Statistical Learning of High-Dimensional Directed Acyclic Graphical Models

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Yiping Yuan

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Advised by Xiaotong Shen

Feb, 2015

Acknowledgements

I would like to express my endless gratitude to my advisor, Dr. Xiaotong Shen, for supporting me over the years, for teaching me the true spirit of research and for guiding me in my personal development. I have learned a great deal from working with Dr. Shen. His vision and encouragement helped me through hard times. My gratitude also goes to my other committee members, Dr. Wei Pan, Dr. Galin Jones and Dr. Charlie Geyer. It is very rewarding working with Dr. Pan, as his rigorous academic attitude and sharp questions are always inspiring. I want to thank Dr. Jones and Dr. Geyer for their advice and encouragement. In addition, I would like to thank Dr. Zizhuo Wang for bringing an outsider's perspective and enriching the thesis with his expertise in optimization.

I would like to thank my friends at University of Minnesota with whom I have had the pleasure of working over the years. Thanks go to Yunzhang Zhu, Sen Yuan, Ben Sherwood, Yi Yang, Xin Zhang, Jie Ren as well as my other fellow students in the School of Statistics.

I also want to thank my parents for supporting me. Although I live across the pacific ocean over the years, I always feel their love and care. Finally, no words could express my special thanks to my beloved wife Jing. It has been a wonderful journey going through my five-year study with you.

Dedication

This dissertation is dedicated to my family, especially,

to my brilliant and outrageously loving and supportive wife, Jing Zhang;

to my always encouraging and supportive parents, Zhongwen Yuan and Li Cao, and parents-in law, Aibao and Chunhua Zhang.

Abstract

Directed acyclic graphs (DAGs) are widely used to describe directional relations among interacting units. Directional relations are estimated by reconstructing a DAG's structure, which is a great challenge when the total ordering of a DAG is unknown. In such a situation, existing methods such as the neighborhood and search-and-score methods suffer greatly, as the overall estimation error accumulates super-exponentially in the number of nodes, especially when a local/sequential approach enumerates edge directions by testing or optimizing a criterion locally. In other words, a local method may break down even for moderately sized graphs. In this thesis, we propose a novel approach to simultaneously identify all estimable directed edges as well as model parameters jointly. This approach uses constrained maximum likelihood with nonconvex constraints reinforcing acyclicity. Computationally, we develop a novel reduction method that constructs a set of active constraints (cubic in the number of nodes) from the super-exponentially many constraints. This, coupled with an alternating direction method of multipliers and a difference convex method, permits efficient computation for large graph learning. Theoretically, we show that the proposed method consistently reconstructs identifiable directions of the true graph, under a degree of reconstructability assumption. This goes beyond the strong faithfulness assumption, commonly used in the literature. Moreover, the method recovers the optimal performance of the oracle estimator in terms of parameter estimation. Numerically, the method compares favorably against its competitors.

Estimation of multiple directed graphs becomes challenging in the presence of inhomogeneous data, where directed acyclic graphs are used to represent causal relations among random variables. To infer causal relations among variables, we estimate multiple directed acyclic graphs given a known partial ordering in Gaussian graphical models. In particular, we propose a constrained maximum likelihood method with nonconvex constraints over elements and element-wise differences of adjacency matrices, for identifying the sparseness structure as well as detecting structural changes over adjacency matrices of the graphs. Computationally, we develop an efficient algorithm based on augmented Lagrange multipliers, the difference convex method, and a novel fast algorithm for solving convex relaxation subproblems. Numerical results suggest that the proposed method performs well against its alternatives for simulated and real data.

For an observational study, correct reconstruction of a DAG's structure from data is not always possible, because a DAG model is often not identifiable, which is the case for a Gaussian graphical model with unequal error variances. We study the problem of reconstruction of a DAG's structure with the help of intervention observations. In particular, we construct a constrained likelihood to regularize intervention in addition to adjacency matrices to identify a DAG's structure and remove redundant intervention variables. Importantly, we show that the constructed constrained likelihood yields correct reconstruction of a DAG's structure consistently provided that the candidate set of intervention variables includes the true informative ones. Computationally, we design efficient algorithms for implementation. In simulations, we show that the proposed method enables to lead higher accuracy of reconstruction with the help of interventional observations.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter, we briefly introduce related graph concepts and background knowledge, and then give an overview of the contribution of this thesis.

1.1 Definitions and Prelinimaries

Directed acyclic graphical models are widely used to represent and visualize directional relations, or parent-child relations, among interacting units, particularly in analyzing gene and social networks [1]. The graphical representation of the model is a *directed acyclic graph* (DAG), which, by definition, is a directed graph without directed cycles.

Major building blocks of a DAG model are nodes, which represent random variables and edges, which encode conditional dependence relations of the enclosing vertices. A DAG $G = (V, E)$ consists of a set of nodes $V = 1, \dots, p$ and a set of directed edges $E \subseteq V \times V$, that is, the edge set is a subset of ordered pairs of distinct nodes. In our setting, each node j represents a random variable X_j and an edge $(i, j) \in E$ can be denoted as $i \rightarrow j$.

If there is a directed edge $i \rightarrow j$, node i is said to be a parent of node j and node$j$ is called a child of node i. The set of parents of node i is denoted by pa_i. If there is a directed path from node i to node j, then node i is called an ancestor of j and j is called an descendant of i. It can be shown that the absence of any directed cycles is equivalent to the existence of an ordering of nodes $\{v_1, v_2, \dots, v_p\}$ such that all edges $v_i \rightarrow v_j$ have $i < j$. Later in this thesis, we will see how the difficulty of the DAG learning differs when the ordering of nodes is known and unknown.

The models encode graph-to-distribution correspondences through directed Markov properties. Let P denote the probabilistic distribution of $(X_1, X_2, \ldots, X_p)$.

Definition 1 *(Local Markov property) P is said to obey the local Markov property to the DAG G if every node is conditionally independent of its non-descendant, non-parent nodes given its parents.*

$$\forall i \in V : i \perp \{nd_i \backslash pa_i\} | pa_i$$

where nd_i are the non-descendants of i.

Definition 2 *(Factorization property) We say that P admits a factorization according to a DAG G if*

$$P(X_1, \ldots, X_p) = \prod_{j=1}^{p} P(X_j | pa_j).$$

In our case of a multivariate Gaussian distribution, both properties in Definition 1 and Definition 2 are equivalent. For more details see [2].

In this thesis, we focus on Gaussian DAGs. We assume throughout the thesis that $\boldsymbol{X} = (X_1, X_2, \ldots, X_p)$ follows multivariate normal distribution. The Gaussian assumption implies that $E(X_i|pa_i)$ is linear in pa_i. Statistically, the model can be written as

$$X_j = \sum_{k \in pa_j} A_{jk} X_k + Z_j, \quad Z_j \sim N(0, \sigma_j^2); \quad j = 1, \ldots, p,$$

where Z_j represents random error and $\boldsymbol{A}$ is the parameter matrix of interest. This thesis is devoted to infer the model from data observed for $\boldsymbol{X}$.

1.2 Overview of the contribution in this thesis

In this thesis, we develop a simultaneous reconstruction approach to estimate the configuration of a DAG and model parameters jointly, especially for the high-dimensional cases. This approach overcomes the difficulties of local and sequential approaches in the literature. Specifically, we propose a constrained likelihood for reconstructing a DAG without a known ordering in Chapter 2. Our novel treatment to this seemingly impossible problem is utilizing a property of doubly stochastic matrices to derive an equivalent form involving only $p^3 - p^2$ active constraints, c.f., Theorem 1. This, combined with a constrained alternating direction method of multipliers [3] and difference convex programming, makes it possible to solve this problem, thus leading to efficient computation involving a complexity of order $O(p^3)$. Theoretically, we develop a theory to quantify what the proposed method can accomplish, where the focus is equal error variance for identifiable DAG models [4]. We show that it consistently reconstructs the true directed acyclic graph under a degree of reconstructability assumption (2.19). This assumption, similar to the "beta-min" condition [5], requires that the minimum separation between the target and candidate models exceeds a certain threshold. Note that the corresponding probabilistic distribution may not be identifiable in general in the presence of equivalence classes of DAGs [6]. With regard to estimating model parameters, it recovers the optimal performance of the oracle estimator.

We next study multiple DAG learning when data are inhomogeneous and proposed a maximum likelihood method to jointly estimate multiple DAGs with a known ordering. To achieve our goal of learning graphical structures, we construct two nonconvex constraints based on the truncated L_1-function (TLP, [7]), as a computational surrogate of the L_0-function, with one constraint imposing sparseness and the other encouraging a common structure. Computationally, with difference convex programming and augmented Lagrange multipliers, nonconvex minimization is solved through a sequence of convex subproblems iteratively. For each subproblem, we develop a fast algorithm to treat a constrained L_1-problem, which we call pairwise coordinate descent algorithm.

In Chapter 4, we study how incorporating interventional data could make a difference in identifying causal directions. For an observational study, a DAG model is often not identifiable. In such cases, correct reconstruction of a DAG's structure from data is impossible. In Chapter 4, we study the problem of reconstruction of a DAG's structure with the help of intervention observations. In particular, we construct a constrained likelihood to regularize intervention in addition to adjacency matrices to identify a DAG's structure and remove redundant intervention variables. Importantly, we show that the constructed constrained likelihood yields correct reconstruction of a DAG's structure consistently, provided that the candidate set of intervention variables includes the true informative ones.

Finally, Chapter 5 summarizes our findings and discusses potential areas for future work. Technical details and proofs can be found in the Appendix.

Chapter 2

Constrained likelihood for reconstructing a directed acyclic Gaussian graph

In this chapter, we introduce the constrained likelihood approach for reconstructing a directed acyclic Gaussian graph. Section 2.1 covers existing methods. Section 2.2 introduces the proposed method. Section 2.3 is devoted to the computational development of the proposed method. Section 2.4 presents theoretical results concerning structure pursuit and parameter estimation. Section 2.5 performs simulation studies to compare with several competing methods, and analyzes a protein network. Section 2.6 discusses the methodology. Finally, the Appendix contains technical proofs.

2.1 Background of DAG Learning

In the literature, most existing methods are designed for a low-dimensional situation, in which the size of a graph is relatively small compared with the sample size. Major approaches emerge in two categories. The first uses multiple local conditional independence tests sequentially [8, 9, 10] to enumerate possible directions through the local Markov property. Such methods usually have a worst case exponential complexity in the number of nodes, including the most popular "PC" algorithm. [11] proposed a high-dimensional modification to the original "PC". This modified "PC" has a complexity of order $O(p^q)$ with p nodes, where q is the maximal neighborhood size. However, this complexity becomes super-exponential when q is large $q = O(p)$ even if the graph is sparse; see Example 2 in Section 4. The second, referred to as "search-and-score", optimizes a goodness-of-fit measure for possible directions in a neighborhood of interest [12, 13, 14]. Computationally, search-and-score methods suffer greatly from the curse of dimensionality due to $O(p!2^{p^2})$ candidate DAGs of p labeled nodes [15]. Moreover, due to their sequential nature, they tend to yield unstable and deteriorating performance for a large graph. The major difficulty comes from acyclicity, as pointed out in [16, 17, 18] and recently [19, 20]. As a result, the recent development of DAG models lags behind that of undirected graphs [21] in the high-dimensional situation. Nevertheless, we would like to mention two recent developments on special situations. [22] proposed a L_1-penalization method for learning the DAG model with a known topological ordering of the nodes. [23] used a L_1-penalization method for interventional data.

Nonconvex optimization for exact learning of a DAG's structure, for instance, [24], focuses on identifying an exact global optimizer for this NP-hard problem. Certain approximations are involved with a worst-case suboptimality bound for an anytime solution, which is an approximate solution when the algorithm can be interrupted at any time before it takes too long. As a result, it is rather difficult, if not impossible, to treat even a moderate graph with more than one hundred nodes. Recently, mixed integer linear programming (MILP) has been used together with certain branch-and-bounds [25, 26, 27, 28].When the maximal number of parent nodes is restricted to one or two, such a method may handle a larger graph than the previous exact methods. Again, such a restriction limits the scope of application. In this article, based on a difference convex algorithm, we seek a good optimizer that is shown to be local for fast convergence, although such an optimizer may be sometimes global. This is in contrast to the counterpart of our DC algorithm, such as Breiman and Cutler's outer approximation method [29] that guarantees globality of a solution at an expense of slow convergence. Importantly, as showed in Table 3, our DC algorithm may have a good chance to identify a global optimizer. This aspect of a DC algorithm has been previously noted in [30] for a linearly constrained indefinite quadratic problem.

In addition to the computational challenges in the high-dimensional situation, theoretical challenges remain. First, it is challenging statistically in that the error of correct reconstruction of a DAG may grow super-exponentially in p. Roughly, this error is no less than $\min(1, \exp(p \log p - b(\varepsilon)n))$ in view of Bahadur's lower bound for the error of each test [31], where $b(\varepsilon)$ is a constant describing the least favorable situation defined by the Kullback-Leibler information. In other words, any local and sequential method, particularly the PC algorithm and its variants, may break down, which occurs roughly when $p \log p$ significantly exceeds n. Second, there is paucity of theory to guide practice for reconstruction of a DAG. One relevant theory is on consistent reconstruction of a DAG's structure for the PC-algorithm [11], which relies on one key assumption, called "strong faithfulness" [8, 32]. Unfortunately, this assumption is rather restrictive, because it induces a small set of distributions as pointed out in [33]. Thus one open problem is whether any computationally feasible method can lead to consistent reconstruction of a DAG beyond the "strong faithfulness".

In this chapter, we develop a simultaneous reconstruction approach to estimate the configuration of a DAG and model parameters jointly. This approach overcomes the aforementioned difficulties of local and sequential approaches. Specifically, we propose a constrained likelihood with a set of $O(p^p)$ nonconvex constraints, to quantify the parameter space for DAG's. This reinforces the local Markov property that is crucial to discovering directional relations. Our novel treatment to this seemingly impossible problem is utilizing a property of doubly stochastic matrices to derive an equivalent form involving only $p^3 - p^2$ active constraints, c.f., Theorem 1. This, combined with a constrained alternating direction method of multipliers [3] and difference convex programming, makes it possible to solve this problem, thus leading to efficient computation involving a complexity of order $O(p^3)$. Theoretically, we develop a theory to quantify what the proposed method can accomplish, where the focus is equal error variance for identifiable DAG models [4]. We show that it consistently reconstructs the true directed acyclic graph under a degree of reconstructability assumption (2.19). This assumption, similar to the "beta-min" condition [5], requires that the minimum separation between the target and candidate models exceeds a certain threshold. Note that the corresponding probabilistic distribution may not be identifiable in general in the presence of equivalence classes of DAGs [6]. With regard to estimating model parameters, it recovers the optimal performance of the oracle estimator.

2.2 Statistical methods

A DAG model encodes a joint probability distribution of a random vector $(X_1, \dots, X_p)$, whose nodes and directed edges represent $X_1, \dots, X_p$ and parent-child dependence relations between any two variables. The parents of X_j, denoted as pa_j, is the set of variables with a direction towards X_j in the graph. The model factorizes the joint distribution of $(X_1, \dots, X_p)$, $P(X_1, \dots, X_p)$, into a product of conditional distributions of each variable given its parents, that is, $\prod_{j=1}^{p} P(X_j|\text{pa}_j)$, where pa_j denotes a parent set of X_j and is defined to be empty if X_j has no parents. This factorization property is equivalent to the local Markov property [34] in the DAG case, and is closely related to antedependence [35], which has been widely used in time series and longitudinal data analysis.

A DAG over nodes $\{1, \cdots, p\}$ is uniquely defined by an $p \times p$ adjacency matrix $\boldsymbol{A}$ in which a nonzero jk-th element A_{jk} of $\boldsymbol{A}$ corresponds to a directed edge from parent node k to child note j with its value A_{jk} indicating the strength of the relation. The DAG does not contain a dicycle, where existence of a dicycle for a directed graph destroys the local Markov property of a DAG.

2.2.1 DAG parameter space

Most statistical methods focus on construction by optimizing a suitable cost function, since the parameter space is usually a simple convex space, for example, the $\mathbb{R}^p$ space for regression and the positive semidefinite cone $\mathbb{S}_+^p = \{\boldsymbol{A} \in \mathbb{R}^{p \times p} | \boldsymbol{A} = \boldsymbol{A}^T, \boldsymbol{A} \succeq 0\}$ for Gaussian undirected graphical models [36]. In contrast, the parameter space of the Gaussian DAG models is defined as $\{\boldsymbol{A} \in \mathbb{R}^{p \times p} : G(\boldsymbol{A}) \text{ is a DAG}\}$, which is nonconvex as a result of nonconvex constraints reinforcing acyclicity of a graph [5]. Yet, characterization of the parameter space remains an open problem. In what is to follow, we develop a method to deal with such an irregular parameter space.

To introduce acyclicity constraints, denote a directed cycle (dicycle) by $(j_1, \dots, j_L)$, a sequence of indices of nodes, where L is the length of the dicycle, and j_1 is required to be the smallest index ($j_1 = \min_{1 \le k \le L} j_k$) so that the dicycle is uniquely defined. For example, a dicycle $1 \leftarrow 2 \leftarrow 3 \leftarrow 1$ is denoted as $(1, 2, 3)$. A DAG, by definition, does not contain a dicycle. This implies that the number of directed edges is smaller than the number of involved nodes in every possible dicycle [37]. Let $I(\cdot)$ be the indicator function. To prevent a dicycle, say $(1, 2, 3)$, from occurring, we introduce a constraint $I(A_{1,2} \neq 0) + I(A_{2,3} \neq 0) + I(A_{3,1} \neq 0) \leq 2$ to require that the number of directed edges be smaller than the number of involved nodes in every possible cycles. Note that this requirement is necessary and sufficient. On this basis, we introduce constraints on entries of $\boldsymbol{A}$ to reinforce acyclicity of any order:

$$\sum_{j_1 = j_{L+1}: 1 \leq k \leq L} I(A_{j_k j_{k+1}} \neq 0) \leq L - 1; \text{ any dicycle } (j_1, \ldots, j_L), L = 2, \ldots, p, \tag{2.1}$$

where order L is the number of nodes in a possible dicycle. It is important to remark that any orders of dicycle are permissible if the corresponding constraints are reinforced in (2.1). For instance, any dicycle of order exceeding 2 are allowed if $L = 3, \cdots, p$ are removed from (2.1). Moreover, $L = 1$ corresponds to self-loops, which is characterized by nonzero diagonals. Critically, the total number of constraints in (2.1) is $\binom{p}{2} + \binom{p}{3}2! + \cdots + \binom{p}{p}(p-1)! = O(p^p)$, which is super-exponential in p. The parameter space of Gaussian DAG models is thus defined by the $O(p^p)$ DAG constraints over $\boldsymbol{A}$.

