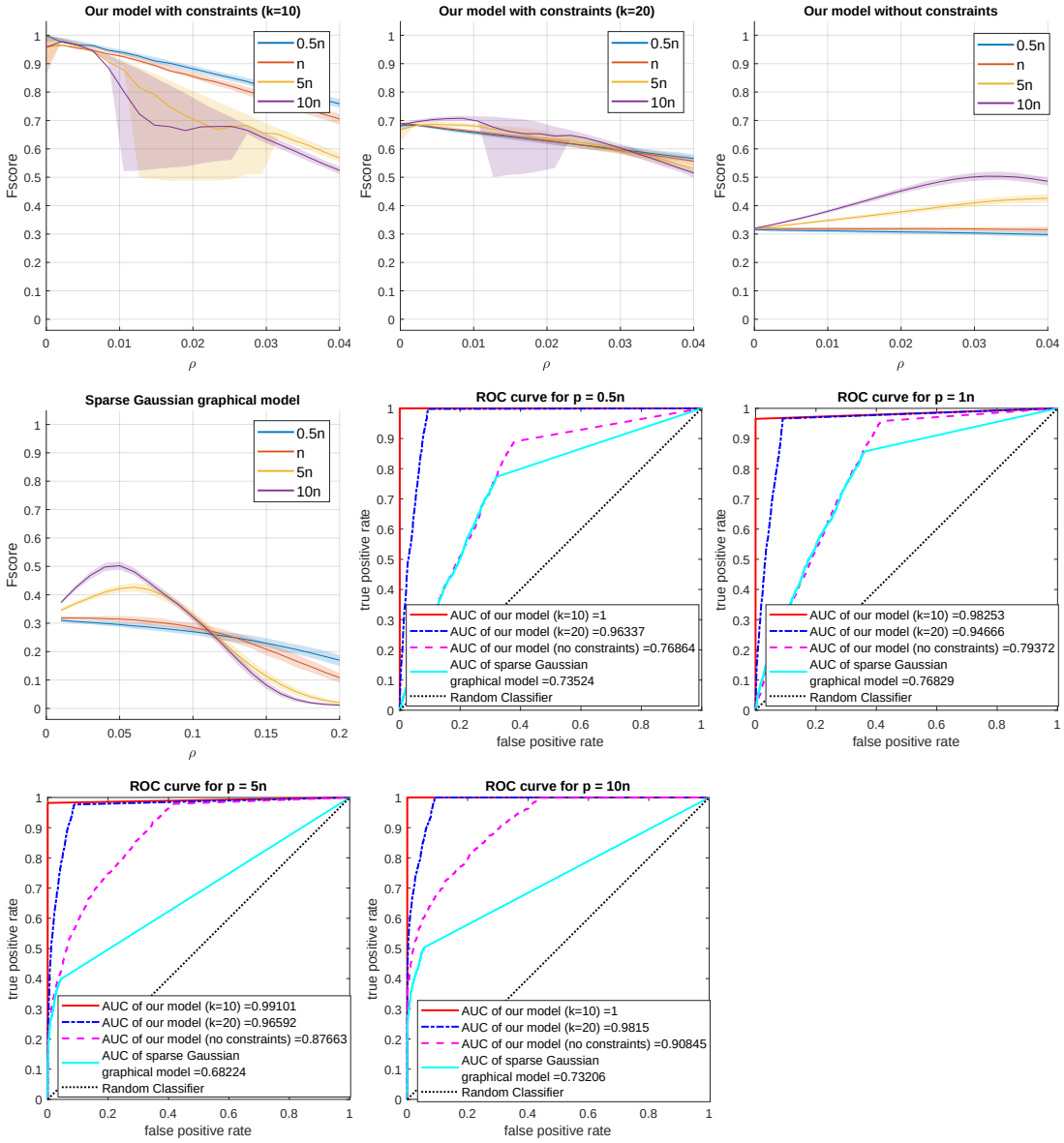


Figure 2: Estimation results for AR(10) model. First four figures shows Fscore v.s. the parameter ρ. Last four figures show the ROC curve of four estimators for each sample size in the first trial.

In each tested estimator, we have one tuning parameter ρ, where we select ρ in the range of 0.005 to 0.05 with 20 equally divided grid points for our estimators and select ρ in the range of 0.01 to 0.1 with 20 equally divided grid points for the sparse Gaussian graphical model. Note that the problem (18) is not expressed in the standard form given in (P), but

it can easily be expressed as such by introducing an additional slack variable. To be precise, the standard form reformulation of (18) is given as follows:

$$\min_{X\in\mathbb{S}^n, x\in\mathbb{R}^m} \Big\{ \langle C, X\rangle - \log\det(X) + \rho\sum_{i<j}|X_{ij}| + \frac{\rho}{n^2}\sum_{i<j}\sum_{s<t}|X_{ij}-X_{st}| \;\Big|\; \mathcal{A}X - x = b, X\succeq 0, x\geq 0\Big\},$$

where $\mathcal{A}X = [(e_i-e_j)^T X(e_i-e_j)]_{(i,j)\in\mathcal{D}} \in \mathbb{R}^{|\mathcal{D}|}$ and $b = l1_{|\mathcal{D}|}$. Then our proposed algorithm in Section 3 can be extended to solve the above standard form reformulation.