Next we present our main result on constraint reduction to reduce the $O(p^p)$ constraints in (2.1) to $p^3 - p^2$ constraints in (2.2). This result is based on duality and properties of permutation matrices, and can be used with any optimizing function.

Theorem 2.2.1 *(Construction of a set of active constraints) The adjacency matrix $\boldsymbol{A}$ satisfies the acyclicity constraints in* (2.1) *if and only if there exists a $p \times p$ dual variable matrix $\boldsymbol{\lambda} = (\lambda_{jk})_{p \times p} \in \mathbb{R}^{p \times p}$ such that the following constraints are satisfied by $\boldsymbol{A}$.*

$$\lambda_{ik} + I(j \neq k) - \lambda_{jk} \geq I(A_{ij} \neq 0); i, j, k = 1, \ldots, p, i \neq j. \tag{2.2}$$

In (2.2), there are $p^3 - p^2$ constraints over $(\boldsymbol{A}, \boldsymbol{\lambda})$ through additional slack variables. This allows us to not only reduce super-exponentially many constraints over $\boldsymbol{A}$ to $p^3 - p^2$ active constraints over $(\boldsymbol{A}, \boldsymbol{\lambda})$ but also achieve simplicity in that each constraint involves only one parameter in $\boldsymbol{A}$ and is linear in λ_{jk} and $I(A_{ij} \neq 0)$.

2.2.2 Constrained maximum likelihood

Statistically, we embed directional effects induced by directed edges through $\boldsymbol{A}$ into a structural equation model. A structural model can be written as:

$$X_j = f_j(\mathrm{pa}_j, Z_j), \quad j = 1, \ldots, p, \tag{2.3}$$

where Z_j is latent error representing unexplained variation in each node, and the local Markov property is defined through parents and latent variables in (2.3).

Now consider a Gaussian structural equation model in which each $f_j(\cdot, \cdot)$ in (2.3) becomes linear in (pa_j, Z_j), and each Z_j follows normal distribution $N(0, \sigma^2)$:

$$X_j = \sum_{k \neq j} A_{jk} X_k + Z_j, \quad Z_j \sim N(0, \sigma^2); \quad j = 1, \ldots, p, \tag{2.4}$$

where A_{jk} is 0 when $k \notin \mathrm{pa}_j$. In (2.4), our objective is to estimate parameters $\boldsymbol{A}$ subject to the requirement that $\boldsymbol{A}$ defines a DAG. This enables us to determine zero-entries of $\boldsymbol{A}$ to identify all parent-child relations as well as to estimate the strengths of the relations defined by nonzero-entries of $\boldsymbol{A}$ simultaneously. Note that individual means can be incorporated by adding intercepts into (2.4) but it is less relevant to reconstruction. For simplicity, we therefore set the means to be zero in what is to follow. Note that in (2.4) an equal variance for Z_j's leads to identifiable DAGs [4]. A more general case with different error variances will be discussed in Section 4.

Given $n \times p$ data matrix $\mathbf{X}$ sampled from (2.4), with its ij-th entry x_{ij} being the i-th observation on the j-th node, the negative loglikelihood, after dropping a constant term, is

$$l(\boldsymbol{A}) = \frac{1}{2} \sum_{j=1}^{p} \sum_{i=1}^{n} \big(x_{ij} - \sum_{k \neq j} x_{ik} A_{jk}\big)^2, \tag{2.5}$$

which is convex in $\boldsymbol{A}$.

For estimation of $\boldsymbol{A}$, we impose a constraint to regularize sparsity of $\boldsymbol{A}$, in addition to constraints defined in (2.1). The first constraint controls nonzero entries of $\boldsymbol{A}$

$$\sum_{j \neq k} I(A_{jk} \neq 0) \leq K,$$

where $I(\cdot)$ is the indicator function, and K is an integer-valued tuning parameter controlling the degree of sparsity.

The DAG learning problem can be formulated as follows:

$$\min_{\boldsymbol{A}} l(\boldsymbol{A}) \tag{2.6}$$

$$\text{subject to } \textstyle\sum_{j \neq k} I(A_{jk} \neq 0) \leq K,$$

$$\textstyle\sum_{j_1 = j_{L+1}: 1 \leq k \leq L} I(A_{j_k j_{k+1}} \neq 0) \leq L - 1; \text{ any dicycle } (j_1, \ldots, j_L), L = 2, \ldots, p.$$

Using the result from Theorem 1, (2.6) is equivalent to

$$\min_{(\boldsymbol{A}, \boldsymbol{\lambda})} l(\boldsymbol{A}) \tag{2.7}$$

$$\text{subject to } \textstyle\sum_{j \neq k} I(A_{jk} \neq 0) \leq K, \tag{2.8}$$

$$\lambda_{ik} + I(j \neq k) - \lambda_{jk} \geq I(A_{ij} \neq 0); i, j, k = 1, \ldots, p, j \neq i. \tag{2.9}$$

Next we approximate the indicator functions to circumvent the difficulty of non-discontinuity in minimization. Specifically, in (4.4) and (2.9), we substitute the indicator functions by its computational surrogate $J_\tau(\cdot)$ where $\mathrm{J}_\tau(x) = \min(\frac{|x|}{\tau}, 1)$ is the truncated L_1-function (TLP) [7], which approximates the indicator function as $\tau \to 0^+$. This yields that

$$\min_{(\boldsymbol{A}, \boldsymbol{\lambda})} l(\boldsymbol{A}) \tag{2.10}$$

$$\text{subject to } \textstyle\sum_{1 \leq j \neq k \leq p} \mathrm{J}_\tau(A_{jk}) \leq K, \tag{2.11}$$

$$\lambda_{ik} + I(j \neq k) - \lambda_{jk} \geq \mathrm{J}_\tau(A_{ij}); i, j, k = 1, \ldots, p, j \neq i. \tag{2.12}$$

Minimization (2.10) subject to (2.11) and (2.12) involves $p^3 - p^2 + 1$ nonconvex constraints in (2.12). Yet, compared to the original formulation with indicator functions, $J_\tau(\cdot)$ is piecewise linear and can be decomposed into the difference of two convex functions. Next we solve this constrained minimization through difference convex programming and the alternating direction method of multipliers (ADMM).

2.3 Computation

This section develops our computational strategy to solve (2.10). Our strategy proceeds in two steps. First, we relax (2.11) and (2.12) using a sequence of approximations involving convex constraints, where each approximation is refined iteratively. Then we solve each convex subproblem with $p^3 - p^2 + 1$ linear constraints by employing a constrained alternating direction method of multipliers. The underlying process iterates until convergence.

For convex relaxation of constraints (2.11) and (2.12), we employ the difference convex (DC) programming. In particular, we decompose J_τ into a difference of two convex functions: $\mathrm{J}_\tau(z) = S_1(z) - S_2(z) = \frac{|z|}{\tau} - \max(\frac{|z|}{\tau} - 1, 0)$. On this ground, we construct a sequence of convex approximating sets iteratively by replacing S_2 in the decomposition at iteration m by its affine majorization at iteration $m-1$. Specifically, we solve

$$\begin{aligned} \min_{(\boldsymbol{A},\boldsymbol{\lambda})} l(\boldsymbol{A}) \qquad & \text{subject to} \sum_{i \neq j} |A_{ij}| w_{ij}^{(m-1)} \leq Z^{(m-1)}, \\ \tau\lambda_{ik} + \tau I(j \neq k) - \tau\lambda_{jk} \quad \geq \quad & |A_{ij}| w_{ij}^{(m-1)} + \tau(1 - w_{ij}^{(m-1)}); i, j, k = 1, \ldots, p, j \neq i, \end{aligned} \tag{2.13}$$

where $w_{ij}^{(m-1)} = I(|\hat{A}_{ij}^{(m-1)}| \leq \tau)$, and $\hat{\boldsymbol{A}}^{(m-1)}$ is the solution at iteration $m-1$; $1 \leq i, j \leq p$. $Z^{(m-1)} = \tau\left(K - \sum_{i \neq j}(1 - w_{ij}^{(m-1)})\right)$. At iteration m, we solve a minimization problem with $p^3 - p^2 + 1$ linear constraints.

To solve (2.13), we consider its equivalent form for efficient computation:

$$\begin{gathered} \min_{(\boldsymbol{A},\boldsymbol{\lambda})} l(\boldsymbol{A}) + \mu \textstyle\sum_{i \neq j} |A_{ij}| w_{ij}^{(m-1)} \text{ subject to} \\ \tau\lambda_{ik} + \tau I(j \neq k) - \tau\lambda_{jk} \geq |A_{ij}| w_{ij}^{(m-1)} + \tau(1 - w_{ij}^{(m-1)}); i, j, k = 1, \ldots, p, j \neq i, \end{gathered} \tag{2.14}$$

where μ is a nonnegative regularizer corresponding to $Z^{(m-1)}$ in (2.13). Their correspondence is as follows: Given a solution of (2.14) $(\tilde{\boldsymbol{A}}(\mu), \tilde{\boldsymbol{\lambda}}(\mu), \mu)$, $\left(\tilde{\boldsymbol{A}}(\mu), \tilde{\boldsymbol{\lambda}}(\mu), Z^{(m-1)}\right)$ is a global minimizer of (2.13), where $Z^{(m-1)} = \sum_{i \neq j} |\tilde{A}_{ij}(\mu)| w_{ij}^{(m-1)}$, and vice versa. Therefore, $Z^{(m-1)}$ can be obtained through bisection of μ such that $Z^{(m-1)} = \sum_{i \neq j} |\tilde{A}_{ij}(\mu)| w_{ij}^{(m-1)}$.

Based on our limited numerical experience, ADMM may significantly expedite convergence, although (2.14) may be solved by a quadratic programming solver. To proceed, let $\boldsymbol{\xi} = \{\xi_{ijk}\}_{p \times p \times p}$ be a slack variable tensor, converting inequality to equality constraints. Then (2.14) becomes

$$\begin{gathered} \min_{(\boldsymbol{A},\boldsymbol{\lambda})} l(\boldsymbol{A}) + \mu \textstyle\sum_{i \neq j} |A_{ij}| w_{ij}^{(m-1)} \\ \text{subject to } \tau\lambda_{ik} + \tau I(j \neq k) - \tau\lambda_{jk} - |A_{ij}| w_{ij}^{(m-1)} - \tau(1 - w_{ij}^{(m-1)}) - \xi_{ijk} = 0; \\ i, j, k = 1, \ldots, p, j \neq i, \xi_{ijk} \geq 0. \end{gathered} \tag{2.15}$$

Next we introduce $\boldsymbol{B}_{p\times p}$ to separate the differentiable from non-differentiable parts involving L_1-norm. Then the problem can be written in the form as

$$\begin{aligned} &\min_{(\boldsymbol{A},\boldsymbol{\lambda})} l(\boldsymbol{A}) + \mu \textstyle\sum_{i\neq j} |B_{ij}| w_{ij}^{(m-1)} \\ &\text{subject to } \tau\lambda_{ik} + \tau I(j \neq k) - \tau\lambda_{jk} - |B_{ij}|w_{ij}^{(m-1)} - \tau(1 - w_{ij}^{(m-1)}) - \xi_{ijk} = 0; \\ &i, j, k = 1, \ldots, p, j \neq i, \xi_{ijk} \geq 0, \boldsymbol{A} - \boldsymbol{B} = \boldsymbol{0}. \end{aligned} \tag{2.16}$$

Following [38], we obtain an augmented Lagrangian by introducing the scaled dual variable tensor $\boldsymbol{y} = \{y_{ijk}\}_{p\times p\times p}$ and the scale dual variable matrix $\boldsymbol{U} = \{y_{ij}\}_{p\times p}$:

$$\begin{aligned} L_\rho(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\lambda}, \boldsymbol{\xi}, \boldsymbol{y}, \boldsymbol{U}) &= l(\boldsymbol{A}) + \mu(\textstyle\sum_{i\neq j} |B_{ij}|w_{ij}^{(m-1)}) + \frac{\rho}{2}\|\boldsymbol{A} - \boldsymbol{B} + \boldsymbol{U}\|_F^2 \qquad (2.17) \\ &+ \textstyle\sum_k \sum_{i\neq j} \frac{\rho}{2} \left(|B_{ij}|w_{ij}^{(m-1)} + \tau(1 - w_{ij}^{(m-1)}) + \xi_{ijk} - \lambda_{ik} - \tau I(j \neq k) + \lambda_{jk} + y_{ijk}\right)^2. \end{aligned}$$

Iteratively, we solve (2.17) over six blocks $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\lambda}, \boldsymbol{\xi}, \boldsymbol{y}, \boldsymbol{U})$. At iteration step $s + 1$,

$$\begin{aligned} \boldsymbol{A}^{(s+1)} &= \text{argmin}_{\boldsymbol{A}} L_\rho(\boldsymbol{A}, \boldsymbol{B}^{(s)}, \boldsymbol{\phi}^{(s)}, \boldsymbol{\lambda}^{(s)}, \boldsymbol{\xi}^{(s)}, \boldsymbol{y}^{(s)}, \boldsymbol{U}^{(s)}), \\ \boldsymbol{B}^{(s+1)} &= \text{argmin}_{\boldsymbol{B}} L_\rho(\boldsymbol{A}^{(s+1)}, \boldsymbol{B}, \boldsymbol{\phi}^{(s)}, \boldsymbol{\lambda}^{(s)}, \boldsymbol{\xi}^{(s)}, \boldsymbol{y}^{(s)}, \boldsymbol{U}^{(s)}), \\ \boldsymbol{\lambda}^{(s+1)} &= \text{argmin}_{\boldsymbol{\lambda}} L_\rho(\boldsymbol{A}^{(s+1)}, \boldsymbol{B}^{(s+1)}, \boldsymbol{\phi}^{(s)}, \boldsymbol{\lambda}, \boldsymbol{\xi}^{(s)}, \boldsymbol{y}^{(s)}, \boldsymbol{U}^{(s)}), \\ \boldsymbol{\xi}^{(s+1)} &= \text{argmin}_{\{\xi_{ijk}\geq 0\}} L_\rho(\boldsymbol{A}^{(s+1)}, \boldsymbol{B}^{(s+1)}, \boldsymbol{\phi}^{(s)}, \boldsymbol{\lambda}^{(s+1)}, \boldsymbol{\xi}, \boldsymbol{y}^{(s)}, \boldsymbol{U}^{(s)}), \\ y_{ijk}^{(s+1)} &= y_{ijk}^{(s)} + |B_{ij}^{(s+1)}|w_{ij}^{(m-1)} + \tau(1 - w_{ij}^{(m-1)}) + \\ &\quad + \xi_{ijk}^{(s+1)} - \tau\lambda_{ik}^{(s+1)} - \tau I(j \neq k) + \tau\lambda_{jk}^{(s+1)}, \\ \boldsymbol{U}^{(s+1)} &= \boldsymbol{U}^{(s)} + \big(\boldsymbol{A}^{(s+1)} - \boldsymbol{B}^{(s+1)}\big), \end{aligned}$$

where analytic formulas are given in the Appendix.

Overall, our computational algorithm is summarized as follows.

Algorithm 1:

Step 1. (Initialization) Supply a good initial estimate $\hat{\boldsymbol{A}}^{(0)}$, such as $\hat{\boldsymbol{A}}^{(0)} = \boldsymbol{0}$.

Step 2. (Iteration) At iteration m, compute $\hat{\boldsymbol{A}}^{(m)}$ by solving (2.13) through our ADMM.

Step 3. (Termination) Terminate when $l(\hat{\boldsymbol{A}}^{(m-1)}) - l(\hat{\boldsymbol{A}}^{(m)}) \leq \varepsilon$, where ε is the the precision tolerance. Then the estimate $\hat{\boldsymbol{A}} = \hat{\boldsymbol{A}}^{(m^*)}$, where m^* is the smallest index at the termination criterion.

Proposition 1 *(Computational property of Algorithm 1) Algorithm 1 converges, which yields a DAG when τ is sufficiently small such that $|A_{ij}| \geq \tau$ for all edge $(i, j) \in E$,* *and is a local minimizer of* (2.10) *subject to* (2.11) *and* (2.12) *in that it satisfies a local optimality condition: For some multipliers* $\nu \geq 0$ *and* $\{\zeta_{ijk} \geq 0\}_{i,j,k=1,\ldots,p,j\neq i}$,

$$\frac{\partial l(\boldsymbol{A})}{\partial A_{ij}} + \frac{\nu}{\tau} s_{ij} + \frac{\sum_{1\leq k\leq p} \zeta_{ijk}}{\tau} s_{ij} \quad = \quad 0; \quad i,j = 1,\cdots,p, \tag{2.18}$$

where s_{ij} *is subdifferential defined as* $s_{ij} = sign(A_{ij})$ *if* $0 < |A_{ij}| < \tau$*;* $A_{ij} \in [-1,1]$ *if* $A_{ij} = 0$*;* $s_{ij} = 0$ *if* $|A_{ij}| > \tau$*;* $s_{ij} = \emptyset$ *if* $|A_{ij}| = \tau$*, is the regular subdifferential of* $J_\tau(|A_{ij}|)$ *at* A_{ij}*, and* $\emptyset$ *is the empty set. The reader may consult [39] for subdifferentials of continuous but nondifferentiable functions.*

The computation complexity for our algorithm in one iteration over six blocks in (2.18) is roughly $O(p^3 + np^2)$ given p^2 parameters in $\boldsymbol{A}$. With regard to convergence, it is usually the case that ADMM converges with modest accuracy within a few tens of iterations, although ADMM can be slow to converge with high accuracy [38]. This is in contrast to fast convergence of the DC part, which has finite termination property [40]. Based on our limited experience, our DC step converged within ten iterations for our examples.

2.4 Theory

This section develops a theory for the constrained maximum likelihood (MLE) with respect to reconstruction of a DAG's structure. We will show that the proposed method recovers the true DAG's structure under the assumption (2.19). We will proceed under the equal variance assumption, which implies that the distributions from different DAG models are identifiable [4].

To introduce notations, let $\mathcal{B} = (G, \boldsymbol{\theta})$ be a parametrized DAG, where $\boldsymbol{\theta} = (\boldsymbol{A}, \sigma)$, $G = G(\boldsymbol{A})$ is a DAG induced by parameters. Let $E = E(\boldsymbol{A}) = \{(i,j) : A_{i,j} \neq 0\}$ be the set of nonzero elements in $\boldsymbol{A}$, which is equivalent to the edge set of the graph. Let $\mathcal{F}_0 = \{\boldsymbol{\theta} = (\boldsymbol{A}, \sigma) : c_{\min}(\boldsymbol{\Omega}) \geq M_1 > 0; \sup_{1\leq j\leq p} |\Omega_{jj}| \leq M_2\}$ be the parameter space containing $(\boldsymbol{A}^0, \sigma^0)$, where $M_1, M_2 > 0$ are two constants, $\boldsymbol{\Omega} = (\boldsymbol{I} - \boldsymbol{A})^T(\boldsymbol{I} - \boldsymbol{A})/\sigma^2$ and $c_{\min}(\boldsymbol{\Omega})$ is the smallest eigenvalue of $\boldsymbol{\Omega}$. Note that the assumption that $\inf_{\boldsymbol{A}\in\mathcal{F}_0} c_{\min}(\boldsymbol{\Omega}) \geq M_1 > 0$ and $\sup_{1\leq j\leq p} |\Omega_{jj}| \leq M_2$ suffices to ensure that the likelihood function is bounded. Denote by $\mathcal{B}^0$, G^0, $\boldsymbol{\theta}^0$, $\boldsymbol{A}^0$, E^0 and $\boldsymbol{\Omega}^0$ the truth.

In what is to follow, we will derive a finite-sample probability error bound for reconstructing the true DAG by the proposed method. As a result, the proposed method not only reconstructs the true DAG when identifiable, but also recovers the optimal parameter estimation of the oracle estimator $(\hat{\boldsymbol{A}}^{OR}, \hat{\sigma}^{OR})$, defined as the maximum likelihood estimator assuming that the true edge set E^0 (or the nonzero set of $\boldsymbol{A}^0$ equivalently) is known as a priori.

Assumption A: For some positive constants M_1 and M_2, $\inf_{\boldsymbol{\Omega}} c_{\min}(\boldsymbol{\Omega}) \geq M_1 > 0$ and $\sup_{1 \leq k \leq p} |\Omega_{kk}| \leq M_2$, where $c_{\min}(\boldsymbol{\Omega})$ is the smallest eigenvalue of $\boldsymbol{\Omega}$ and Ω_{kk} is the kth diagonal of Ω.

Assumption A is a regularity condition on boundedness of entries of $\boldsymbol{\Omega}$. Assumption A implies that $\sigma^2 \geq M_2^{-1}$ since $\Omega_{jj} = \frac{1}{\sigma^2} + \sum_{k \neq j} \frac{1}{\sigma^2} A_{jk}^2$ for a DAG matrix. Moreover, from (2.4), $\sigma^2 \leq \mathrm{Var}(X_j) = \Sigma_{jj} = \boldsymbol{e}_j^T \boldsymbol{\Sigma} \boldsymbol{e}_j \leq 1/c_{\min}(\boldsymbol{\Omega}) \leq 1/M_1$, where $\boldsymbol{e}_k$ is a standard basis vector with kth element being 1. Therefore, σ^2 is bounded above and below.

Next define the degree of reconstructability for DAGs, to be

$$C_{\min}(\boldsymbol{\Omega}^0) = \inf_{\{\boldsymbol{\Omega}_E = (\boldsymbol{I} - \boldsymbol{A}_E)^T (\boldsymbol{I} - \boldsymbol{A}_E)/\sigma^2 : \boldsymbol{A}_E \neq \boldsymbol{A}^0, |E| \leq |E^0|, \boldsymbol{A}_E \text{ satisfies (2.1)}\}} \frac{-\log(1 - h^2(\boldsymbol{\Omega}_E, \boldsymbol{\Omega}^0))}{\max(|E^0 \backslash E|, 1)},$$

where $h^2(\boldsymbol{\Omega}, \boldsymbol{\Omega}^0) = 1 - \sqrt{\frac{(det(\boldsymbol{\Omega}) det(\boldsymbol{\Omega}^0))^{1/2}}{\det(\frac{\boldsymbol{\Omega} + \boldsymbol{\Omega}^0}{2})}}$ is the Hellinger-distance between $\boldsymbol{\Omega}$ and $\boldsymbol{\Omega}^0$ under (2.4). Note that under regularity conditions, a Taylor's expansion of $\log det(\boldsymbol{\Omega})$ at $\boldsymbol{A}^0$ yields that

$$\begin{aligned} &-2\log\big(1 - h^2(\boldsymbol{\Omega}_E, \boldsymbol{\Omega}^0)\big) = -\frac{1}{2}\big(\log\det\big(\boldsymbol{\Omega}_E\big) + \log\det\big(\boldsymbol{\Omega}^0\big)\big) + \log\det\Big(\frac{\boldsymbol{\Omega}_E + \boldsymbol{\Omega}^0}{2}\Big) \\ &\geq \frac{1}{8}(\vec{\boldsymbol{A}}_E - \vec{\boldsymbol{A}}^0)^T \boldsymbol{H} (\vec{\boldsymbol{A}}_E - \vec{\boldsymbol{A}}^0) \geq c^* |E^0 \setminus E| c_{\min}(\boldsymbol{H}) \gamma_{\min}^2, \end{aligned}$$

for some constant $c^* > 0$, where $\vec{\boldsymbol{A}}$ is a p^2 vector representation of $\boldsymbol{A}$, where $\gamma_{\min} \equiv \gamma_{\min}(\boldsymbol{A}^0) = \min\{|A_{jk}^0| : A_{jk}^0 \neq 0, j \neq k\}$ is the minimal of nonzero entries of $\boldsymbol{A}^0$ whose ijth element is A_{ij}, and $\boldsymbol{H} = \Big(\frac{\partial^2(-\log\det((\boldsymbol{I} - \boldsymbol{A})^T(\boldsymbol{I} - \boldsymbol{A})/\sigma^2))}{\partial^2 \boldsymbol{A}}\Big)|_{\boldsymbol{A} = \boldsymbol{A}^0}$ is the $p^2 \times p^2$ Hessian matrix of $-\log det(\boldsymbol{\Omega})$. Then $C_{\min}(\boldsymbol{\Omega}^0) \geq c^* \gamma_{\min}^2 c_{\min}(\boldsymbol{H})$, see [41] for such an expansion.

In view of the foregoing connection, the degree of reconstructability measures the overall degree of difficulty for reconstruction, which is closely related to two terms. Whereas $\gamma_{min}^2(\boldsymbol{A}^0)$ reflects the signal strength in terms of the minimal nonzero size of $\mathbf{\Omega}^0$, $c_{\min}(\boldsymbol{H})$ can be thought of as the local curvature of log-determinant of $\mathbf{\Omega}^0$, measuring dependency among entries of $\mathbf{\Omega}^0$. In a sense, both the terms are critical for reconstruction.

Our key assumption requires the degree of reconstructability exceeds a certain threshold that is proportional to $n^{-1}\max(\log p, |E^0|)$.

Assumption B: (Degree of reconstructability). Assume that

$$C_{\min}(\mathbf{\Omega}^0) \geq 4c_2^{-1}n^{-1}\max(\log p, |E^0|), \tag{2.19}$$

for some positive constant $c_2 > 0$, say $c_2 = \frac{2}{27}\frac{1}{963}$.

For reconstruction, we show that the proposed method enables us to consistently reconstruct the true DAG, yielding the optimal estimation, provided that the degree of reconstructability exceeds a certain level $n^{-1}\log p$.

The next theorem says that the oracle estimator $\hat{\mathbf{\Omega}}^{OR} = (\boldsymbol{I} - \hat{\boldsymbol{A}}^{OR})^T(\boldsymbol{I} - \hat{\boldsymbol{A}}^{OR})/(\hat{\sigma}^{OR})^2$ is constructed by a global minimizer $\hat{\boldsymbol{\theta}}^{L_0} = (\hat{\boldsymbol{A}}^{L_0}, \hat{\sigma}^{L_0})$ of (2.7) subject to (4.4) and (2.9). As $n, p, |E^0| \to \infty$, the reconstructed DAG $\hat{G}^{L_0}$ is consistent for the true DAG G^0; moreover, the optimal performance $Eh^2(\hat{\mathbf{\Omega}}^{OR}, \mathbf{\Omega}^0)$ as measured by the Hellinger-risk is recovered by $Eh^2(\hat{\mathbf{\Omega}}^{L_0}, \mathbf{\Omega}^0)$.

Theorem 2.4.1 (L_0-method) *Under Assumption A, if $K = |E^0|$ in (4.4), then there exists a constant $c_2 > 0$, say $c_2 = \frac{2}{27}\frac{1}{963}$, such that for any $(n, p, |E^0|)$,*

$$P\left(\hat{G}^{L_0} \neq G^0\right) \leq P(\hat{\mathbf{\Omega}}^{L_0} \neq \hat{\mathbf{\Omega}}^{OR}) \leq \exp\big(-c_2 n C_{\min}(\mathbf{\Omega}^0) + 2\log\left(p(p+1)+1\right) + 1) \tag{2.20}$$

Under Assumption B, $P\left(\hat{G}^{L_0} \neq G^0\right) \to 0$, and $\frac{Eh^2(\hat{\boldsymbol{\theta}}^{L_0}, \boldsymbol{\theta}^0)}{Eh^2(\hat{\mathbf{\Omega}}^{OR}, \mathbf{\Omega}^0)} \to 1$, as $n, p, |E^0| \to \infty$.

A similar result is established by the proposed estimator–the minimizer $\hat{\boldsymbol{\theta}}^T = (\hat{\boldsymbol{A}}^T, \hat{\boldsymbol{D}}^T)$ of (2.10) subject to (2.11) and (2.12) given additional Assumption C.

Assumption C: For some positive constants d_1, d_2, d_3,

$$h^2(\mathbf{\Omega}, \mathbf{\Omega}^0) \geq d_1 h^2(\mathbf{\Omega}_\tau, \mathbf{\Omega}^0) - d_3 p\tau^{d_2}, \tag{2.21}$$

where $\mathbf{\Omega}_\tau = (\boldsymbol{I} - \boldsymbol{A}_\tau)^T((\boldsymbol{I} - \boldsymbol{A}_\tau)/\sigma^2$, and the ijth element of $\boldsymbol{A}_\tau$ is defined as $A_{ij}I(|A_{ij}| \geq \tau)$.

Theorem 2.4.2 (Approximate L_0-method) *Under Assumption A, if $K = |E^0|$ and*

$\tau \leq C_{\min}(\mathbf{\Omega}^0)M_1/4p$, then there exists a constant $c_2 > 0$, say $c_2 = \frac{4}{27}\frac{1}{1926}$, such that for any $(n, |E^0|, p)$,

$$P\left(\hat{G}^T \neq G^0\right) \leq P(\hat{\mathbf{\Omega}}^T \neq \hat{\mathbf{\Omega}}^{OR}) \leq \exp\left(-c_2 n C_{min}(\mathbf{\Omega}^0) + 2\log\left(p(p-1)+1\right)+1\right).$$

Under Assumption B, $P\left(\hat{G}^T \neq G^0\right) \to 0$, $\frac{Eh^2(\hat{\Omega}^T, \mathbf{\Omega}^0)}{Eh^2(\hat{\Omega}^{OR}, \mathbf{\Omega}^0)} \to 1$, as $n, p, |E^0| \to \infty$.

The graphical structure is recovered by $\hat{\boldsymbol{\theta}}^{L_0}$ and its computational surrogate $\hat{\boldsymbol{\theta}}^T$ as well as the optimal performance of the oracle estimator.

Next we comment on technical conditions in Assumptions A and B. Assumption A is a regularity condition for $\mathbf{\Omega}$. Assumption B may be viewed as an alternative to the strong faithfulness, which is defined as follows. Given $\kappa \in (0, 1)$, a multivariate Gaussian distribution is said to be κ-strong-faithful to a DAG $G = (V, E)$ if for any $i, j \in V$ and any $S \subset V \backslash \{i, j\}$:

$$\min\{|\text{corr}(X_i, X_j|X_S)| : j \text{ not } d\text{-separated from } i, 1 \leq i, j \leq p\} > \kappa, \tag{2.22}$$

where κ is of order $\sqrt{s_0 \log p/n}$, and s_0 is some kind of sparsity measure. Note that for a pair of (i, j), the number of possible set S is $\sum_{j=1}^{p-2}\binom{p-2}{j} = 2^{p-2}$. If there is a directed edge between i and j, then j is not d-separated from i given any of these S. Therefore, for this (i, j) pair alone, there are actually exponentially many conditions to fulfill. In other words, the κ-strong-faithfulness condition excludes exponentially many sets of distributions with nonzero Lebesgue measures. Even though these sets overlap, empirical studies in [33] show that the proportion of κ-unfaithful distributions could approach 1 in some situations. This suggests (2.22) is very restrictive. As argued in the introduction, a PC algorithm may not work when $p \log p >> n$ in view of Bahadur's lower bound, suggesting that (2.22) may break down in this case. It seems that Assumption B is not subject to this restriction, although a direct connection between Assumption B and (2.22) remains unclear.

As a technical remark, we note that the proposed methodology is applicable to the situation of nonequal error variances in (2.4), although the focus of this chapter is the identifiable situation assuming equal error variances. This error variance assumption seems sensible for consistent DAG reconstruction and natural for applications with variables from a similar domain, which has been commonly used in time series models.

When the error variances are not equal in (2.4), our method continue to work with a minor modification of the likelihood function. In such a situation, DAG models are no longer identifiable, and the equivalence classes are estimated as opposed to DAGs; see [6] for a detailed discussion.

2.5 Numerical Results

This section examines operating characteristics of the proposed method, and compares it against its strong competitors via simulations in terms of estimation accuracy of directed edges and parameter estimation. Specifically, the proposed method is contrasted with three top performers. They are a test-based PC algorithm [8], a score-and-search Hill Climbing method [42] and a hybrid version Max-Min Hill Climbing (MMHC, [43]), denoted as PC, HC and MMHC, respectively. For our method, we code in C and embed into an R-package. For PC, we use the R-package pcalg. For HC and MMHC, we use the R-package bnlearn. In what follows, two simulated examples are considered in Section 5.1, in addition to one real data example in Section 5.2.

For accuracy of estimating directed edges of a graph, we consider the false positive rate (FPR) and the false discovery rate (FDR), defined as FPR = FP/(FP + TN) and FDR = FP/(TP + FP), where TP, FP, TN and FN are true positive, false positive, true negative and false negative numbers of edge estimation, respectively.

For overall accuracy of estimating the DAG structure, we employ a commonly used measure–Structural Hamming Distance (SHD). The SHD between two DAGs is the required number of edge insertions, deletions or flips to transform one graph to another graph, c.f., [43]. A smaller SHD indicates closeness of two graphs. To compute the SHD, one may consider the R-package pcalg.

For accuracy of parameter estimation of the adjacency matrix $\boldsymbol{A}$, we use the Frobenius norm loss (FL) to measure discrepancy between an estimator $\hat{\boldsymbol{A}}$ and the truth $\boldsymbol{A}^0$:

$$FL(\hat{\boldsymbol{A}}, \boldsymbol{A}^0) = \sqrt{\sum_{i=1}^{p}\sum_{j=1}^{p}(A_{ij} - A_{ij}^0)^2} \tag{2.23}$$

where $\boldsymbol{A}^0$ is the true covariance matrix.

For the proposed method, two tuning parameters (τ, K) are estimated using a tuning set. As suggested in [7], τ needs to be sufficiently small for a good approximation. In our case, τ is set to be $\{0.1, 0.01, 0.001\}$, K is an integer valued from 1 to 150, an upper bound of the maximum number of edges in the graph, controlling the degree of sparsity of a graph. Then we minimize the predicted log-likelihood in (2.5) over an independent tuning set $\boldsymbol{X}_{val}$ of size 1000 with regard to (τ, K). For the PC and the MMHC, the level of significance is set to be 0.05 [11]. The HC method does not involve a tuning parameter.

2.5.1 Simulated examples

Example 1 considers a sparse neighborhood graph requiring each node has a sparse neighborhood [11]. Note that a sparse graph does not necessarily have sparse neighborhoods. Example 2 concerns a sparse graph with non-sparse neighborhoods.

Example 1: (Sparse neighborhood) This example concerns a DAG with 100 nodes using a generation mechanism as described in [11], where a random graph is generated without any structure. First we construct the true adjacency matrix $\boldsymbol{A}$. To begin with, set $\boldsymbol{A} = \boldsymbol{0}$. Next, we generate the edge set. Given a prespecified ordering of the 100 nodes, we replace every matrix entry of $\boldsymbol{A}$ in the lower triangle, or below the diagonal, by a random sample of 0 or 1 following the Bernoulli distribution with success probability $s = 0.02$, where 1 indicates existence of an edge and s controls the degree of sparseness of a model. Then we parametrize the adjacency matrix by replacing all the entries of value 1 by 0.5, a value indicating the signal strength. Finally, given $\boldsymbol{A}$, a random sample is generated according to (2.4), where σ is set to be 1. Results are shown in Table 1.

Example 2: (Non-sparse neighborhood) This example is modified from Example 1 to generate a DAG of 100 nodes with a special structure of a "hub" node. Instead of generating the edge set from Boernoulli sampling, the only directed edges are set to be ones from the first node, the "hub" node, to the next 49 nodes. The rest 50 nodes are independent variables. Such highly connected "hub" nodes are of special interest because they are the backbones of the network architecture. Evidently, the neighborhood of the first node is not sparse. Yet, the overall graph is still sparse. The other settings remain the same as in Example 1. Results are shown in Table 2.