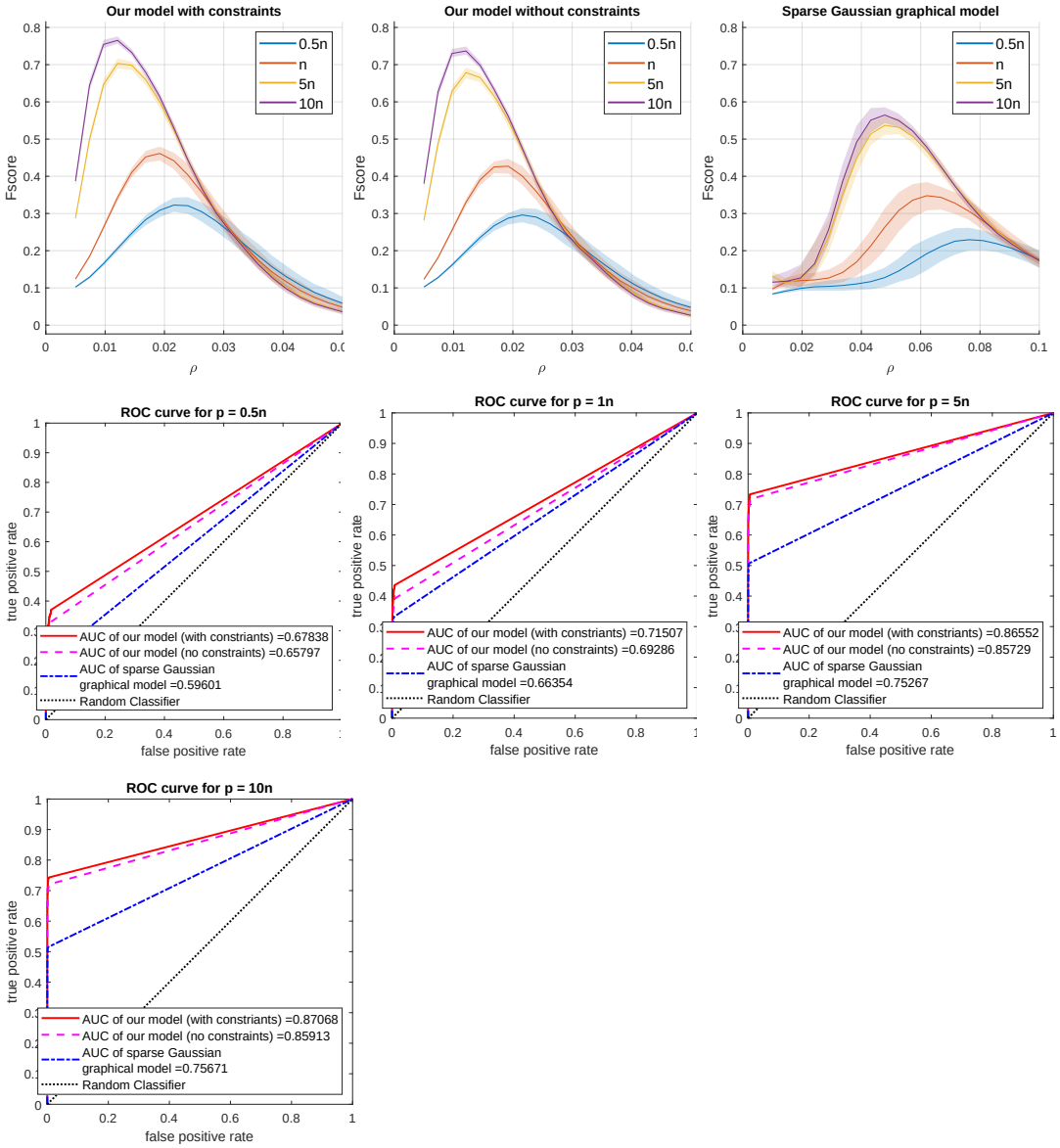
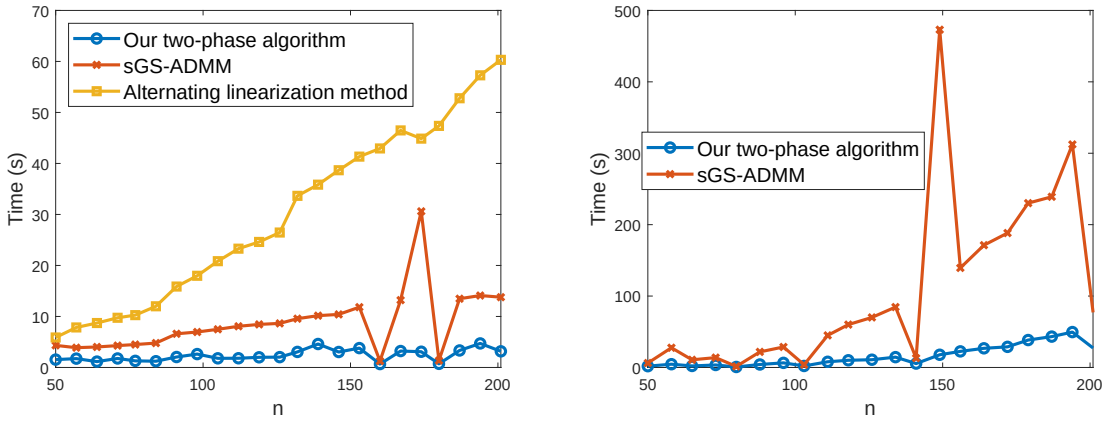


Figure 3: Numerical results on modular graph recovery. First three figures shows Fscore v.s. the parameter ρ of different models and different sample sizes. Last four figures show the ROC curve of three estimators for each sample size in the first trial.

Figure 3 shows the experimental results of three estimators for modular graph recovery, where the first three plots show the Fscore versus the parameter ρ of different estimators and different sample sizes. We can see that among the three estimators, our model with contraints performs better than the one without constraints, and both of them perform much better than the sparse Gaussian graphical model. The ROC curve of each estimator in the first trial is shown in the last four plots of Figure 3. We find our estimators have higher AUC than the sparse Gaussian graphical model in all cases, which demonstrates the superior performance of our proposed estimators in modular graph recovery. Moreover, our model with constraints also performs better than the one without constraints in the sense of AUC.



(a) Comparison for the unconstrained case

(b) Comparison for the constrained case

Figure 4: Comparison among algorithms for solving the model (P) on modular graph recovery.