With regard to the accuracy of estimating the structure of a DAG, as measured

Table 2.1: Averaged false positive rate (FPR), false discover rate (FDR), Frobenius norm loss (FL), and Structural Hamming Distance (SHD), as well as their standard errors (in parenthesis), for four competing methods based on 100 simulation replications in Example 1. Here “Ours”, ”HC”, ”MMHC” and ”PC” denote ours, the HC, the MMHC and the PC methods. Note that N/A means that a method does not yield parameter estimation.

n	p	Method	FPR	FDR	FL	SHD
50	100	Ours	0.01(0.003)	0.62(0.07)	5.4(0.2)	104.37(8.8)
		HC	0.181(0.019)	0.92(0.01)	18.7(23.1)	895.2(91.7)
		mmHC	0.009(0.001)	0.49(0.06)	8.5(0.4)	94.8(8.6)
		PC	0.009(0.001)	0.46(0.04)	NA	90.5(7.1)
100	100	Ours	0.014(0.002)	0.58(0.05)	4.9(0.1)	88.75(9.5)
		HC	0.05(0.004)	0.72(0.02)	4.4(0.2)	245.1(19.0)
		mmHC	0.009(0.001)	0.36(0.03)	8.5(0.2)	62.4(6.0)
		PC	0.009(0.001)	0.33(0.03)	NA	57.5(5.4)
1000	100	Ours	0.007(0.002)	0.29(0.08)	3.2(0.5)	39.375(7.9)
		HC	0.012(0.001)	0.36(0.02)	1.6(0.2)	161.5(5.5)
		mmHC	0.004(0.001)	0.16(0.02)	1.6(0.2)	22.2(3.1)
		PC	0.010(0.001)	0.31(0.03)	NA	48.1(6.5)

Table 2.2: Averaged false positive rate (FPR), false discover rate (FDR), Frobenius norm loss (FL), and Structural Hamming Distance (SHD), as well as their standard errors (in parenthesis), for four competing methods based on 100 simulation replications in Example 2. Here “Ours”, ”HC”, ”MMHC” and ”PC” denote ours, the HC, the MMHC and the PC methods. Note that N/A means that a method does not yield parameter estimation. Here $*$ represents no return value after 24 hour running time.

n	p	Method	FPR	FDR	FL	SHD
50	100	Ours	0.002(0.003)	0.630(0.183)	3.6(0.3)	53.1(9.6)
		HC	0.200(0.022)	0.98(0.01)	37.2(91.7)	1003.0(105.5)
		mmHC	0.015(0.001)	0.94(0.02)	4.8(0.1)	117.1(5.6)
		PC	0.013(0.001)	0.96(0.02)	NA	101.2(5.9)
100	100	Ours	0.005(0.001)	0.50(0.12)	3.0(0.4)	47.5(12.3)
		HC	0.051(0.003)	0.86(0.01)	4.5(0.1)	259.6(12)
		mmHC	0.018(0.001)	0.92(0.01)	4.4(0.1)	130.2(4.8)
		PC	0.025(0.001)	0.94(0.01)	NA	157.6(5)
1000	100	Ours	0(0)	0(0)	0.2(0.02)	0(0)
		HC	0.01(0.001)	0.51(0.03)	0.7(0.1)	50.5(6.8)
		MMHC$*$	NA	NA	NA	NA
		PC $*$	NA	NA	NA	NA

by the SHD, the proposed method performs the best in Example 2, and slightly worse than the MMHC algorithm in Example 1. The HC algorithm and our method gives relatively robust results across two examples. However, the performance of the HC is not satisfactory. The poor performance of HC may be partly due to inappropriate use of the BIC for the model selection in a high-dimensional situation. while the PC and the MMHC, which relies on the PC, do not deliver robust performance across the examples. PC and MMHC perform well in Example 1 but lead to deteriorated performance in Example 2, where the non-sparse neighborhood of the "hub" node becomes the curse of the two methods. Moreover, in the case $n = 1000$ in Example 2, the PC and the MMHC fail to return a solution after 24 hour running time. In contrast, the proposed method gives consistent performance in Example 2 with the smallest FPR, PFD and SHD values and a large amount of improvement over other competing methods.

With regard to accuracy of parameter estimation, the proposed method performs best in most cases except one case with a large sample size ($n = 1000$ and $p = 100$) in Example 1, as measured by the Frobenius norm loss.

Overall, the proposed method compares favorably against top performers in the literature.

2.5.2 Oracle properties

In this simulation study, we demonstrate the theoretical result in Theorem 2 that the proposed method is able to correctly identify the oracle estimator asymptotically. There are two purposes for this. First, the simulation will demonstrate that the asymptotic property can be realized in a finite-sample situation. Second, an concordance between our and the oracle estimators indicates that our DC algorithm yields a global optimizer, because the oracle estimator is in fact the global optimizer of our nonconvex problem in certain sense. For comparison, HC, MMHC and PC are included. Still, the PC algorithm does not yield parameter estimation.

As in the setting of Example 1, we randomly generate a DAG with $p = 10, 20$ nodes, with the number of edges equal to the number of nodes. As suggested by Table 3, the proposed method has a good agreement rate of 54% and 84% for $n = 1000$ and $p = 10, 20$. This says that it has a good probability of 54% and 84% to yield a global optimizer in these cases. This aspect of a DC algorithm agrees with the finding of [30] for a different problem.

Table 2.3: Frobenius norm loss (FL), and oracle rate (OR) for three competing methods based on 100 simulation replications in Example 3. Here “Ours”, ”HC”, and ”MMHC” denote ours, the HC and the MMHC.

n	p	Method	FL	OR
1000	10	Ours	0.2(0.1)	84%
		HC	0.1(0.1)	66%
		mmHC	0.1(0.1)	79%
		PC	NA	0%
1000	20	Our	0.5(0.4)	54%
		HC	0.2(0.1)	15%
		mmHC	0.2(0.1)	50%
		PC	NA	0%

2.5.3 An example demonstrating consistency

Similar to the setting of Example 1, we randomly generate a DAG with $p = 10$ nodes, as depicted in Figure 2.1. In this case, we study accuracy of reconstructing a DAG’s structure of the proposed method, HC and MMHC as a function of the sample size n while $p = 10$ is held fixed. Note that the DAG is fully identifiable. Furthermore, PC is not considered because it only gives partially directed graphs. The results are displayed in Figure 2.2 for $n = 20, 100, 500$.

As n increases from $n = 20$ to $n = 500$, the proposed method continues to improve until identifying all directions correctly. By comparison, HC and the MMHC recover the true skeleton but miss several directions, which remains missed as n increases.

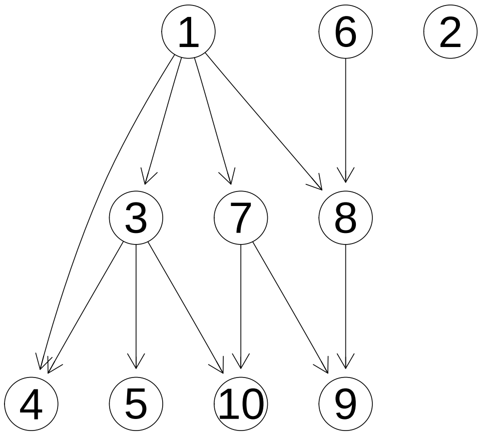


Figure 2.1: DAG representation of the true network in Section 5.2.

2.5.4 Analysis of cell signaling data

This section applies the proposed method to analyze multivariate flow cytometry data in [44]. Data were collected after a series of stimulatory cues and inhibitory interventions, with cell reactions terminated 15 minutes after stimulation by fixation, to profile the effects of each condition on the intracellular signaling networks of human primary naive CD4+ T cells. 7466 flow cytometry measurements were made over eleven phosphorylated proteins and phospholipids from nine experimental conditions. The primary goal of this experiment is to infer causal influences in cellular signaling networks through perturbations with molecular interventions. This requires a multivariate in lieu of univariate approach, which is possible given the simultaneous measurements. Note that the data is interventional from nine experiments whereas our method and the other three are designed for observational data. Therefore, we centered data from each experiment separately before combining them to remove intervention effects on means so that the data becomes more observational.

This is a well-studied example and we use the representation in Figure 3.1 from [44] as a benchmark. A direction from one node to another is interpreted as a directional influence between the corresponding two proteins. The reader may consult [44] for details about the data.

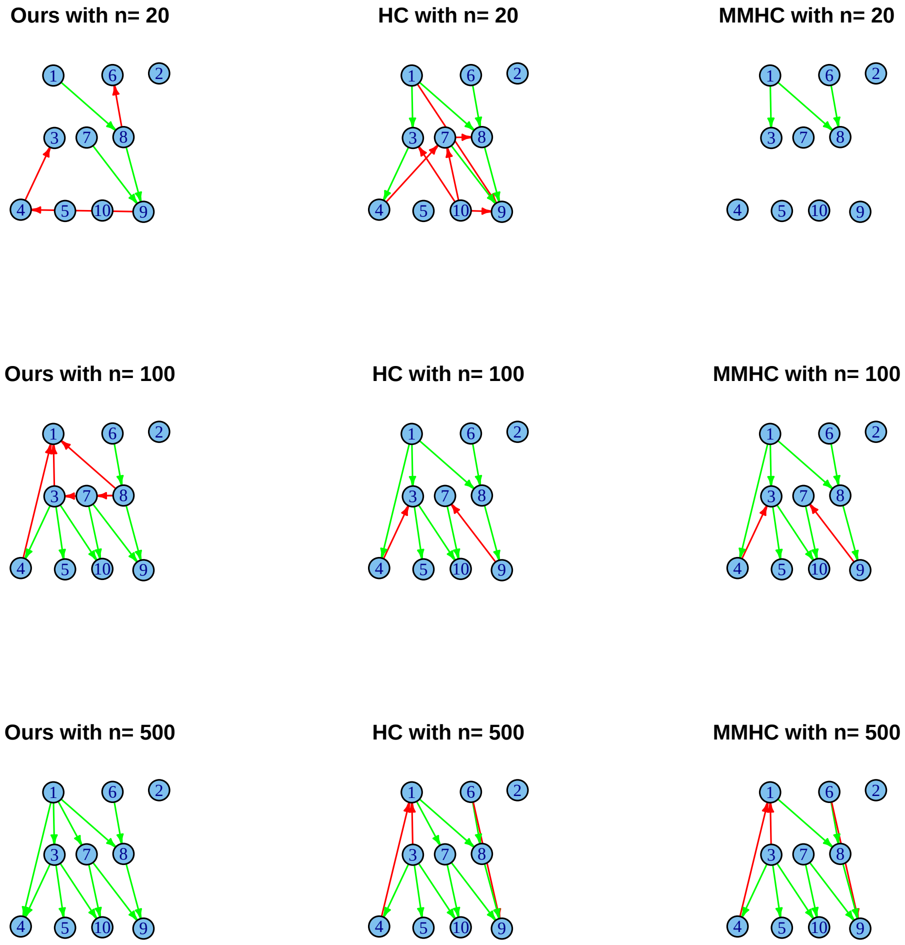


Figure 2.2: Reconstructed networks by various methods. True and false discoveries are marked with green and red lines, where wrong directions are considered to be false in this case.

We fit (2.4) with one tenth samples for training and nine tenths for tuning. The learned networks are shown in Figure 2.4. An edge is marked in green if it matches one in Figure 3.1; otherwise it is red, including wrong orientation cases. If an edge does not even match the skeleton in Figure 3.1, then it is marked in dashes. All four methods give similar results due to the large sample size relative to the number of nodes. The proposed method identifies nine true edges, one wrong edge. All the other three methods have more false discoveries. In addition, the PC has fewer correct directions. Unfortunately, however, all methods miss several known edges. One possible reason is that many directional relations in protein-signaling networks may behave nonlinearly, which can not be captured by a linear model. Analysis based on discretization data [44, 19] tends to capture such nonlinearity and thus more true edges. Another possible reason is that we removed, before the analysis, intervention effects, which may be crucial in identifying directional relations. Incorporating interventional data is one future direction to extend our method.

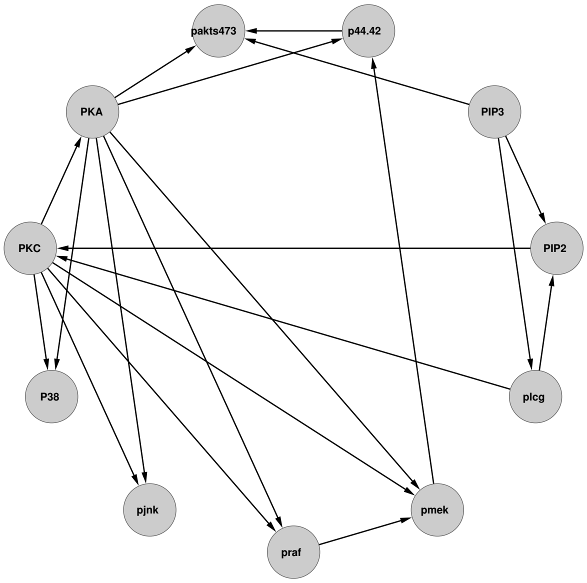
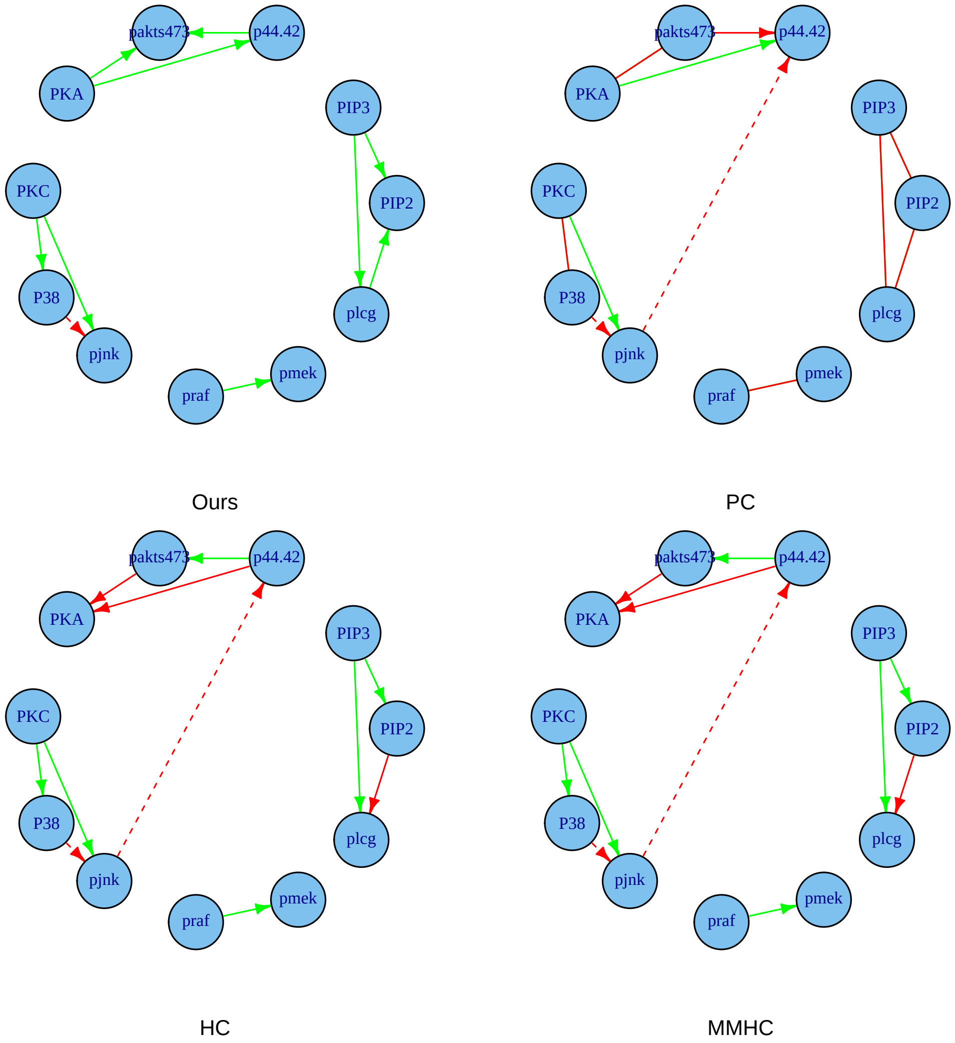


Figure 2.3: Display of a consensus protein network consisting of eleven proteins.

2.6 Discussion

This article proposes a method for reconstructing the graphical structure of a directed acyclic graph. The method identifies the true DAG by estimating the adjacency matrix of the graph using a constrained likelihood maximization incorporating acyclicity of a DAG through constraints.



Ours

PC

HC

MMHC

Figure 2.4: Reconstructed networks using the proposed method and the other three methods. Correct discoveries are marked in green, whereas false discoveries are displayed in red. The network constructed by the PC is partially directed.

For reconstruction of directional relations, we introduce a method that is dramatically different from conventional methods to overcome two major difficulties. The first concerns super-exponentially many constraints, which is addressed by a novel constraint reduction method reducing to cubic in the number of nodes. The second is identifiability of directional relations. The proposed method enables us to reconstruct all the directional relations when possible through the acyclicity constraints.

The proposed method, particularly the idea of identifying active constraints from super-exponentially many constraints, may be useful and generalized to other problems, especially for nonlinear models. Further investigation is necessary.

Chapter 3

Maximum likelihood estimation of multiple directed acyclic graphs

In this chapter, we go beyond the single DAG reconstruction discussed in Chapter 2 and propose a constrained likelihood approach to learning multiple DAGs from inhomogeneous data. Another deviation from Chapter 2 is that we assume the ordering of nodes is known. The rest of this chapter is organized as follows.Section Section 3.1 introduce the background of DAG learning when the ordering is known. 3.2 introduces the methodology, followed by our computational development in 3.3. Operating characteristics of the proposed method are examined on simulated and real data in Sections 3.4 and3.5, respectively.

3.1 Background

Reconstruction of a DAG given a partial ordering is equivalent to sparse estimation of Cholesky decomposition of a covariance matrix, and thus is computationally feasible [22]. The ordering information is usually determined by a natural ordering of temporal observations, previous experiments and prior knowledge. In this article, we reconstruct multiple DAGs given a known partial ordering. To our knowledge, estimation of structural changes over multiple DAGs has not been yet explored, although that over multiple undirected graphs has been studied in [45, 46, 47]. In practice, identification of a change in causality structure arises from detecting a change corresponding to that of experimental conditions or responding to a certain event or treatment.

This chapter focuses on maximum likelihood estimation of multiple DAGs under a structural equation model. It is known that maximum likelihood estimation breaks down when the number of variables exceeds the sample size. Even for a moderately sized problem, it always yields a complete graph and does not estimate graphical structures well. Therefore, different methods using penalization have been proposed for sparse estimation of graphical models [48, 49, 36, 22]. To achieve our goal of learning graphical structures, we construct two nonconvex constraints based on the truncated L_1-function (TLP, [7]), as a computational surrogate of the L_0-function, with one constraint imposing sparseness and the other encouraging a common structure. Computationally, with difference convex programming and augmented Lagrange multipliers, nonconvex minimization is solved through a sequence of convex subproblems iteratively. For each subproblem, we develop a fast algorithm to treat a constrained L_1-problem, which we call pairwise coordinate descent algorithm.

3.2 Statistical methodology

Given L p-dimensional vectors of random variables $\boldsymbol{X}^{(l)} = (X_1^{(l)}, \dots, X_p^{(l)})^T$ with a known partial ordering, one from each population, we use L DAGs to describe causal relations within each population and to explore differences among the populations. That is, each component $X_i^{(l)}$ corresponds to one node in the lth DAG, with a directed edge between two nodes indicating a causal relation between them. Without loss of generality,we assume that $\boldsymbol{X}^{(l)}$ has been sorted according to its partial order, which means a causal relation is only possible from $X_i^{(l)}$ to $X_j^{(l)}$ for $i < j$.

To model causality among the components of $\boldsymbol{X}^{(l)}$, consider a structural equation model of the form

$$\boldsymbol{X}^{(l)} = \boldsymbol{A}^{(l)}\boldsymbol{X}^{(l)} + \boldsymbol{\epsilon}^{(l)}, \quad l = 1, \cdots, L, \tag{3.1}$$

where $\boldsymbol{A}^{(l)}$ is an adjacency matrix in which a nonzero jk-th element of $\boldsymbol{A}^{(l)}$ corresponds to a directed edge from parent node k to child note j with its value $A_{jk}^{(l)}$ indicating the strength of the relationship, and $\boldsymbol{\epsilon}^{(l)} = (\epsilon_1^{(l)}, \cdots, \epsilon_p^{(l)})^T$ is an independent latent variable vector representing unexplained variations in the nodes and acting as random noises. Note that $A_{i,j}^{(l)} = 0$ for $i < j$, since $\boldsymbol{X}^{(l)}$'s are assumed to be ordered. In addition, $A_{ii}^{(l)} = 0$, for all i, as a self-loop is not allowed in a DAG. Therefore, the adjacency matrices $\boldsymbol{A}^{(l)}$, $l = 1, \ldots, L$, are lower triangular with zero diagonal elements, that is, $A_{i,j}^{(l)} = 0,\ j \geq i$. The model basically says that each $\boldsymbol{X}_j^{(l)}$ depends linearly on its parent variables and some latent variable $\epsilon_j^{(l)}$. Here we assume that $\epsilon_1^{(l)}, \cdots, \epsilon_p^{(l)}$ follow independent normal distributions, that is, $\epsilon_j^{(l)} \overset{ind}{\sim} N(0, (\sigma_j^{(l)})^2)$. This implies that $\boldsymbol{X}^{(l)}$'s follow multivariate normal distributions. Note that (3.1) becomes Gaussian autoregressive model when the subscript of $X_i^{(l)}$ denotes the consecutive time. Our likelihood method is readily generalizable to other distributions. The reader may consult [50] for (3.1) and structural equations.

In (3.1), nonzero entries of $\boldsymbol{A}^{(l)}$ are uniquely specified by the lth DAG. Thus, we estimate $(\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(L)})$ to preserve a common structure and identify differences among them.

For a total of $n = \sum_{l=1}^{L} n_l$ random samples, n_l samples are drawn according to (3.1) for each l to form an $n \times p$ data matrix $\mathbf{X}^{(l)} = (\boldsymbol{x}_1^{(l)}, \ldots, \boldsymbol{x}_p^{(l)})$, where each $\boldsymbol{x}_j^{(l)} = (x_{1,j}^{(l)}, \ldots, x_{n_l,j}^{(l)})^T$ is an n_l-dimensional column vector for each node, and samples from different populations $\mathbf{X}^{(l)}$ are assumed to be independent. Note that an arbitrary mean vector can be incorporated by adding an intercept to (3.1).

Let $k^- = \{j = 1, \ldots, k-1\}$ be a set of indices, with $k = 1$ indicating the null set. Let $\mathbf{X}_{j^-}^{(l)} = (\boldsymbol{x}_1^{(l)}, \ldots, \boldsymbol{x}_{j-1}^{(l)})$, $\boldsymbol{A}_{j,j^-}^{(l)} = \left(A_{j,1}^{(l)}, \ldots, A_{j,j-1}^{(l)}\right)$. The likelihood of $\mathbf{X}^{(l)}$ can be written as

$$f\left(\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(L)}\right) = \prod_{l=1}^{L} f\left(\mathbf{X}^{(l)}\right) = \prod_{l=1}^{L}\prod_{j=1}^{p} f\left(\boldsymbol{x}_j^{(l)} | \mathbf{X}_{j^-}^{(l)}\right) = \prod_{j=1}^{p}\prod_{l=1}^{L} f\left(\boldsymbol{x}_j^{(l)} | \mathbf{X}_{j^-}^{(l)}\right). \quad (3.2)$$

This yields the negative log-likelihood

$$-\log f\left(\mathbf{X}^{(1)},\ldots,\mathbf{X}^{(L)}\right)=\sum_{j=1}^{p}\left(\sum_{l=1}^{L}\left(-\log f\left(\boldsymbol{x}_j^{(l)}|\mathbf{X}_{j^-}^{(l)}\right)\right)\right). \tag{3.3}$$

Using the fact that $X_j^{(l)}|\boldsymbol{X}_{j^-}^{(l)}$ follows $N\left(\boldsymbol{A}_{j,j^-}^{(l)}\boldsymbol{X}_{j^-}^{(l)},(\sigma_j^{(l)})^2\right)$ from (3.1), we obtain that

$$-\log f\left(\mathbf{X}^{(1)},\ldots,\mathbf{X}^{(L)}\right)=\sum_{j=1}^{p}\left(\sum_{l=1}^{L}\left(\frac{1}{2(\sigma_j^{(l)})^2}\left\|\boldsymbol{x}_j^{(l)}-\mathbf{X}_{j^-}^{(l)}\left(\boldsymbol{A}_{j,j^-}^{(l)}\right)^T\right\|^2+n_j\log\sigma_j^{(l)}\right)\right). \tag{3.4}$$

Maximizing (3.2), equivalently minimizing (3.4), may result in over-fitting and lead to fully connected DAGs, especially when the number of unknown parameters exceeds the sample size. We therefore regularize (3.3) through nonconvex constraints to pursue sparsity and detect structural changes. Note that the constrained approach is not equivalent to its penalized regularization counterpart because of nonconvexity in this case.

Our method is to regularize the number of nonzeros of the adjacency matrices as well as the number of pairwise differences between the corresponding entries across adjacency matrices. Let $\mathscr{L}$ be a set of index pairs in which a pair (l,s) indicates the possibility that $\boldsymbol{A}^{(l)}$ and $\boldsymbol{A}^{(s)}$ share some common entries and can be grouped or collapsed if data suggest so. Constraints are used to regularize, which are in the form:

$$\sum_{l=1}^{L}\|\boldsymbol{A}^{(l)}\|_0\le t_1,\qquad \sum_{(l,s)\in\mathscr{L}}\|\boldsymbol{A}^{(l)}-\boldsymbol{A}^{(s)}\|_0\le t_2. \tag{3.5}$$

where $\|\boldsymbol{A}\|_0:=\sum_{i=1}^{p}\sum_{j=1}^{p}I(|A_{i,j}|\neq 0)$, is the L_0-norm of $\boldsymbol{A}$, or the number of nonzero entries of $\boldsymbol{A}$, $t_1\ge 0$ and $t_2\ge 0$ are tuning parameters corresponding to the number of nonzeros of the adjacency matrices and the number of element-wise differences with respect to $\mathscr{L}$. A complete set $\mathscr{L}=\{(l,s):1\le l<s\le L\}$ is used unless additional information is available. For example, a temporal set $\mathscr{L}=\{(l,l+1):1\le l\le L-1\}$ is used in dynamic networks with l representing consecutive times.

To allow for different degrees of sparsity over different rows of lower triangular matrices $\boldsymbol{A}^{(l)}$, we replace (3.5) by p pairs of row-wise sparsity through $2p$ different tuning parameters $\{t_{1,j},t_{2,j}\}$:

$$\sum_{l=1}^{L}\sum_{k=1}^{j-1}\left\|A_{j,k}^{(l)}\right\|_0 \leq t_{1,j}, \quad \sum_{(l,s)\in\mathscr{L}}\sum_{k=1}^{j-1}\left\|A_{j,k}^{(l)}-A_{j,k}^{(s)}\right\|_0 \leq t_{2,j}, \quad j=1,\cdots,p. \tag{3.6}$$

This is computationally feasible because the log-likelihood (3.3) is separable or decomposable in j. In fact, minimizing (3.3) subject to (3.6) reduces to p subproblems.

To circumvent computational difficulty of minimizing (3.3) subject to (3.6), we approximate the L_0 funtion there by a surrogate function, the truncated L_1 function (TLP, [7]), defined as $\mathrm{P}_\lambda(x) = \min\left(\frac{|x|}{\lambda}, 1\right)$. As λ tends to 0, the TLP recovers the L_0-function exactly. Now the constraints in (3.6) become

$$\sum_{l-1}^{L}\sum_{k=1}^{j-1}\mathrm{P}_\lambda(A_{j,k}^{(l)}) \leq t_{1,j}, \quad \sum_{(l,s)\in\mathscr{L}}\sum_{k=1}^{j-1}\mathrm{P}_\lambda(A_{j,k}^{(l)}-A_{j,k}^{(s)}) \leq t_{2,j}, \quad j=1,\cdots,p. \tag{3.7}$$

To simplify tuning, we introduce a single constraint for each row as opposed to the two constraints in (3.7), with new tuning parameters (κ_j, t_j) corresponding to $(t_{1,j}, t_{2,j})$,

$$\sum_{l=1}^{L}\sum_{k=1}^{j-1}\mathrm{P}_\lambda(A_{j,k}^{(l)}) + \kappa_j\sum_{(l,s)\in\mathscr{L}}\sum_{k=1}^{j-1}\mathrm{P}_\lambda(A_{j,k}^{(l)}-A_{j,k}^{(s)}) \leq t_j \quad j=1,\ldots,p, \tag{3.8}$$

where κ_j seeks a trade-off between sparsity and grouping.

Based on the foregoing discussion, we solve (3.3) subject to (3.8) by solving its equivalent form through p subproblems:

$$\begin{aligned} &\min\sum_{l=1}^{L}\left(\frac{1}{2(\sigma_j^{(l)})^2}\left\|\boldsymbol{x}_j^{(l)}-\mathbf{X}_{j^-}^{(l)}\left(\boldsymbol{A}_{j,j^-}^{(l)}\right)^T\right\|^2 + n_l\log\sigma_j^{(l)}\right), \text{ subject to} \\ &\sum_{l=1}^{L}\sum_{k=1}^{j-1}\mathrm{P}_\lambda(A_{j,k}^{(l)}) + \kappa_j\sum_{(l,s)\in\mathscr{L}}\sum_{k=1}^{j-1}\mathrm{P}_\lambda(A_{j,k}^{(l)}-A_{j,k}^{(s)}) \leq t_j, \quad j=1,\ldots,p. \end{aligned} \tag{3.9}$$

These p subproblems are of the same type, hence we only need to consider a general form. Let $\mathbf{Y}^{(l)}$ be a vector of length n_l, corresponding to $\boldsymbol{x}_j^{(l)}$ in (3.9), $\mathbf{X}^{(l)}$ be an n_l by m matrix, corresponding to $\mathbf{X}_{j^-}^{(l)}$ with $m = j-1$, and $\boldsymbol{\beta}^{(l)}$ be m-dimensional vector corresponding to $\boldsymbol{A}_{j,j^-}^{(l)}$. Then, a general form is,

$$\min \sum_{l=1}^{L} \left(\frac{1}{2\sigma_l^2} \|\mathbf{Y}^{(l)} - \mathbf{X}^{(l)}\boldsymbol{\beta}^{(l)}\|^2 + n_l \log \sigma_l \right), \text{ subject to}$$
$$\sum_{l=1}^{L}\sum_{j=1}^{m} \min\left(\frac{|\beta_j^{(l)}|}{\lambda}, 1\right) + \kappa \sum_{(l,s)\in\mathscr{L}} \sum_{j=1}^{m} \min\left(\frac{|\alpha_j^{(ls)}|}{\lambda}, 1\right) \le t. \tag{3.10}$$

where $\alpha_j^{(ls)} = \beta_j^{(l)} - \beta_j^{(s)}$, and $\zeta = (\beta_j^{(l)}{}_{l=1,\ldots,L}^{j=1,\ldots,m}, \alpha_j^{(ls)}{}_{j=1,\ldots,m}^{(l,s)\in\mathscr{L}})$ is our new set of variables to be optimized. In addition, a new constraint $T\zeta = 0$ is imposed, namely, $\beta_j^{(l)} - \beta_j^{(s)} - \alpha_j^{(ls)} = 0$, for $j = 1, \ldots, m,\ (l, s) \in \mathscr{L}$.

3.3 Computation

This section develops a computational method for nonconvex minimization (3.10) through difference convex programming, augmented Lagrange multipliers and our pairwise coordinate descent algorithm.

3.3.1 Difference convex programming

For minimization in (3.10), we employ difference convex (DC) programming, which leads to a finite-step termination due to piecewise linearity of the TLP function [7]. Here P_λ can be decomposed into a difference of two convex functions:

$$\mathrm{P}_\lambda(x) = \min\left(\frac{|x|}{\lambda}, 1\right) = \frac{|x|}{\lambda} - \max\left(\frac{|x|}{\lambda} - 1, 0\right). \tag{3.11}$$

This in turn yields a decomposition of the left-hand side of (3.10) into a difference of two convex part, that is,

$$S_1(\zeta) - S_2(\zeta) \le t,$$

where

$$S_1(\zeta) = \sum_{l=1}^{L}\sum_{j=1}^{m} \frac{|\beta_j^{(l)}|}{\lambda} + \kappa \sum_{(l,s)\in\mathscr{L}} \sum_{j=1}^{m} \frac{|\alpha_j^{(ls)}|}{\lambda}, \tag{3.12}$$

$$S_2(\zeta) = \sum_{l=1}^{L}\sum_{j=1}^{m} \max\left(\frac{|\beta_j^{(l)}|}{\lambda} - 1, 0\right) + \kappa \sum_{(l,s)\in\mathscr{L}} \sum_{j=1}^{m} \max\left(\frac{|\alpha_j^{(ls)}|}{\lambda} - 1, 0\right). \tag{3.13}$$

This constructs a sequence of convex approximation sets that are contained in the original nonconvex set iteratively by replacing S_2 at iteration k with its affine majorization at iteration $k-1$. At iteration k we minimize (3.10) subject to $T\zeta = 0$ and a relaxed constraint

$$\begin{aligned}&\sum_{l=1}^{L}\sum_{j=1}^{m}|\beta_j^{(l)}|I(|\hat{\beta}_j^{(l)[k-1]}| \le \lambda) + \kappa \sum_{(l,s)\in\mathscr{L}}\sum_{j=1}^{m}|\alpha_j^{(ls)}|I(|\hat{\alpha}_j^{(ls)[k-1]}| \le \lambda)\\ \le&\lambda\left(t - \sum_{l=1}^{L}\sum_{j=1}^{m} I(|\hat{\beta}_j^{(l)[k-1]}| > \lambda) - \kappa \sum_{(l,s)\in\mathscr{L}}\sum_{j=1}^{m} I(|\hat{\alpha}_j^{(ls)[k-1]}| > \lambda)\right), \qquad (3.14)\end{aligned}$$

where $\hat{\beta}_j^{(l)[k-1]}$ is the estimate of $\beta_j^{(l)}$ at the $(k-1)$th iteration, and $\hat{\alpha}_j^{(ls)[k-1]}$ is the estimate of $\alpha_j^{(ls)}$ at the $(k-1)$th iteration.

3.3.2 Augmented Lagrange multipliers

The constraint $T\zeta = 0$ defined by slack variables $\alpha_j^{(ls)}$ is treated through the augmented Lagrange multipliers, which is designed to convert a constrained problem to an unconstrained one. At iteration w, we minimize $S(\zeta)$ subject to (3.14)

$$\begin{aligned}S(\zeta) =& \sum_{l=1}^{L}\left(\frac{1}{2\sigma^{(l)2}}\|\mathbf{Y}^{(l)} - \mathbf{X}^{(l)}\boldsymbol{\beta}^{(l)}\|^2 + n_l \log \sigma^{(l)}\right) + \sum_{(l,s)\in\mathscr{L}}\sum_{j=1}^{m}\tau_j^{(ls)[w]}(\beta_j^{(l)} - \beta_j^{(s)} - \alpha_j^{(ls)})\\ &+ \frac{1}{2}\sum_{(l,s)\in\mathscr{L}}\sum_{j=1}^{m}\nu_j^{(ls)[w]}(\beta_j^{(l)} - \beta_j^{(s)} - \alpha_j^{(ls)})^2, \qquad (3.15)\end{aligned}$$

where $\tau_j^{(ls)[w]}$ are Lagrangian multipliers for $T\zeta = 0$ and $\nu_j^{(ls)[w]}$ control the convergence speed of the algorithm.They are updated until convergence:

$$\tau_j^{(ls)[w+1]} = \tau_j^{(ls)[w]} + \mu_j^{(ls)[w]}(\beta_j^{(l)} - \beta_j^{(s)} - \alpha_j^{(ls)}), \quad \mu_j^{(ls)[w+1]} = \rho\mu_j^{(ls)[w]},$$

where $\rho \in (1, 2)$ is pre-determined. At iteration k in the DC loop and at iteration w in the augmentation loop, we minimize (3.15) subject to (3.14). This weighted Lasso problem is solved by a pairwise coordinate descent algorithm to be introduced next.

3.3.3 Pairwise coordinate descent

A Lasso problem [51] solves

$$\min_{\beta_1,\ldots,\beta_m} f(\boldsymbol{\beta}), \text{ subject to } \sum_{j=1}^{m} |\beta_j| \leqslant t, \tag{3.16}$$

where f is a convex cost function.

For (3.16), coordinate descent methods are applicable to its regularization version [52, 53]. However, such a method breaks down for (3.16), as its solution may be trapped [52]. Here we develop a directional blockwise coordinate descent method, to solve (3.15) subject to (3.14). The main idea is to seek an optimal solution only over the simplex boundary, where the solution lies. This directional search strategy overcomes the difficulty for an optimal solution to be trapped at the constraint boundary.

A solution of (3.16) exists when f is continuous since a feasible region of $\boldsymbol{\beta}$ is compact. For sparse learning, the Lasso constraint is active only when the feasible region defined by the constraint excludes all global minima of $f(\boldsymbol{\beta})$, in which any solution of (3.16) lies on its boundary. Instead of searching coordinatewisely, we move along directions $\Delta|\beta_i| = -\Delta|\beta_j|$, to search over the boundary, where $\Delta|\beta_i|$ is the change in the absolute value of β_i after a step. Note that the pair of (i, j) is chosen so that the pair gives the steepest descent in the cost function value among all pairs.

The algorithm is summarized as follows.

Algorithm 1 Pairwise coordinate descent

Step 1. (Initialization) Input a initial value β^0 for β satisfying $||\beta_0||_1 = t$.

Step 2. (Iteration) Find the steeping direction satisfying $\Delta|\beta_i| = -\Delta|\beta_j|$. Perform an exact line search to determine the best step length and update.

Step 3. (Stopping rule) Terminate when the following subdifferential condition is satisfied: there exists $\lambda > 0$, s.t. $-\text{sign}(\beta_j)\frac{\partial f(\beta)}{\partial \beta_j} = \lambda$ for $\beta_j \neq 0$, and $\left|\frac{\partial f(\beta)}{\partial \beta_j}\right| < \lambda$ for $\beta_j = 0$.

Convergence of the algorithm is assured by Theorem 1.