4.2.5 Comparison with other algorithms

Up to our knowledge, there is no other existing algorithm in the literature which is suitable to solve (P) for large n. It is widely accepted that the interior-point methods (IPMs) with direct solvers are generally efficient and robust for small- and medium-sized log-determinant problems as can be seen in Toh (1999); Tütüncü et al. (2003). However, the complicated structure of our model (P) makes it very difficult to apply the IPMs directly. In order to apply the IPMs, we need to introduce additional constraints and variables to reformulate the problem (P) as follows:

$$\min_{X\in\mathbb{S}^n, y^{\pm}\in\mathbb{R}^{\bar{n}}, z^{\pm}\in\mathbb{R}^{\hat{n}}} \Big\{\langle C, X\rangle - \log\det X + \rho\sum_{i=1}^{\bar{n}}(y_i^+ + y_i^-) + \lambda\sum_{i=1}^{\hat{n}}(z_i^+ + z_i^-)\Big\}$$
$$\text{s.t.}\quad \mathcal{A}X = b,\quad \mathcal{B}X - y^+ + y^- = 0,\quad Dy^+ - Dy^- - z^+ + z^- = 0,$$
$$X \succeq 0,\quad y^+, y^-, z^+, z^- \geq 0,$$

where $\hat{n} = \bar{n}(\bar{n}-1)/2$, $\mathcal{B}: \mathbb{S}^n \to \mathbb{R}^{\bar{n}}$ is the linear map defined in Section 2.3, $D \in \mathbb{R}^{\hat{n}\times\bar{n}}$ is defined as $Dy := \mathcal{B}(ye^T - ey^T)$. Even for the small-sized problem when $n = 50$, the above problem contains more than $800,000$ constraints. The penalty on the pairwise differences makes it impossible to apply the IPMs to solve the model (P) even for small-sized problems.

Here, we compare our two-phase algorithm with the stand-alone sGS-ADMM proposed in Section 3.1, and the alternating linearization method in Scheinberg et al. (2010). Note that the authors in Scheinberg et al. (2010) proposed the alternating linearization method to solve the sparse Gaussian graphical models, which could be modified to solve the unconstrained form (2) of our model by changing the soft-thresholding operator to the proximal mapping of $Q(\cdot)$ provided in (10). Figure 4 shows the comparison among three algorithms for solving the model (P) with and without constraints on modular graph recovery problems, where the alternating linearization method is not used in the constrained case as it is not applicable. We fix $\lfloor n/n_G \rfloor = 20$, $\rho = 0.1$ and $\lambda = \rho/n^2$. Note that both the two-phase algorithm and the stand-alone sGS-ADMM perform better than the alternating linearization method, especially for the relatively large n. In addition, we can see that the pALM in Phase II indeed accelerates the computation.

(n, n_G, m)	ρ	$\max\{R_P, R_D, R_C\}$		R_G		Iteration		Time	
		T	S	T	S	T	S	T	S
(500,10,11981)	0.1	1.68e-7	9.99e-7	4.03e-9	8.75e-7	11(195)	5451	00:02:42	00:38:05
(500,10,11981)	0.05	5.84e-7	1.00e-6	8.47e-9	5.22e-7	17(305)	17732	00:04:42	02:04:02
(1000,20,47971)	0.1	4.10e-7	**8.12e-4**	4.05e-8	**1.32e-3**	19(351)	**3751**	00:36:47	**05:00:01**
(1000,20,47971)	0.05	9.50e-7	**1.30e-3**	5.07e-8	**7.89e-4**	20(363)	**3391**	00:39:13	**05:00:00**
(2000,50,38006)	0.1	9.24e-7	**1.07e-5**	5.96e-8	**3.23e-5**	16(294)	**4044**	00:45:36	**05:00:02**
(3000,50,43498)	0.1	4.63e-7	**1.28e-5**	9.18e-7	**4.09e-5**	5(78)	**2485**	00:34:22	**05:00:01**
(4000,50,62395)	0.1	4.17e-7	**1.70e-5**	1.44e-7	**5.63e-5**	7(116)	**1235**	00:57:59	**05:00:04**

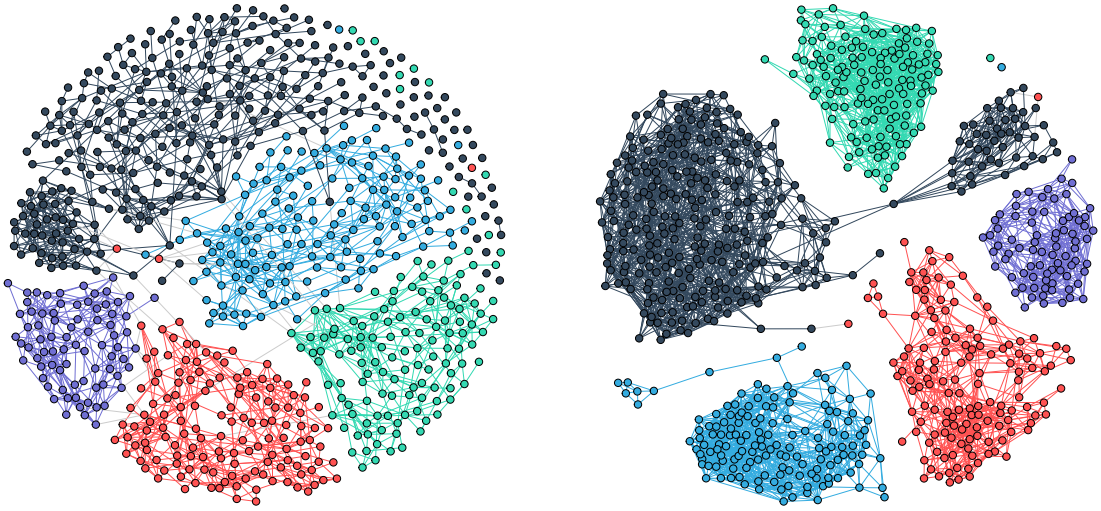
Table 1: Performance of the algorithms for autogressive models. '11(195)' means 'pALM iterations (total inner SSN iterations)' in Phase II of our two-phase algorithm. A value in bold means that the algorithm fails to solve the instance to the required accuracy.

(n, n_G, m)	ρ	$\max\{R_P, R_D, R_C\}$		R_G		Iteration		Time	
		T	S	T	S	T	S	T	S
(64,4,1008)	0.1	4.64e-7	1.00e-6	1.32e-7	1.73e-7	3(19)	2341	00:00:02	00:00:18
(64,4,1008)	0.05	7.24e-7	9.96e-7	5.44e-8	6.17e-8	5(22)	2394	00:00:02	00:00:21
(400,40,7980)	0.1	1.86e-7	1.00e-6	1.05e-8	3.17e-7	4(59)	3974	00:01:04	00:14:59
(400,40,7980)	0.05	6.35e-7	1.00e-6	2.96e-8	2.25e-7	13(233)	6073	00:02:21	00:24:53
(800,80,31960)	0.1	1.61e-7	1.00e-6	1.04e-9	5.31e-8	4(48)	2612	00:08:11	01:29:39
(800,80,31960)	0.05	6.46e-7	9.99e-7	2.25e-9	1.17e-7	16(293)	2945	00:15:56	02:08:03
(1000,100,49950)	0.1	1.08e-7	9.99e-7	1.08e-7	3.36e-8	2(12)	1475	00:13:05	02:02:43

Table 2: Performance of the algorithms for modular graph recovery.

Next we show more experimental results on the cases with larger n to demonstrate the efficiency and robustness of our proposed two-phase algorithm, see Table 1 and Table 2. In Table 1, we compare our two-phase algorithm (T) and the stand-alone sGS-ADMM (S)

for estimating large-scale concentration matrices for autogressive models. The alternating linearization method is not performed here since in Figure 4(a), we can see that it performs not as well as the other two algorithms for relatively large n, and it is not applicable for constrained cases. We set the maximum computation time of each algorithm as 5 hours. We can see that our two-phase algorithm outperforms the stand-alone sGS-ADMM by a large margin. For example, we are able to compute a highly accurate solution for a large instance with matrix dimension $n = 1000$ and 47971 linear constraints in about half an hour, while the sGS-ADMM takes 5 hours to get an approximate solution with lower accuracy. Table 2 shows the performance of two algorithms for solving large-scale modular graph recovery problems. It can also be seen that our two-phase algorithm gives much better performance than the stand-alone sGS-ADMM.



(a) Sparse Gaussian graphical model (b) Our proposed model

Figure 5: Visualization of the estimated result for the Cancer genome data set.

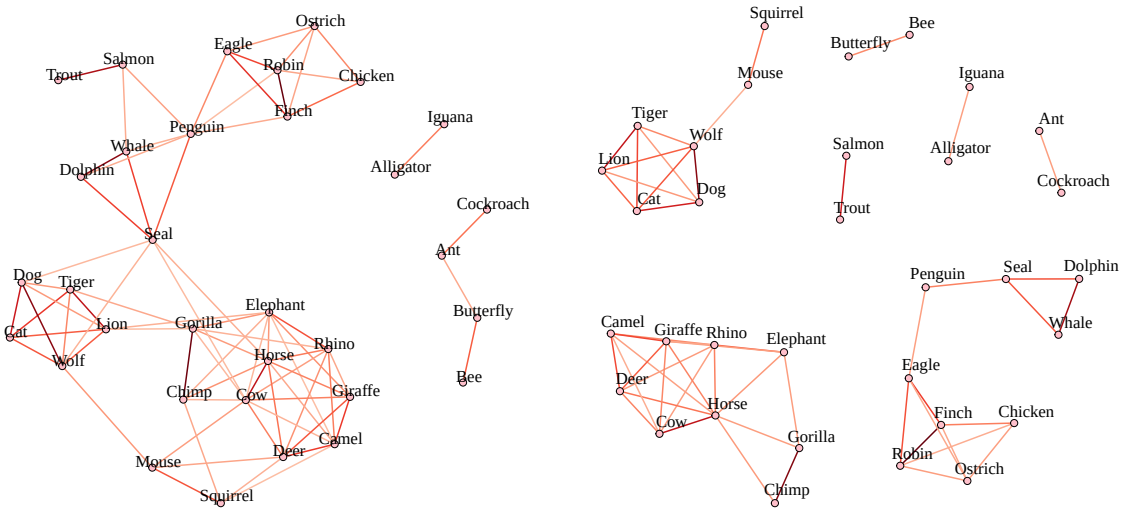
4.3 Experiments on Real Data

In this subsection, we apply our proposed model on some real data to see how it works on estimating the Gaussian graphical model with sparsity and clustering structure. The visualization is constructed using the software `spectralGraphTopology` [1].

Cancer genome data set. We consider the RNA-Seq Cancer Genome Atlas Research Network (Weinstein et al., 2013; Kumar et al., 2020). In the data set, there are $n = 801$ labeled samples, and each of them has $p = 20531$ features. The data set consists of five types of cancer, which are labeled with dots with five colors in the figure. Our goal is to study the similarities among the samples based on the given features assuming that we do not know the true labels. We apply the unconstrained model (2) with $\rho = 0.09$ and $\lambda = \rho/n^2$. The problem is solved by the two-phase algorithm in 59 seconds. Figure 5 presents the visualization of the estimated results of our estimator and the sparse Gaussian graphical

1. `https://CRAN.R-project.org/package=spectralGraphTopology`

model. Note that as we set $\lambda = \rho/n^2$ in our estimator (2), both of these two models contain one tuning parameter, which allows for a fair comparison. The edges between the vertices represent the similarities between them. That is, there is an edge between dot i and dot j if and only if the estimated Σ^{-1} satisfies $(\Sigma^{-1})_{ij} \neq 0$. One can see from the figure that the penalty on the pairwise differences that we add to the original sparse Gaussian graphical model indeed helps us to give better estimation of the graph and detect more meaningful clusters. For the estimated result of the sparse Gaussian graphical model in Figure 5(a), the clustering result is not clear and there are many redundant edges. As for the result of our proposed model in Figure 5(b), we can see that the samples are clustered into true groups except for four samples. The clustering result is consistent with the label information and the samples in different groups are completely separated.