Theorem 3.3.1 *For (3.16), stationary points and minimum points coincide. If the minimizer is unique, the algorithm converges to it.*

If the cost function has nuisance parameters in addition to $\boldsymbol{\beta}$, for example, $\sigma^{(l)}$, then we need to treat them as unconstrained coordinates and update them coordinately in every iteration.

A DC loop and an augmentation loop converge in only a few steps in practice [7]. Taking advantage of Algorithm 1 in the inner loop, our method is capable of treating multiple graphs of over thousands of variables in real time. This is desirable for our nonconvex minimization problem.

3.4 Simulations

This section examines operating characteristics of the proposed method, and contrasts it against its competitors through simulations. In particular, the proposed method, denoted by "nonconvex", is examined together with its convex counterpart—our method with the L_1-function replacing the TLP, denoted by "convex", and a sparse L_1 method from [22] for DAGs individually, denoted by "DAGlasso".

In simulations, two DAGs are considered and a complete set $\mathscr{L} = \{(1,2)\}$ is used for possible grouping. All simulations are performed in R. Performance metrics for sparsity and grouping pursuit are the number of false positives (FP), the number of false negatives (FN), the number of correctly identified differences between graphs (TD), and the number of falsely identified differences between graphs (FD). Overall, we use the Matthews Correlation Coefficient (MCC) [54] as a performance metric, which is commonly used in binary classification, and is defined as,

$$\mathrm{MCC} = \frac{(\mathrm{TP} \times \mathrm{TN}) - (\mathrm{FP} \times \mathrm{FN})}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}} \tag{3.17}$$

where TP, TN, FP and FN correspond to true positives, true negatives, false positives and false negatives, respectively. A larger value of MCC gives better a fit with 1 being the best and -1 being the worst.

Two graphs are generated as follows. The first DAG $\mathscr{G}_1$ is generated at random, with the number of nodes $p = 50, 100, 200$, having the average probability of connecting one node to another with higher ordering: 0.02. The second DAG $\mathscr{G}_2$ is obtained by removing a number of edges from $\mathscr{G}_1$ and this number is controlled to be less than 1%

of the total edges in $\mathscr{G}_1$. For $p = 50$, the two DAGs are just identical. For $p = 100, 200$, removal is done by deleting all edges connecting from a specific node, which mimics a situation when a certain gene from $\mathscr{G}_1$ becomes isolated due to some treatment effect. Finally, we generate a random sample from $\mathscr{G}_1$ and $\mathscr{G}_2$ respectively with equal sample size $n_1 = n_2 = n$.

For the proposed method, there are three data-dependent tuning parameters (λ, κ, t) in (3.10). It has been shown in [7] that estimation based on the TLP is not sensitive to the choice of λ as long as λ is small enough. A common set of values for λ is $\{0.1, 0.01, 0.001\}$, and κ is a tuning parameter ranged between 0 and 1, with three to five values, based on our limited numerical experience. This makes tuning three parameters much easier with an effort focused on tuning t. In this simulation study, tuning parameters for all methods are optimized for prediction over an independent data set of size 1000 for each graph.

Table 3.1: Estimated quantities and their corresponding estimated standard errors (in parentheses) based on 100 simulation replications. For $p = 50$, there are 20 edges in $\mathscr{G}_1$ and 20 edges in $\mathscr{G}_2$ with no differences. For $p = 100$, there are 88 edges in $\mathscr{G}_1$ and 86 edges in $\mathscr{G}_2$ with two differences. For $p = 200$, there are 427 edges in $\mathscr{G}_1$ and 422 edges in $\mathscr{G}_2$ with 5 differences.

n	p	method	FP	FN	TD	FD	MCC
50	50	DAGlasso	15.7(3.9)	0.2(0.4)	0(0)	15.3(3.8)	0.84(0.03)
		convex	3.9(2.7)	0(0)	0(0)	0.3(0.5)	0.95(0.03)
		nonconvex	0.3(0.8)	0.1(0.3)	0(0)	0(0.2)	0.99(0.01)
	100	DAGlasso	73.7(8.6)	1.1(1)	2(0)	72.3(8.2)	0.83(0.02)
		convex	18.9(6.2)	0.4(0.9)	1.7(0.5)	0.9(1)	0.95(0.02)
		nonconvex	4.7(3.5)	1.6(2)	1.6(0.6)	0.2(0.5)	0.98(0.02)
	200	DAGlasso	406.5(21.7)	14.5(3.8)	4.9(0.3)	402.1(20.8)	0.80(0.01)
		convex	122.3(15.3)	46.7(9.2)	2.5(1)	1.4(1.2)	0.90(0.01)
		nonconvex	30.1(11.2)	32.9(9.5)	3.2(1.2)	0.6(0.9)	0.96(0.01)
100	50	DAGlasso	6.7(2.6)	0(0)	0(0)	6.6(2.6)	0.92(0.03)
		convex	1.2(1.5)	0(0)	0(0)	0.1(0.3)	0.98(0.02)
		nonconvex	0(0.3)	0(0)	0(0)	0(0.2)	1.00(0.01)
	100	DAGlasso	32.3(5.9)	0(0.1)	2(0)	31.8(6)	0.92(0.01)
		convex	6.3(3.2)	0(0)	1.9(0.3)	0.5(0.7)	0.98(0.01)
		nonconvex	1(1.3)	0(0)	2(0.1)	0.1(0.3)	1.00(0.01)
	200	DAGlasso	162.7(14.0)	0.1(0.3)	5(0)	158.2(13.7)	0.91(0.01)
		convex	28.9(7.1)	0.1(0.4)	4.5(0.7)	1(0.9)	0.98(0.01)
		nonconvex	3.7(3.2)	1(1.8)	4.8(0.4)	0.3(0.7)	1.00(0.01)

As suggested in Table 1, the proposed method compares favorably against its competitors across all the situations, in terms of the Matthews Correlation Coefficients, accuracy of estimation, and detection of a structural change. Interestingly, seeking common structures or detection of structural changes is more critical for multiple graphs than a single graph, especially when they share some common structures. In addition, our nonconvex method improves significantly over its convex counterpart. In most cases, our nonconvex method yields a MMC value close to 1, suggesting almost perfect learning of structures. In this sense, a nonconvex method is useful in estimating directed graphs.

3.5 Analysis of *cell signaling* data

This section applies the proposed method to analyze multivariate flow cytometry data in [44]. Data were collected after a series of stimulatory cues and inhibitory interventions, with cell reactions stopped at 15 minutes after stimulation by fixation, to profile the effects of each condition on the intracellular signaling networks of human primary naive CD4+ T cells. Over 10,000 flow cytometry measurements were made over 11 phosphorylated proteins and phospholipids from 14 experimental conditions. The main purpose of this experiment is to infer casual influences in cellular signaling networks through perturbations with molecular interventions. The simultaneous measurements permits multivariate as opposed to univariate approaches. Data sets were available; see [44] for more details. The DAG representation of the network is displayed in Figure 3.1 [44]. A direction from node $\boldsymbol{X}$ to node $\boldsymbol{Y}$ is interpreted as a causal influence from $\boldsymbol{X}$ to $\boldsymbol{Y}$.

A DAG model was fit in [44] with data from the first nine conditions, following called Group 1, whereas the rest five conditions, denoted as Group 2 were not used for estimation. Note that all five conditions in Group 2 employed ICAM-2, a general perturbation, in addition to perturbations used in Group 1. In our analysis, we are interested in whether the usage of ICAM-2 in Group 2 activate or inhibit some causal relationship in the network. Thus, data have been split into two datasets: 7466 samples from Group 1 and 4206 samples from Group 2. We fit a two-DAG model with one tenth samples for training and nine tenth for tuning. The graphical result of reconstructed

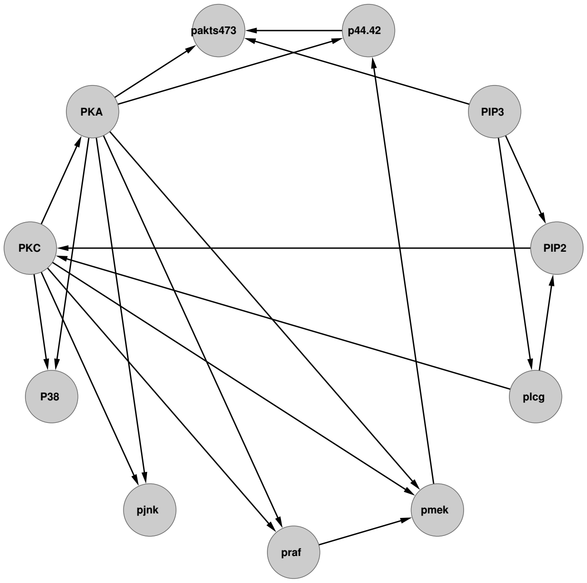
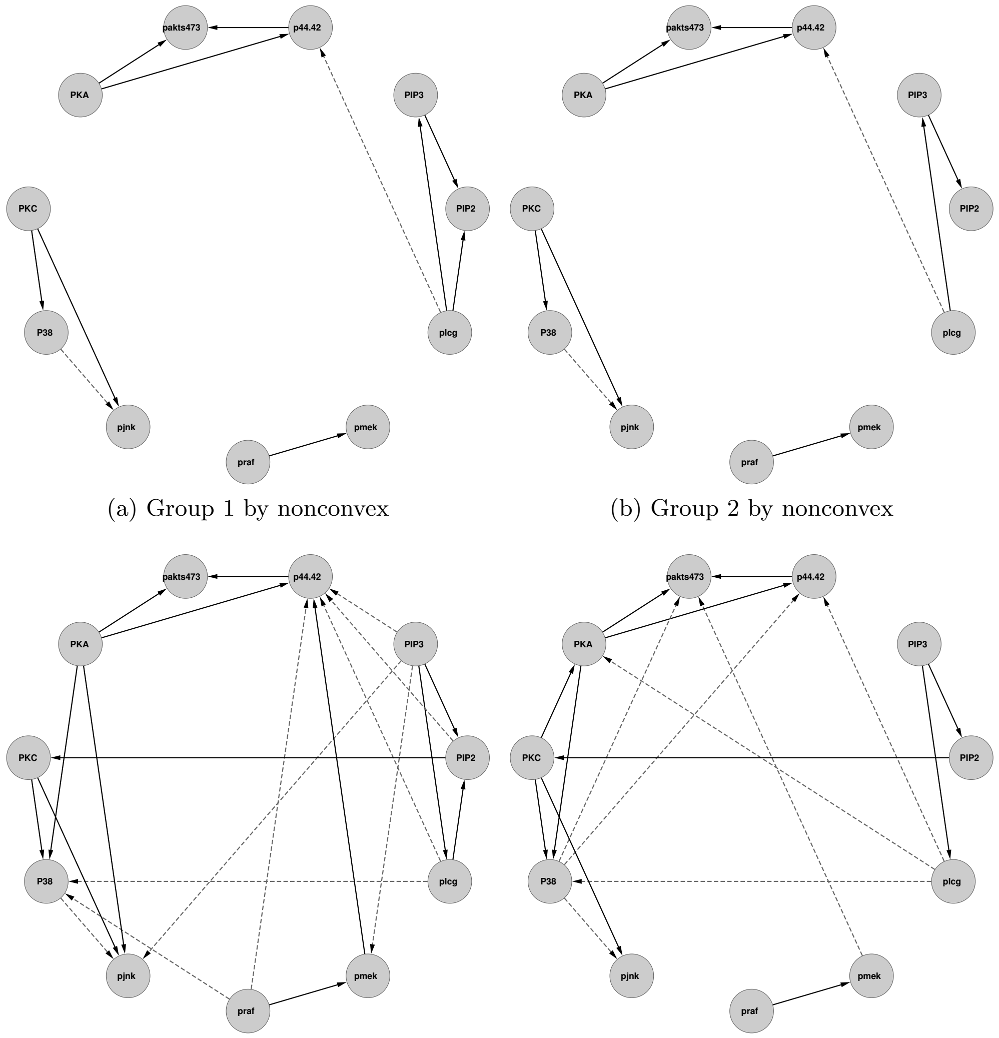


Figure 3.1: DAG representation for 11 proteins.

networks are displayed in Figure 3.2, where the DAGlasso [22] based on individual graphs is used for comparison. Correct edges are marked with solid arrows, while false positives are indicated by long dash arrows.

Our method is more reliable in that it gives much fewer false positives with almost the same number of true positives as compared to the DAGlasso, and identical links and differences between the two groups are likely to be real, which may be cross-validated experimentally. By comparison the two graph reconstructed by the DAGlasso are not so consistent and the estimated differences are mainly false discoveries.

In the analysis, several known links were missed by the proposed method and the DAGlasso method. This is because many causal relations in protein-signaling networks are believed to be nonlinear, which may not be detectable by methods based on linear models, such as our method and the DAGlasso. In fact, to our knowledge, no linear methods could reconstruct most links. This nonlinearity goes beyond the scope of the linear casual effect specified in (3.1).



(a) Group 1 by nonconvex

(b) Group 2 by nonconvex

(c) Group 1 by DAGlasso

(d) Group 2 by DAGlasso

Figure 3.2: Analysis of cell signalling data: Correct edges are marked with solid arrows, while false positives are indicated by long dashes.

Chapter 4

Learning causal networks with intervention covariates

Identifying casual relations among variables is central to many scientific investigation, as in the social, behavioral and biomedical sciences. Mathematically, the causal relations can be described by a DAG, hence that learning a DAG's structure leads to discovery of casual relations. Unfortunately, for an observational study, correct reconstruction of a DAG's structure from data is impossible, because a DAG model is often not identifiable, which is the case for a Gaussian graphical model with unequal error variances. In this chapter, we study the problem of reconstruction of a DAG's structure with the help of intervention observations. In particular, we construct a constrained likelihood to regularize intervention in addition to adjacency matrices to identify a DAG's structure and to remove redundant intervention variables.

4.1 Introduction to interventions

Directed acyclic graphical models are useful to describe pairwise causal relations between random variables, defined by a certain Markov property [34], with each node and directed edge representing one variable and the corresponding pairwise casual relation. The models have been widely used in gene and social networks [1, 44]. To identify causal relations, intervention observations are usually collected in addition to observational attributes [55]. The central topic this article addresses is reconstruction of a DAG model based on interventional data and pertinent issues with respect to the effect of intervention.

To introduce the problem of reconstruction of a DAG's structure with inventions, we begin with identifiability. It is known that DAG models are identifiable up to Markov equivalence in that directed edges are usually not all identifiable based on observational data alone [55]. However, with additional interventional data, it is generally belief that invention may lead to model identifiability thus correct reconstruction, particularly in biological experiments. For instance, intervention occurs in a form of randomized treatments in a clinical trail or of gene knockdown or knockout experiments in systems biology. In such a situation, some or all random variables are controlled, permitting discrimination of ambiguous edges connecting to these controlled variables. Yet exactly how intervention impacts reconstruction of a DAG's structure remains unknown.

In the literature, methods have been proposed to incorporate interventional data for learning causal relations; see [56] for references therein. Most of these, for instance, [57, 23, 58] assume known intervention, that is, affected variables of the intervention are known *priori* before data collection. In practice, it is often impossible to have such knowledge, as in system biology, where the effect of various chemicals intervening a system is not precisely known. One exception, to our knowledge, is the Bayesian method of [59], which intends for a small problem, say 20 nodes, due to inherited exponential complexity in the number of nodes. Therefore, scalable methods are in need as well as a theory describing the effect of intervention, particularly for a large graph.

In this chapter, we propose a method to estimate pairwise causal relations as well as the intervention effect from intervention covariates jointly. In particular, the proposed method is showed to enable to improve a model's identifiability, where inhomogeneity created by intervention facilitates reconstruction of causal relations. Most importantly, it reconstructs a DAG's structure when the model identifiable and an interventional equivalence class generally.

4.2 Learning with unknown interventions

Consider a causal system consisting of p random variables $(Y_1, \dots, Y_p)$ described by a DAG $\mathcal{G}$, with each node representing one variable and directed edges encoding causal (parent-child) relations between any two variables. The causal model factorizes the joint distribution of $(Y_1, \dots, Y_p)$, $P(Y_1, \dots, Y_p)$, into a product of conditional distributions of each variable given its parents, that is, $\prod_{j=1}^{p} P(Y_j|\mathbf{pa}_j)$, where $\mathbf{pa}_j$ denotes a parent set of Y_j and is defined to be empty if X_j has no parents.

In many situations, reconstructing directionality of relations from data may be impossible, due to nonidentifiablity with regard to reconstruction of a DAG. To overcome this difficulty, the notion of intervention is introduced and is generally believed that it can increase the degree of identifiability, thus leading to better reconstructability of a DAG. The main idea of intervention is as follows: A causal system is added into a set of W intervention variables $\{X_1, X_2, \dots, X_W\}$, continuous or discrete, where the behavior of $(Y_1, \cdots, Y_p)$ is measured and observed and some of intervention variables $\{X_1, X_2, \dots, X_W\}$ may be redundant or noninformative. Due to these extraneous variables whose values representing controllable experiment conditions, the system becomes more identifiable. However, the effect of intervention has not been quantified although it is intuitively appealing.

Given $(X_1, \cdots, X_W)$ and $(Y_1, \cdots, Y_p)$ with $i \in \mathcal{M}$ indicating interventions, the joint distribution of $(Y_1, \cdots, Y_p)$ given $(X_1, \cdots, Y_W)$ becomes

$$P(Y_1, \cdots, Y_p|X_1, \cdots, X_W) = \prod_{j \notin \mathcal{M}} P(Y_j|\mathbf{pa}_j) \prod_{j \in \mathcal{M}} \tilde{P}(Y_j|\mathbf{pa}_j, X_1, \cdots, X_W), \tag{4.1}$$

where $\tilde{P}(Y_j|\mathbf{pa}_j, X_1, \cdots, X_W)$ is an unknown probability distribution of Y_j under intervention. If $\tilde{P}(Y_j|\mathbf{pa}_j, X_1, \cdots, X_W) = \tilde{P}(Y_j|X_1, \cdots, X_W)$ is independent of the parents, then it is called perfect intervention [59]. Otherwise, it is dependent intervention [60] and practically relevant. Throughout this article, we shall consider dependent intervention.