(a) Sparse Gaussian graphical model (b) Our proposed model

Figure 6: Visualization of the estimated result for the Animals data set.

Animals data set. Following the idea in Egilmez et al. (2017), we also present some illustrations on how our model performs on categorical (non-Gaussian) data. To deal with the categorical data, we compute the input matrix C as the sum of the sample covariance matrix and the identity matrix scaled by $1/3$, where the $1/3I_n$ term is added based on the variational Bayesian approximation result in Banerjee et al. (2008) for binary data. We use the Animals data set (Kemp and Tenenbaum, 2008; Egilmez et al., 2017; Kumar et al., 2020) consisting of binary values which are answers to $p = 102$ questions for $n = 33$ animals. In the graph, vertices denote animals and edge weights represent the similarities between them. We aim to find the similarities among the animals. Since the conditional independence pattern is unknown in this real application, we apply the unconstrained model (2) with $\rho = 0.05$ and $\lambda = \rho/n^2$, which means that there is only one tuning parameter ρ involved in our estimator. The problem is solved by our two-phase algorithm within 1 second. The visualization of the estimated graphs by our model and the sparse Gaussian graphical model is presented in Figure 6. One can see from the estimated result by our

model that the animals are clustered into various meaningful groups. For example, the cluster of animals consisting of Horse, Elephant, etc, are large herbivorous mammals, while the cluster of animals consisting of Tiger, Lion, etc, are carnivorous mammals.

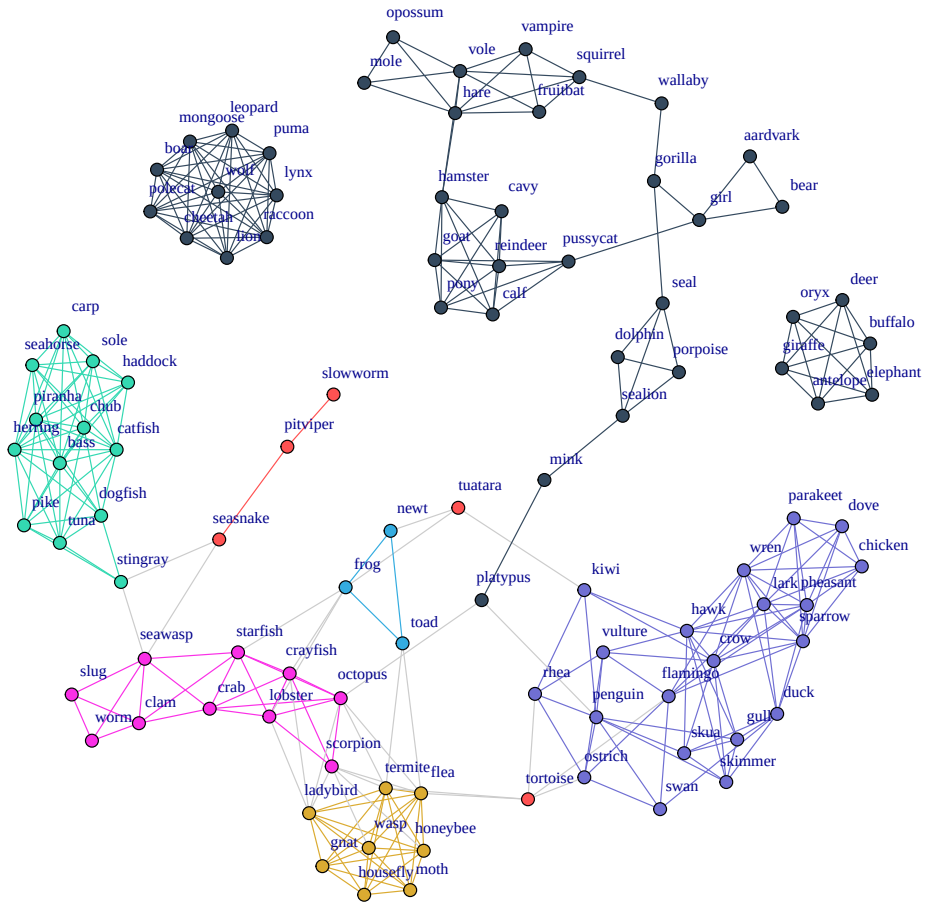


Figure 7: Visualization of the estimated result for the Zoo data set.

Zoo data set. We consider the Zoo data set from the UCI Machine Learning Repository, which contains $n = 100$ animals and each animal has $p = 17$ Boolean-valued attributes. Note that in this data set, the sample size p is much smaller than the number of animals n, which makes it difficult to estimate the concentration matrix. The data set contains seven types of animals which are known. To be specific, the set contains 41 kinds of mammals, 20 kinds of birds, 5 kinds of reptiles, 13 kinds of fish, 3 kinds of amphibians, 8 kinds of bugs and 10 invertebrates. Each type is labeled in the figure by a different color: black, violet, red, green, blue, yellow and pink, respectively. Since the data is the categorical data, we use the same technique as the case for the Animals data set to compute the input matrix C as the sum of the sample covariance matrix and the identity matrix scaled by $1/3$. In the experiment we take $\rho = 0.05$ and $\lambda = \rho/n^2$. The problem is solved by the two-phase algorithm within 1 second. We compare the clustering result of the model (2) with the true groups in Figure 7. As one can see, the animals belonging to each group are clustered

together except for the reptiles. Since the sample size is much smaller than the number of animals in this data set, there exist some wrong connections across different clusters, which are indicated by the grey colored edges in the figure. Some of the wrong connections are consistent with our usual expectation. For example, there exists an edge between platypus and penguin since they are both vertebrate warm blooded animals that lay eggs. Note that the animals belonging to the relatively large groups: mammals, birds and fish, are clearly separated. In addition, the cluster consisting of mammals is further divided into three sub groups: the carnivorous mammals like lion, the large herbivorous mammals like elephant, and the small herbivorous mammals like squirrel.