To introduce our model, let $\boldsymbol{A} = (A_{ij})_{p \times p}$ be an adjacency matrix, which uniquely determines a DAG, where $A_{ij} \neq 0$ encodes an edge from node j to node i. Moreover, an intervention vector $\boldsymbol{B} = (B_{iw})_{p \times W}$ captures the intervention effect, whose the iwth entry B_{iw} gives the strength and direction of the intervention of X_w on Y_i. Importantly, $B_{jw} = 0$ for $j = 1, \cdots, p$, represents no intervention of X_w on Y_j, hence that X_w is noninformative and should be removed. Now placing $\boldsymbol{A}$ and $\boldsymbol{B}$ in the framework of

Gaussian structural equation models, we obtain that

$$Y_j = \sum_{k \neq j} A_{jk} Y_k + \sum_{w=1}^{W} B_{wj} X_w + Z_j, \quad Z_j \sim N(0, \sigma_j^2); \quad j = 1, \ldots, p, \tag{4.2}$$

where Z_j is a latent error representing unexplained variation in each node. Our objective is to estimate $\boldsymbol{A}, \boldsymbol{B}$ and determine their zero entries subject to the DAG requirement for $\boldsymbol{A}$. As a result, a DAG's structure is identified through $\boldsymbol{A}$, as well as a smallest dimension of the set of informative intervention variables, under which a DAG model becomes identifiable, through $\boldsymbol{B}$.

Under (4.2), two data matrices $\mathbf{Y} = (y_{ij})_{n \times p}$ and $\mathbf{X} = (x_{ij})_{n \times W}$ are observed, with n representing the sample size, possibly from different experiments under different interventions. This leads to the negative loglikelihood

$$l(\boldsymbol{\Phi}) = \sum_{j=1}^{p} \Big(\frac{1}{2\sigma_j^2} \sum_{i=1}^{n} \big(y_{ij} - u_j - \sum_{k \neq j} A_{jk} y_{ik} - \boldsymbol{B}_j^T \boldsymbol{x}_i \big)^2 + \frac{n}{2} \log \sigma_j^2 \Big), \tag{4.3}$$

where $\boldsymbol{\Phi} = (\boldsymbol{A}, \boldsymbol{B}, \sigma_1^2, \cdots, \sigma_p^2)$. We estimate $\boldsymbol{A}$ and $\boldsymbol{B}$ based on $\mathbf{Y}$ and $\mathbf{X}$. It is known *priori* that some intervention variables may be redundant. Towards our objective of learning nonzero patterns of $\boldsymbol{A}$ and $\boldsymbol{B}$, we impose sparsity constraints on them to regularize nonzero entries:

$$\sum_{1 \leq j \neq l \leq p} I(A_{jl} \neq 0) \leq K_1, \qquad \sum_{1 \leq j \leq p, 1 \leq l \leq W} I(B_{jl} \neq 0) \leq K_2, \tag{4.4}$$

where K_1 and K_2 are nonnegative integer-valued tuning parameters. It is important to note that the constraint on $\boldsymbol{B}$ aims to remove zero entries thus zero-columns of $\boldsymbol{B}$, which is equivalent to performing variable selection for intervention variables. To reinforce the DAG requirement, we impose additional constraints to reinforce the DAG requirement as in Theorem 2.2.1 to ensure no loops to occur:

$$\lambda_{ik} + I(j \neq k) - \lambda_{jk} \geq I(A_{ij} \neq 0); i, j, k = 1, \ldots, p, i \neq j, \tag{4.5}$$

$$\boldsymbol{\lambda} \in \mathbb{R}^{p^2} \tag{4.6}$$

For computation, we replace the indicator functions in (4.4) and (4.5) by its computational surrogate $J_\tau(z) = \min(\frac{|z|}{\tau}, 1)$ [7] to circumvent the difficulty of non-discontinuity in optimization. This yields that

$$\min_{(\boldsymbol{\Phi},\boldsymbol{\lambda})} l(\boldsymbol{\Phi}) \tag{4.7}$$

$$\text{subject to } \textstyle\sum_{1\leq j<l\leq p} \mathrm{J}_\tau(A_{jl}) \leq K_1, \tag{4.8}$$

$$\textstyle\sum_{1\leq j\leq p, 1\leq l\leq W} \mathrm{J}_\tau(B_{jl}) \leq K_2, \tag{4.9}$$

$$\lambda_{ik} + I(j \neq k) - \lambda_{jk} \geq \mathrm{J}_\tau(A_{ij}); i, j, k = 1, \ldots, p, j \neq i. \tag{4.10}$$

where $\mathrm{J}_\tau(z)$ approximates the indicator function as $\tau \to 0^+$.

Minimizing (4.7) in $(\boldsymbol{\Phi}, \boldsymbol{\lambda})$ subject to constraints (4.8), (4.9) and (4.10) yields the constrained maximum likelihood estimate (CMLE).

4.3 Computation

We now develop our computational strategy to minimize (4.7) with respect to $(\boldsymbol{\Phi}, \boldsymbol{\lambda})$ subject to nonconvex constraints (4.8), (4.9) and (4.10). Our strategy proceeds in two steps. First, we relax (2.11) and (2.12) using a sequence of approximations involving convex constraints, where each approximation is refined iteratively. Then we solve each convex subproblem with $p^3 - p^2 + 1$ linear constraints by employing a constrained alternating direction method of multipliers. The underlying process iterates until convergence.

For convex relaxation of nonconvex constraints (4.8), (4.9) and (4.10), we employ difference convex (DC) programming. In particular, we decompose J_τ into a difference of two convex functions: $\mathrm{J}_\tau(z) = S_1(z) - S_2(z) \equiv \min(\frac{|z|}{\tau}, 1) = \frac{|z|}{\tau} - \max(\frac{|z|}{\tau} - 1, 0)$. On this ground, we construct a sequence of convex approximating sets iteratively by replacing S_2 in the decomposition at iteration m by its affine majorization at iteration $m - 1$. Specifically, we solve

$$\begin{aligned}
\min_{(\boldsymbol{\Phi},\boldsymbol{\lambda})} \quad & l(\boldsymbol{\Phi})\text{subject to} && (4.11)\\
\frac{1}{\tau}\sum_{1\leq j\neq l\leq p}|A_{jl}|w_{jl}^{(m-1)} &\leq K_1 - \sum_{1\leq j<l\leq p}(1-w_{jl}^{(m-1)}),\\
\frac{1}{\tau}\sum_{1\leq j\leq p,1\leq l\leq W}|B_{jl}|v_{jl}^{(m-1)} &\leq K_2 - \sum_{1\leq j\leq p,1\leq l\leq W}(1-v_{jl}^{(m-1)}),\\
\lambda_{js}+\tau I(l\neq s)-\lambda_{ls} &\geq |A_{jl}|_1 w_{jl}^{(m-1)}+\tau(1-w_{jl}^{(m-1)}); j,l,s=1,\ldots,p, j\neq l,
\end{aligned}$$

where $(\hat{\boldsymbol{A}}^{(m-1)}, \hat{\boldsymbol{B}}^{(m-1)})$ is the solution at iteration $m-1$; $1 \leq i,j \leq p$, $w_{jl}^{(m-1)} = I(\|\hat{A}_{jl}^{(m-1)}\|_1 \leq \tau)$, and $v_{jl}^{(m-1)} = I(|\hat{B}_{jl}^{(m-1)}| \leq \tau)$.

To solve (4.11), we consider its equivalent form for efficient computation:

$$\begin{aligned}
\min\nolimits_{(\boldsymbol{\Phi},\boldsymbol{\lambda})} l(\boldsymbol{\Phi}) + \tfrac{\mu_1}{\tau}\textstyle\sum_{1\leq j\neq l\leq p}|A_{jl}|w_{jl}^{(m-1)} + \tfrac{\mu_2}{\tau}\textstyle\sum_{1\leq j\leq p,1\leq l\leq W}|B_{jl}|v_{jl}^{(m-1)} \text{ subj to}\\
\lambda_{js}+\tau I(l\neq s)-\lambda_{ls} \geq |A_{jl}|_1 w_{jl}^{(m-1)}+\tau(1-w_{jl}^{(m-1)}); j,l,s=1,\ldots,p, j\neq l, (4.12)
\end{aligned}$$

where μ_1 and mu_2 are nonnegative regularizers corresponding to the first and second constraints in (4.11).

To proceed, let $\boldsymbol{\xi} = \{\xi_{ijk}\}_{p\times p\times p}$ be a slack variable tensor, converting inequality to equality constraints. And we introduce $\boldsymbol{F}_{p\times p}$ to separate the differentiable from non-differentiable parts involving L_1-norm of $\boldsymbol{A}$. Then the problem can be written in the form as

$$\begin{aligned}
\min\nolimits_{(\boldsymbol{\Phi},\boldsymbol{F},\boldsymbol{\lambda})} l(\boldsymbol{\Phi}) + \tfrac{\mu_1}{\tau}\textstyle\sum_{1\leq j\neq l\leq p}|F_{jl}|w_{jl}^{(m-1)} + \tfrac{\mu_2}{\tau}\textstyle\sum_{1\leq j\leq p,1\leq l\leq W}|B_{jl}|v_{jl}^{(m-1)}\\
\text{subject to } \lambda_{js}+\tau I(l\neq s)-\lambda_{ls}-|F_{jl}|_1 w_{jl}^{(m-1)}-\tau(1-w_{jl}^{(m-1)})-\xi_{ijk}=0;\\
j,l,s=1,\ldots,p,\ j\neq l,\ \xi_{ijk}\geq 0,\ \boldsymbol{A}-\boldsymbol{F}=\boldsymbol{0}. \qquad (4.13)
\end{aligned}$$

Following [38], we obtain an augmented Lagrangian by introducing the scaled dual variable tensor $\boldsymbol{y} = \{y_{ijk}\}_{p\times p\times p}$ and the scale dual variable matrix $\boldsymbol{U} = \{u_{ij}\}_{p\times p}$:

$$\begin{aligned} L_\rho(\boldsymbol{A},\boldsymbol{B},\boldsymbol{F},\boldsymbol{\lambda},\boldsymbol{\xi},\boldsymbol{y},\boldsymbol{U},\boldsymbol{\sigma}) &= l(\boldsymbol{A},\boldsymbol{B},\boldsymbol{\sigma}) \\ &+\tfrac{\mu_1}{\tau}\textstyle\sum_{1\le j\neq l\le p}|F_{jl}|w_{jl}^{(m-1)}+\tfrac{\mu_2}{\tau}\textstyle\sum_{1\le j\le p,1\le l\le W}|B_{jl}|v_{jl}^{(m-1)} \\ &+\textstyle\sum_{1\le s\le p}\sum_{1\le j\neq l\le p}\tfrac{\rho}{2}\left(|F_{jl}|w_{jl}^{(m-1)}+\tau(1-w_{jl}^{(m-1)})+\xi_{jls}-\lambda_{jl}-\tau I(l\neq s)+\lambda_{ls}+y_{jls}\right)^2 \\ &+\tfrac{\rho}{2}\|\boldsymbol{F}-\boldsymbol{A}+\boldsymbol{U}\|_F^2. \end{aligned} \tag{4.14}$$

Iteratively, we solve (4.14) over blocks $(\boldsymbol{A},\boldsymbol{B},\boldsymbol{F},\boldsymbol{\lambda},\boldsymbol{\xi},\boldsymbol{y},\boldsymbol{U},\boldsymbol{\sigma})$. Updating formulas are similar to ones described in Chapter 2.

4.4 Numerical examples

This section examines operating characteristics of the proposed method, and demonstrates how intervention improves reconstructability of a DAG's structure.

In this simulated example, the proposed method is applied to interventional data $(\mathbf{Y}^{(I)},\mathbf{X})$ and that in Chapter 2 to observational data $\mathbf{Y}^{(O)}$. The result is summarized in Table 4.1.

Simulation setting: This example contrasts the proposed method with intervention and that without intervention to understand the intervention effect. In particular, a DAG with 50 nodes is used with a random generation mechanism as described in [11], see Figure 4.1 for a display. Then edge weights were randomly drawn from a uniform distribution on $[-2,-1]\cup[1,2]$. The error variances for all nodes of the DAG is set to be a decreasing sequence from 1 to 0.5 with equal distance. Two random samples were generated: interventional $\mathbf{Y}^{(I)},\mathbf{X}$ and observational $\mathbf{Y}^{(O)}$. For the interventional sample, we construct 50 interventional variables X_i, each has a intervention effect on one corresponding DAG node Y_i, that is, the matrix $\boldsymbol{B}$ is a diagonal matrix. The diagonal values of $\boldsymbol{B}$ are drawn randomly from a uniform distribution over $[-4,-2]\cup[2,4]$. Note that this design of $\boldsymbol{B}$ ensures that the DAG is fully identifiable in view of the result of Theorem 2. As a result, $\mathbf{X}_{200\times 50}$ is drawn from independent $N(0,1)$, and $Y^{(I)}$ is generated according to (4.2). The observational sample $\mathbf{Y}^{(O)}$ is generated according to (4.2) with $\boldsymbol{B}=\mathbf{0}$.

For accuracy of estimating directed edges of a graph, we consider the false negative rate (FNR) and the false discovery rate (FDR), defined as FNR = FN/(TP + FN) and FDR = FP/(TP + FP), where TP, FP, TN and FN are true positive, false positive, true negative and false negative numbers of edge estimation, respectively.

For overall accuracy of estimating the DAG structure, we employ a commonly used measure–Structural Hamming Distance (SHD). The SHD between two DAGs is the required number of edge insertions, deletions or flips to transform one graph to another graph, c.f., [43]. A smaller SHD indicates closeness of two graphs. To compute the SHD, one may consider the R-package pcalg.

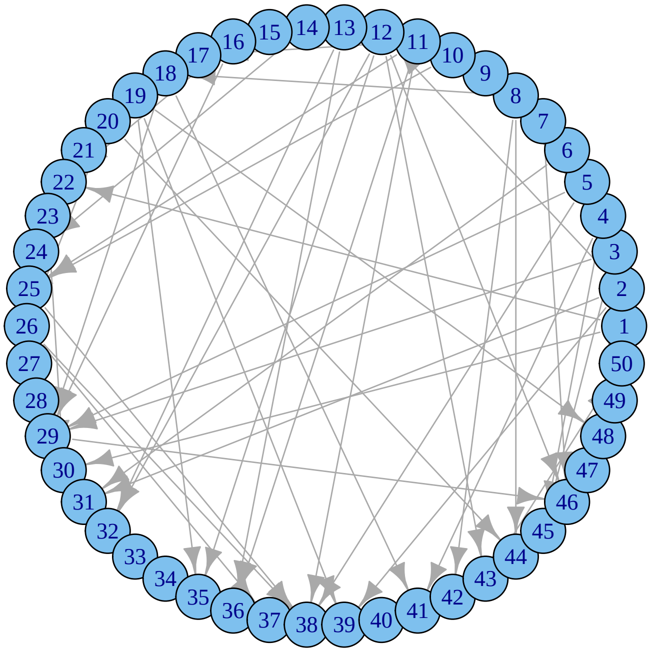


Figure 4.1: The DAG used in the simulation study

As suggested in Table 4.1, the proposed method compares favorable against the

Table 4.1: Averaged false negative rate (FNR), false discover rate (FDR), and Structural Hamming Distance (SHD), as well as their standard errors (in parenthesis), for four competing methods based on 100 simulation replications. Here "Non-Int" and "Int" denote the method in Chapter 2 applied to observational data only, and the proposed method applied to interventional data.

(n, p)	Method	FNR(A)	FDR(A)	SHD
(100,50)	Non-Int	0.05(0.01)	0.75(0.04)	117.9(15.1)
(100,50)	Int	0.01(0)	0.49(0.08)	49.6(8.5)
(200,50)	Non-Int	0.03(0.01)	0.66(0.04)	85.8(12.4)
(200,50)	Int	0.01(0)	0.25(0.07)	25.6(6.9)
(1000,50)	Non-Int	0.01(0)	0.24(0.06)	19.2(6.9)
(1000,50)	Int	0(0)	0.1(0.03)	8.2(2.6)

method "Non-Int" with observational data. The significant improvement in terms of structure learning implies the benefit of incorporating interventional data.

Chapter 5

Conclusions

This thesis set out to explore novel statistical learning methods for Gaussian DAG models, especially in the high-dimensional cases. Our contribution lies in providing a general framework for simultaneous learning of DAG structures and model parameters, which ultimately enables better learning performance and efficient computation for large networks. Our proposed methods overcome the drawbacks of local and sequential search algorithms in the literature. Moreover, our theoretical analysis provides more insight into the nature of the problem as well as finite sample error bounds for statistical learning.

5.1 Summary of major findings

We now summarize our contributions specifically for each chapter. In Chapter 2, we proposed a novel constrained likelihood approach for learning the DAG model from observational data. The constrained likelihood framework enables us to learn nonzero patterns and numerical values of the adjacency matrix $\boldsymbol{A}$ simultaneously. This overcomes the error accumulation resulting from previous local search algorithms which separate structure learning and parameter estimation. Computationally, we design a representation of the DAG parameter space with cubically many constraints. This, combined with a constrained alternating direction method of multipliers and difference convex programming, makes it possible to solve this problem with hundreds of nodes, thus leading to efficient computation involving a complexity of order $O(p^3)$. Theoretically, we develop a theory to quantify what the proposed method can accomplish, where the focus is equal error variance for identifiable DAG models. We show that it consistently reconstructs the true directed acyclic graph under a degree of reconstructability assumption (2.19).

In Chapter 3, we moved from learning a single DAG model to learning multiple DAGs when data are inhomogeneous. We proposed a maximum likelihood method for such tasks. Another different scope from Chapter 2 is that we assume a known ordering of nodes. Our proposed method is the first of its kind to address such a task for real applications. Also in this chapter, we proposed a pairwise coordinate descent algorithm, which is a fast algorithm for solving constrained $L-1$ problems.

In Chapter 4, we study the problem of reconstruction of a DAG's structure with the help of intervention observations with unknown intervention effects. This allows us to analyze more general interventional data where intervention effects are unknown before the experiments. Thus we have an objective of learning intervention effects in addition to learning the DAG itself. In particular, we construct a constrained likelihood to regularize intervention in addition to adjacency matrices to identify a DAG's structure and remove redundant intervention variables. Computationally, we design an efficient algorithm for implementation. In simulations, we show that the proposed method leads to higher accuracy of reconstruction with the help of interventional data.

5.2 Extensions and future work

In addition to providing immediate important findings, the work in this thesis has also motivated a variety of future projects.

5.2.1 Computational alternatives

The computational approach described in Chapter 2 provides a novel and efficient way to search for a DAG that fits data best. However, current implementation still struggles computationally when the number of nodes exceed one thousand. For large-scale problems, we are seeking a necessary and sufficient partition rule for our method. This will permit fast computation by partitioning nodes into disjoint subsets and applying our method to each subset. Another direction for computational improvement could be finding an alternative approach to the $O(p^3)$ constraints for acyclicity. Checking acyclicity is computationally efficient with linear time complexity in the number of nodes and the number of edges. If we can quantify the number of times we need to check acyclicity, we may no long need explicit constraints for acyclicity. In addition, we are seeking distributed algorithms in order to scale up our method. Such increasing computing power would also enable us to investigate nonlinear causal relationships, which is another direction of my research.