5. Conclusion

In this paper, we propose a new model to estimate the concentration matrix via learning the sparsity and hidden clustering structure. In addition, we design an efficient two-phase algorithm to solve the underlying large scale convex optimization to high accuracy. Specifically, we design the sGS-ADMM in the first phase to generate an initial point to warm start the second phase of the pALM, where each of its subproblems is solved by the SSN method. Numerical experiments on both synthetic data and real data have demonstrated the good performance of our model, as well as the efficiency and robustness of our proposed algorithm.

Acknowledgments

We would like to thank the Action Editor Dr. David Wipf and the anonymous referees for their helpful suggestions that have significantly improved our paper.

The research of Meixia Lin is supported by the Singapore University of Technology and Design under MOE Tier 1 Grant SKI 2021_02_08. The research of Defeng Sun is supported by Hong Kong Research Grants Council under the NSFC/RGC Joint Research grant N_PolyU504/19 and the GRF grant 15307822 and Shenzhen Research Institute of Big Data, Shenzhen 518000 grant 2019ORF01002. The research of Kim-Chuan Toh is supported by the Ministry of Education of Singapore under Academic Research Fund Grant number: MOE2019-T3-1-010. The research of Chengjing Wang is supported by the Zhejiang Provincial Natural Science Foundation of China under Grant LTGY23H240002 and the National Natural Science Foundation of China under Grant U21A20169.

Appendix A. Proof of Proposition 3(c)

For any $u \in \mathbb{R}^{\bar{n}}$, we can see that

$$q^*(u) = \sup_{x\in\mathbb{R}^{\bar{n}}} \Big\{\langle x, u\rangle - \rho\|x\|_1 - \lambda\langle w, P_x x\rangle\Big\} = \sup_{\hat{x}\in\mathcal{D}} \sup_{P\in\mathcal{P}_n} \Big\{\langle P\hat{x}, u\rangle - \rho\|\hat{x}\|_1 - \lambda\langle w, \hat{x}\rangle\Big\},$$

where $\mathcal{P}_n$ denotes the set of $n \times n$ permutation matrices. Given $\hat{x} \in \mathcal{D}$, we have that

$$\langle P\hat{x}, u\rangle = \langle \hat{x}, P^T u\rangle \le \langle \hat{x}, P_u u\rangle, \quad \forall P \in \mathcal{P}_n,$$

according to Chebyshev's sum inequality. Therefore

$$q^*(u) = \sup_{\hat{x}\in\mathcal{D}} \left\{\langle \hat{x}, P_u u - \lambda w\rangle - \rho\|\hat{x}\|_1\right\}$$
$$= \begin{cases} 0, & \text{if } \sum_{i=1}^{k}((P_u u-\lambda w)_i - \rho) \le 0, \quad \sum_{i=k}^{n}((P_u u-\lambda w)_i + \rho) \ge 0, \quad \forall k = 1,\cdots,\bar{n}, \\ +\infty, & \text{otherwise,} \end{cases}$$

where the last equality follows from the following lemma.

Lemma 12 *Given any $y \in \mathbb{R}^{\bar{n}}$ and $\rho > 0$, it holds that*

$$\sup_{x\in\mathcal{D}} \left\{\phi(x) = \langle x, y\rangle - \rho\|x\|_1\right\} = \begin{cases} 0, & \textit{if } \sum_{i=1}^{k}(y_i-\rho) \le 0, \quad \sum_{i=k}^{\bar{n}}(y_i+\rho) \ge 0, \quad \forall k = 1,\cdots,\bar{n}, \\ +\infty, & \textit{otherwise,} \end{cases}$$

where the convex set $\mathcal{D}$ is defined in Proposition 3.

Proof Suppose for all $k = 1,\cdots,\bar{n}$,

$$\sum_{i=1}^{k}(y_i - \rho) \le 0, \quad \sum_{i=k}^{\bar{n}}(y_i + \rho) \ge 0.$$

For any $x \in \mathcal{D}$, we have

$$x_1 \ge x_2 \ge \cdots \ge x_{\bar{n}-1} \ge x_{\bar{n}}.$$

There must exists $j \in \{0, 1, \cdots, \bar{n}\}$ such that

$$+\infty = x_0 \ge x_1 \ge \cdots \ge x_j \ge 0 \ge x_{j+1} \ge \cdots \ge x_{\bar{n}} \ge x_{\bar{n}+1} = -\infty.$$

Define $u \in \mathbb{R}^{\bar{n}}$ as

$$u_1 = x_1 - x_2, \quad \cdots, \quad u_{j-1} = x_{j-1} - x_j, \quad u_j = x_j,$$
$$u_{j+1} = -x_{j+1}, \quad u_{j+2} = x_{j+1} - x_{j+2}, \quad \cdots, \quad u_{\bar{n}} = x_{\bar{n}-1} - x_{\bar{n}}.$$

Then we can see that $u \ge 0$ and

$$x_1 = u_1 + u_2 + \cdots + u_j, \quad x_2 = u_2 + \cdots + u_j, \quad x_j = u_j,$$
$$x_{j+1} = -u_{j+1}, \quad x_{j+2} = -u_{j+1} - u_{j+2}, \quad x_{\bar{n}} = -u_{j+1} - \cdots - u_{\bar{n}}.$$

Thus it holds that

$$\phi(x) = \sum_{i=1}^{j} x_i(y_i - \rho) + \sum_{i=j+1}^{\bar{n}} x_i(y_i + \rho)$$
$$= \sum_{i=1}^{j}(u_i + \cdots + u_j)(y_i - \rho) + \sum_{i=j+1}^{\bar{n}}(-u_{j+1} - \cdots - u_i)(y_i + \rho)$$
$$= \sum_{i=1}^{j} u_i(y_1 + \cdots + y_i - i\rho) + \sum_{i=j+1}^{\bar{n}}(-u_i)(y_i + \cdots + y_{\bar{n}} - (\bar{n} + 1 - i)\rho) \le 0.$$

Together with the fact that $0 \in \mathcal{D}$ and $\phi(0) = 0$, we have

$$\sup_{x \in \mathcal{D}} \phi(x) = 0.$$

Next we consider the case when there exists $k_0 \in \{1, \cdots, \bar{n}\}$ such that $\sum_{i=1}^{k_0}(y_i - \rho) > 0$. Take $x \in \mathcal{D}$ as

$$x_1 = x_2 = \cdots = x_{k_0} = a > 0, \quad x_{k_0+1} = x_{k_0+2} = \cdots = x_{\bar{n}} = 0.$$

Then

$$\phi(x) = a \sum_{i=1}^{k_0}(y_i - \rho) \to +\infty, \quad \text{as} \quad a \to +\infty,$$

which means $\sup_{x \in \mathcal{D}} \phi(x) = +\infty$. Similarly, it could be proved that when there exists $k_0 \in \{1, \cdots, \bar{n}\}$ such that $\sum_{i=k_0}^{n}(y_i + \rho) < 0$, we have $\sup_{x \in \mathcal{D}} \phi(x) = +\infty$. This completes the proof. ■

References

Christophe Ambroise, Julien Chiquet, and Catherine Matias. Inferring sparse Gaussian graphical models with latent structure. *Electronic Journal of Statistics*, 3:205–238, 2009.

Onureena Banerjee, Laurent El Ghaoui, and Alexandre d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine Learning Research*, 9(Mar):485–516, 2008.

Michael J. Best and Nilotpal Chakravarti. Active set algorithms for isotonic regression; A unifying framework. *Mathematical Programming*, 47(1-3):425–439, 1990.

Howard D. Bondell and Brian J. Reich. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR. *Biometrics*, 64(1):115–123, 2008.

Caihua Chen, Bingsheng He, Yinyu Ye, and Xiaoming Yuan. The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent. *Mathematical Programming*, 155(1-2):57–79, 2016.

Liang Chen, Defeng Sun, and Kim-Chuan Toh. An efficient inexact symmetric Gauss–Seidel based majorized ADMM for high-dimensional convex composite conic programming. *Mathematical Programming*, 161(1-2):237–270, 2017.

Joachim Dahl, Lieven Vandenberghe, and Vwani Roychowdhury. Covariance selection for nonchordal graphs via chordal embedding. *Optimization Methods and Software*, 23(4): 501–520, 2008.

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

Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. Information-theoretic metric learning. In *International Conference on Machine Learning (ICML)*, pages 209–216, 2007.

Inderjit S. Dhillon and Joel A. Tropp. Matrix nearness problems with Bregman divergences. *SIAM Journal on Matrix Analysis and Applications*, 29(4):1120–1146, 2008.

Xiaowen Dong, Dorina Thanou, Pascal Frossard, and Pierre Vandergheynst. Learning Laplacian matrix in smooth graph signal representations. *IEEE Transactions on Signal Processing*, 64(23):6160–6173, 2016.

John Duchi, Stephen Gould, and Daphne Koller. Projected subgradient methods for learning sparse Gaussians. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 153–160, 2012.

Hilmi E. Egilmez, Eduardo Pavez, and Antonio Ortega. Graph learning from data under Laplacian and structural constraints. *IEEE Journal of Selected Topics in Signal Processing*, 11(6):825–841, 2017.

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

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Wenjiang Fu and Keith Knight. Asymptotics for lasso-type estimators. *The Annals of Statistics*, 28(5):1356–1378, 2000.

Charles J. Geyer. On the asymptotics of constrained M-estimation. *The Annals of Statistics*, 22(4):1993 – 2010, 1994.

Jiye Han and Defeng Sun. Newton and quasi-Newton methods for normal maps with polyhedral sets. *Journal of Optimization Theory and Applications*, 94(3):659–676, 1997.

Jean-Baptiste Hiriart-Urruty, Jean-Jacques Strodiot, and V. Hien Nguyen. Generalized Hessian matrix and second-order optimality conditions for problems with $C^{1,1}$ data. *Applied Mathematics and Optimization*, 11(1):43–56, 1984.

Søren Højsgaard. Graphical models for sparse data: Graphical Gaussian models with vertex and edge symmetries. In *Proceedings in Computational Statistics (COMPSTAT)*, pages 105–116. Springer, 2008.

Søren Højsgaard and Steffen L. Lauritzen. Restricted concentration models – graphical Gaussian models with concentration parameters restricted to being equal. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*. Citeseer, 2005.

Søren Højsgaard and Steffen L. Lauritzen. Graphical Gaussian models with edge and vertex symmetries. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5):1005–1027, 2008a.

Søren Højsgaard and Steffen L. Lauritzen. Inference in graphical Gaussian models with edge and vertex symmetries with the gRc package for R. *Journal of Statistical Software*, 23:1–26, 2008b.

Jean Honorio, Dimitris Samaras, Nikos Paragios, Rita Goldstein, and Luis E. Ortiz. Sparse and locally constant Gaussian graphical models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 745–753, 2009.

Mohammad Javad Hosseini and Su-In Lee. Learning sparse Gaussian graphical models with overlapping blocks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3808–3816, 2016.

Cho-Jui Hsieh, Mátyás A Sustik, Inderjit S. Dhillon, and Pradeep Ravikumar. QUIC: Quadratic approximation for sparse inverse covariance estimation. *Journal of Machine Learning Research*, 15(1):2911–2947, 2014.

Timothy R. Hughes, Matthew J. Marton, Allan R. Jones, Christopher J. Roberts, Roland Stoughton, Christopher D. Armour, Holly A. Bennett, Ernest Coffey, Hongyue Dai, Yudong D. He, Matthew J. Kidd, Amy M. King, Michael R. Meyer, David Slade, Pek Y. Lum, Sergey B. Stepaniants, Daniel D. Shoemaker, Daniel Gachotte, Kalpana Chakraburtty, Julian Simon, Martin Bard, and Stephen H. Friend. Functional discovery via a compendium of expression profiles. *Cell*, 102(1):109–126, 2000.

Vassilis Kalofolias. How to learn a graph from smooth signals. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 920–929. PMLR, 2016.

Charles Kemp and Joshua B. Tenenbaum. The discovery of structural form. *Proceedings of the National Academy of Sciences*, 105(31):10687–10692, 2008.

Brian Kulis, Mátyás Sustik, and Inderjit S. Dhillon. Learning low-rank kernel matrices. In *International Conference on Machine Learning (ICML)*, pages 505–512, 2006.

Sandeep Kumar, Jiaxi Ying, José Vinícius de Miranda Cardoso, and Daniel Palomar. Structured graph learning via Laplacian spectral constraints. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Sandeep Kumar, Jiaxi Ying, José Vinícius de M. Cardoso, and Daniel P. Palomar. A unified framework for structured graph learning via spectral constraints. *Journal of Machine Learning Research*, 21(22):1–60, 2020.

Steffen Lauritzen and Søren Højsgaard. Graphical models with edge and vertex symmetreis. In *Scientific Program; Abstracts of the 7th World Congress in Probability and Statistics*, page 135, 2008.

Steffen L. Lauritzen. *Graphical Models*, volume 17. Clarendon Press, 1996.

Xudong Li, Defeng Sun, and Kim-Chuan Toh. QSDPNAL: A two-phase augmented Lagrangian method for convex quadratic semidefinite programming. *Mathematical Programming Computation*, pages 1–41, 2018.

Xudong Li, Defeng Sun, and Kim-Chuan Toh. An asymptotically superlinearly convergent semismooth Newton augmented Lagrangian method for Linear Programming. *SIAM Journal on Optimization*, 30(3):2410–2440, 2020.

Meixia Lin, Yong-Jin Liu, Defeng Sun, and Kim-Chuan Toh. Efficient sparse semismooth Newton methods for the clustered lasso problem. *SIAM Journal on Optimization*, 29(3): 2026–2052, 2019.

Zhaosong Lu. Adaptive first-order methods for general sparse inverse covariance selection. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2000–2016, 2010.

Benjamin Marlin, Mark Schmidt, and Kevin Murphy. Group sparse priors for covariance estimation. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 383–392, 2012.

Benjamin M. Marlin and Kevin P. Murphy. Sparse Gaussian graphical models with unknown block structure. In *International Conference on Machine Learning (ICML)*, pages 705–712. ACM, 2009.

Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société mathématique de France*, 93:273–299, 1965.

Sebastian Petry, Claudia Flexeder, and Gerhard Tutz. Pairwise fused lasso. *Technical Report 102, Department of Statistics, University of Munich*, 2011.

Pradeep Ravikumar, Martin J Wainwright, Garvesh Raskutti, and Bin Yu. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935 – 980, 2011.

R. Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

R. Tyrrell Rockafellar. *Convex Analysis*, volume 36. Princeton University Press, 1997.

R. Tyrrell Rockafellar and Roger J.-B. Wets. *Variational Analysis*, volume 317. Springer Science & Business Media, 2009.

Leiba Rodman and Tamir Shalom. On inversion of symmetric Toeplitz matrices. *SIAM Journal on Matrix Analysis and Applications*, 13(2):530–549, 1992.

Katya Scheinberg, Shiqian Ma, and Donald Goldfarb. Sparse inverse covariance selection via alternating linearization methods. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 23. Curran Associates, Inc., 2010.

Yiyuan She. Sparse regression with exact clustering. *Electronic Journal of Statistics*, 4: 1055–1096, 2010.

Siqi Sun, Yuancheng Zhu, and Jinbo Xu. Adaptive variable clustering in Gaussian graphical models. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 931–939, 2014.

Siqi Sun, Hai Wang, and Jinbo Xu. Inferring block structure of graphical models in exponential families. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 939–947, 2015.

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

Kim-Chuan Toh. Primal-dual path-following algorithms for determinant maximization problems with linear matrix inequalities. *Computational Optimization and Applications*, 14(3):309–330, 1999.

Reha H. Tütüncü, Kim-Chuan Toh, and Michael J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming*, 95(2):189–217, 2003.

Chengjing Wang, Defeng Sun, and Kim-Chuan Toh. Solving log-determinant optimization problems by a Newton-CG primal proximal point algorithm. *SIAM Journal on Optimization*, 20(6):2994–3013, 2010.

John N. Weinstein, Eric A. Collisson, Gordon B. Mills, Kenna R. Mills Shaw, Brad A. Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, Joshua M. Stuart, and Cancer Genome Atlas Research Network. The cancer genome atlas pan-cancer analysis project. *Nature Genetics*, 45(10):1113, 2013.

Jiaxi Ying, José Vinícius de Miranda Cardoso, and Daniel Palomar. Nonconvex sparse graph learning under Laplacian constrained graphical model. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 7101–7113, 2020.

Xianxue Yu, Guoxian Yu, and Jun Wang. Clustering cancer gene expression data by projective clustering ensemble. *PloS One*, 12(2), 2017.

Ming Yuan and Yi Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

Xin-Yuan Zhao, Defeng Sun, and Kim-Chuan Toh. A Newton-CG augmented Lagrangian method for semidefinite programming. *SIAM Journal on Optimization*, 20(4):1737–1765, 2010.