5.2.2 Network learning incorporating additional covariates

One important application of DAG models is constructing biological networks. We demonstrated favorable performance of our method through analyzing cell-signaling data. Yet, many gene expression data come with covariates such as SNP markers, which also may include thousands of variables. This brings a promising project of network construction incorporating additional covariates and modeling the influence of genetic information on gene expression. This project targets at real-world applications involving massive amounts of data. Moreover, many biological networks are dynamic in nature. Such Dynamic network modeling is another promising direction for future research.

Turning to practice, we are also developing R packages for the proposed methods. The thesis provide a practical guide on how to carry out the proposed methods. Yet it would be more convenient to have free reliable software ready for applications.

5.2.3 Identifiability issues

The issue of identifiability with the DAG model is consistent with the general belief that identifying cause-and-effect relationships by observation alone is not always possible. However, with suitable assumptions, the model becomes identifiable. In addition to the aforementioned equal error and known ordering assumptions, theories are in need for other suitable assumptions under which causal relationships become identifiable. One direction is to develop new theories regarding identifiability issues as this is of primary interest in finding causal relationships. Another research direction is to design intervention experiments to collect interventional data that complement information from observational data in causal inference. Many scientific causality investigations rely on such interventional experiments. Therefore, it is of both theoretical and practical interest to investigate network learning with interventional data.

References

[1] N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science Signalling*, 303(5659):799, 2004.

[2] Steffen L Lauritzen. *Graphical models.* Oxford University Press, 1996.

[3] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.

[4] J. Peters and P. Bühlmann. Identifiability of gaussian structural equation models with equal error variances. *Biometrika, first published online, doi: 10.1093/biomet/ast043*, 2013.

[5] S. van de Geer and P. Bühlmann. l_0-penalized maximum likelihood for sparse directed acyclic graphs. *The Annals of Statistics*, 41(2):536–567, 2013.

[6] S. A. Andersson, D. Madigan, and M. D. Perlman. A characterization of markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2):505–541, 1997.

[7] X. Shen, W. Pan, and Y. Zhu. Likelihood-based selection and sharp parameter estimation. *Journal of the American Statistical Association*, 107:223–232, 2012.

[8] P. Spirtes, C.N. Glymour, and R. Scheines. *Causation, prediction, and search*, volume 81. The MIT Press, 2000.

[9] Luis M. de Campos. Independency relationships and learning algorithms for singly connected networks. *Journal of Experimental & Theoretical Artificial Intelligence*, 10(4):511–549, 1998.

[10] J. Cheng, R. Greiner, J. Kelly, D. Bell, and W. Liu. Learning bayesian networks from data: an information-theory based approach. *Artificial Intelligence*, 137(1-2):43–90, 2002.

[11] M. Kalisch and P. Bühlmann. Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *The Journal of Machine Learning Research*, 8:613–636, 2007.

[12] L. E. Brown, I. Tsamardinos, and C. F. Aliferis. A comparison of novel and state-of-the-art polynomial bayesian network learning algorithms. In *Proceedings of the national conference on artificial intelligence*, volume 20, page 739. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

[13] D. M. Chickering. Optimal structure identification with greedy search. *The Journal of Machine Learning Research*, 3:507–554, 2003.

[14] L. M. de Campos. A scoring function for learning bayesian networks based on mutual information and conditional independence tests. *Journal of Machine Learning Research*, 7(2):2149, 2007.

[15] R. W. Robinson. Counting unlabeled acyclic digraphs. In *Combinatorial Mathematics V*, pages 28–43. Springer, 1977.

[16] Wray Buntine. Theory refinement on bayesian networks. In *Proceedings of the Seventh conference on Uncertainty in Artificial Intelligence*, pages 52–60. Morgan Kaufmann Publishers Inc., 1991.

[17] Gregory F Cooper and Edward Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9(4):309–347, 1992.

[18] D. M. Chickering, D. Heckerman, and C. Meek. Large-sample learning of bayesian networks is np-hard. *The Journal of Machine Learning Research*, 5:1287–1330, 2004.

[19] Mark Schmidt and Kevin Murphy. Modeling discrete interventional data using directed cyclic graphical models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 487–495. AUAI Press, 2009.

[20] Brandon Malone, Kustaa Kangas, Matti Järvisalo, Mikko Koivisto, and Petri Myllymäki. Predicting the hardness of learning bayesian networks. *AAAI Conference on Artificial Intelligence*, 2014.

[21] Han Liu, John Lafferty, and Larry Wasserman. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *The Journal of Machine Learning Research*, 10:2295–2328, 2009.

[22] A. Shojaie and G. Michailidis. Penalized likelihood methods for estimation of sparse high-dimensional directed acyclic graphs. *Biometrika*, 97:519–538, 2010.

[23] F. Fu and Q. Zhou. Learning sparse causal gaussian networks with experimental intervention: Regularization and coordinate descent. *Journal of the American Statistical Association*, 108(501):288–300, 2013, http://www.tandfonline.com/doi/pdf/10.1080/01621459.2012.754359.

[24] Mikko Koivisto and Kismat Sood. Exact bayesian structure discovery in bayesian networks. *The Journal of Machine Learning Research*, 5:549–573, 2004.

[25] Tommi Jaakkola, David Sontag, Amir Globerson, and Marina Meila. Learning bayesian network structure using lp relaxations. In *International Conference on Artificial Intelligence and Statistics*, pages 358–365, 2010.

[26] Cassio P De Campos and Qiang Ji. Efficient structure learning of bayesian networks using constraints. *The Journal of Machine Learning Research*, 12:663–689, 2011.

[27] Robert Peharz and Franz Pernkopf. Exact maximum margin structure learning of bayesian networks. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 1047–1054, 2012.

[28] James Cussens, Mark Bartlett, Elinor M Jones, and Nuala A Sheehan. Maximum likelihood pedigree reconstruction using integer linear programming. *Genetic epidemiology*, 37(1):69–83, 2013.

[29] Leo Breiman and Adele Cutler. A deterministic algorithm for global optimization. *Mathematical Programming*, 58(1-3):179–199, 1993.

[30] Le Thi Hoai An and Pham Dinh Tao. Solving a class of linearly constrained indefinite quadratic problems by dc algorithms. *Journal of Global Optimization*, 11(3):253–285, 1997.

[31] R. R. Bahadur, J. C. Gupta, and S. L. Zabell. Large deviations, tests and estimates. *Asymptotic Theory of Statistical Tests and Estimation*, Hoeffding Festschrift (I. M. Chakravarti, ed.):33–64, Academic Press, New York, 1980.

[32] J. Zhang and P. Spirtes. Strong faithfulness and uniform consistency in causal inference. In *Proceedings of the nineteenth conference on uncertainty in artificial intelligence*, pages 632–639. Morgan Kaufmann Publishers Inc., 2002.

[33] C. Uhler, G. Raskutti, P. Bühlmann, and B. Yu. Geometry of the faithfulness assumption in causal inference. *The Annals of Statistics*, 41(2):436–463, 2013.

[34] D. Edwards. *Introduction to graphical modelling.* Springer Verlag, 2000.

[35] D. L. Zimmerman and V. A. Nunez-Anton. *Antedependence models for longitudinal data.* CRC Press, 2010.

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

[37] Martin Grötschel, Michael Jünger, and Gerhard Reinelt. On the acyclic subgraph polytope. *Mathematical Programming*, 33(1):28–42, 1985.

[38] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[39] R. T. Rockafellar and R. J. Wets. *Variational Analysis.* Springer-Verlag, New York, 2003.

[40] X. Shen, W. Pan, Y. Zhu, and H. Zhou. On constrained and regularized high-dimensional regression. *Annals of the Institute of Statistical Mathematics*, pages 1–26, 2013.

[41] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge University Press, 2004.

[42] K. B. Korb and A. . Nicholson. *Bayesian artificial intelligence*, volume 1. Chapman & Hall/CRC, 2003.

[43] I. Tsamardinos, L. E. Brown, and C.F. Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, 65:31–78, 2006.

[44] K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523, 2005.

[45] Shuheng Zhou, John Lafferty, and Larry Wasserman. Time varying undirected graphs. *Machine Learning*, 80(2-3):295–319, 2010.

[46] M. Kolar, L. Song, A. Ahmed, and E.P. Xing. Estimating time-varying networks. *The Annals of Applied Statistics*, 4(1):94–123, 2010.

[47] J. Guo, E. Levina, G. Michailidis, and J. Zhu. Joint estimation of multiple graphical models. *Biometrika*, 98(1):1, 2011.

[48] N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462, 2006.

[49] M. Yuan and Y. Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19, 2007.

[50] J. Pearl. *Causality: models, reasoning and inference.* Cambridge Univ Press, 2000.

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

[52] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

[53] T. T. Wu and K. Lange. Coordinate descent algorithms for lasso penalized regression. *The Annals of Applied Statistics*, 2(1):224–244, 2008.

[54] P. Baldi, S. Brunak, Y. Chauvin, C.A.F. Andersen, and H. Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412–424, 2000.

[55] J. Pearl. Statistics and causal inference: A review. *Test*, 12(2):281–345, 2003.

[56] A. Hauser and P. Bühlmann. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *The Journal of Machine Learning Research*, 13(1):2409–2464, 2012.

[57] B. Ellis and W. H. Wong. Learning causal bayesian network structures from experimental data. *Journal of the American Statistical Association*, 103(482):778–789, 2008.

[58] F. Fu and Q. Zhou. Learning sparse causal gaussian networks with experimental intervention: regularization and coordinate descent. *Journal of the American Statistical Association*, 108(501):288–300, 2013.

[59] D. Eaton and K. P. Murphy. Exact bayesian structure learning from uncertain interventions. In *International Conference on Artificial Intelligence and Statistics*, pages 107–114, 2007.

[60] Frederick Eberhardt and Richard Scheines. Interventions and causal inference. *Philosophy of Science*, 74(5):981–995, 2007.

[61] J. D. Dixon and B. Mortimer. Permutation groups. In *Combinatorial mathematics V*. Springer, 1996.

[62] X. Shen and H. S. Huang. Grouping pursuit through a regularization surface. *Journal of the American Statistical Association*, 105:727–739, 2010.

[63] A. N. Kolmogorov and V. M. Tikhomirov. ε-entropy and ε-capacity of sets in function spaces. *Uspekhi Matematicheskikh Nauk*, 14(2):3–86, 1959.

[64] W. H. Wong and X. Shen. Probability inequalities for likelihood ratios and convergence rates of sieve mles. *The Annals of Statistics*, 23(2):339–362, 1995.

[65] M. Ossiander. A central limit theorem under metric entropy with l_2 bracketing. *The Annals of Probability*, 15(3):897–919, 1987.

[66] P. Stanica. Good lower and upper bounds on binomial coefficients. *Journal of Inequalities in Pure and Applied Mathematics*, 2(3):30, 2001.

[67] M.R. Osborne, B. Presnell, and B.A. Turlach. On the lasso and its dual. *Journal of Computational and Graphical statistics*, pages 319–337, 2000.

Appendix A

Technical details

A.1 Technical details for Chapter 2

A.1.1 Analytic updating expressions for ADMM in (2.18)

For $\boldsymbol{A}$ direction, the following optimization need to be solved,

$$\min_{\boldsymbol{A}} \sum_{j=1}^{p} \big(\frac{1}{2}\sum_{i=1}^{n}\big(x_{ij} - \sum_{k\neq j} x_{ik}A_{jk}\big)^2\big) + \frac{\rho}{2}\|\boldsymbol{A} - \boldsymbol{B}^{(s)} + \boldsymbol{U}^{(s)}\|_F^2 \tag{A.1}$$

This problem is separable in j. Therefore, for each row $\boldsymbol{A}_{j,j^-}$, with $A_{j,j}$ excluded, we solve

$$\min_{\boldsymbol{A}_{j,j^-}} \big(\frac{1}{2}\sum_{i=1}^{n}\big(x_{ij} - \sum_{k\neq j} x_{ik}A_{jk}\big)^2\big) + \frac{\rho}{2}\|\boldsymbol{A}_{j,j^-} - \boldsymbol{B}^{(s)}_{j,j^-} + \boldsymbol{U}^{(s)}_{j,j^-}\|^2, \tag{A.2}$$

where the minimizer for $\boldsymbol{A}_{j,j^-}$ satisfies:

$$\big(\boldsymbol{X}^T_{j^-}\boldsymbol{X}_{j^-} + \rho\boldsymbol{I}\big)\boldsymbol{A}_{j,j^-} = \boldsymbol{X}^T_{j^-}\boldsymbol{x}_j + \rho(\boldsymbol{B}^{(s)}_{j,j^-} - \boldsymbol{U}^{(s)}_{j,j^-}). \tag{A.3}$$

Hence we only need to compute the matrix inverse of $\big(\boldsymbol{X}^T\boldsymbol{X} + \rho\boldsymbol{I}\big)$ once, as all other inverse matrices are calculated through the formula: $(\boldsymbol{\Omega}_{-j})^{-1} = (\boldsymbol{\Omega}^{-1})^{-j} - (\boldsymbol{\Omega}^{-1})_{j^-j}$ $(\boldsymbol{\Omega}^{-1})_{jj^-}/(\boldsymbol{\Omega}^{-1})^{jj}$. Note that the factorization of $\boldsymbol{X}^T_{j^-}\boldsymbol{X}_{j^-} + \rho\boldsymbol{I}$ can be cached to speed up subsequent updates.

To compute $\boldsymbol{B}$, we solve the following subproblem subject to $\boldsymbol{B} \geq 0$.

$$\begin{aligned}\min_{\boldsymbol{B}} \qquad & \tfrac{\rho}{2}\|\boldsymbol{B}-\boldsymbol{A}^{(s+1)}-\boldsymbol{U}^{(s)}\|_F^2 \\ & +\textstyle\sum_k\sum_{i\neq j}\tfrac{\rho}{2}(|B_{ij}|w_{ij}^{(m-1)}+\tau(1-w_{ij}^{(m-1)})-C_{ijk}^{(s)})^2 \\ & +\mu(\textstyle\sum_{i\neq j}|B_{ij}|w_{ij}^{(m-1)}),\end{aligned} \tag{A.4}$$

where $C_{ijk}^{(s)}=\lambda_{ik}^{(s)}+\tau I(j\neq k)-\lambda_{jk}^{(s)}-y_{ijk}^{(s)}-\xi_{ijk}^{(s)}$. This elementwise minimization leads to

$$B_{ij}^{(s+1)}=\begin{cases}\operatorname{sign}(A_{ij}^{(s+1)}+U_{ij}^{(s)})\left(\frac{\rho(|A_{ij}^{(s+1)}+U_{ij}^{(s)}|+\sum_k C_{ijk})-\mu}{(1+p)\rho}\right)^+ & \text{if } w_{ij}^{(m-1)}=1,\\ A_{ij}^{(s+1)}+U_{ij}^{(s)} & w_{ij}^{(m-1)}=0,\end{cases} \tag{A.5}$$

$$\boldsymbol{U}^{(s+1)}=\boldsymbol{U}^{(s)}+\big(\boldsymbol{A}^{(s+1)}-\boldsymbol{B}^{(s+1)}\big). \tag{A.6}$$

For $(\boldsymbol{\lambda}^{s+1},\xi^{s+1})$, the updating formulas are:

$$\begin{aligned}\boldsymbol{\lambda}^{s+1} &:= \boldsymbol{M}_{p\times p}\boldsymbol{W}_{p\times p}^{s},\\ \xi_{ijk}^{s+1} &:= \max(0,(\tau\lambda_{ik}^{s}+\tau I(j\neq k)-\tau\lambda_{jk}^{s}-|B_{ij}^{s+1}|-y_{ijk}^{s}));i,j,k=1,\ldots,p.\end{aligned}$$

where

$$\boldsymbol{M}_{p\times p}=\frac{1}{\tau}\begin{pmatrix}1 & 0 & 0 & \ldots & 0\\ 1 & \frac{2}{p} & \frac{1}{p} & \ldots & \frac{1}{p}\\ 1 & \frac{1}{p} & \frac{2}{p} & ... & \frac{1}{p}\\ \vdots & \vdots & & \ddots & \vdots\\ 1 & \frac{1}{p} & \ldots & \frac{1}{p} & \frac{2}{p}\end{pmatrix},$$

$$\begin{aligned}W_{1j}^{s+1} &= 1,\\ W_{ik}^{s+1} &= \tfrac{1}{2}(\tau+\textstyle\sum_j(|B_{kj}^{s+1}|w_{ij}^{(m-1)}+\tau(1-w_{ij}^{(m-1)})+\xi_{ijk}^{s+1}+y_{ijk}^{s})\\ &\quad -\textstyle\sum_j(|B_{kj}^{s+1}|w_{ij}^{(m-1)}+\tau(1-w_{ij}^{(m-1)})+\xi_{jik}^{s+1}+y_{jik}^{s})); \quad i\neq k,\\ W_{kk}^{s+1} &= \tfrac{1}{2}(-(p-1)\tau+\textstyle\sum_j(|B_{kj}^{s+1}|w_{ij}^{(m-1)}+\tau(1-w_{ij}^{(m-1)})+\xi_{kjk}^{s+1}+y_{kjk}^{s})\\ &\quad -\textstyle\sum_j(|B_{kj}^{s+1}|w_{ij}^{(m-1)}+\tau(1-w_{ij}^{(m-1)})+\xi_{jkk}^{s+1}+y_{jkk}^{s})),\end{aligned}$$

for $i,j,k=1,\ldots,p$.

A.1.2 Technical proofs

Before proving Theorem 1, we need a technical lemma.

Lemma 1

For $k = 1, \ldots, p$, let $c_{ij}^k = I(A_{ij} \neq 0)$ if $i \neq j$; $c_{ij}^k = 0$ if $i = j = k$; $c_{ij}^k = 1$ otherwise. Then constraints in (2.1) are equivalent to those as follows: for $k = 1, \ldots, p$,

$$\begin{aligned} (p-1) \geq \quad & \max_{q_{ij}} \sum_{1 \leq i,j \leq p} c_{ij}^k q_{ij} && \text{(A.7)} \\ & \text{subj to} \sum_{j=1}^p q_{ij} = 1, \sum_{i=1}^p q_{ij} = 1, q_{ij} \geq 0, i,j = 1, \ldots, p. && \text{(A.8)} \end{aligned}$$

Proof of Lemma 1: Note that (A.8) defines the class of all doubly stochastic matrices and satisfies the unimodular property, which is a convex polytope. By the Birkhoff von Neumann theorem [61], the vertices of this polytope are precisely the permutation matrices having exactly one unit entry in each row and each column and zeros elsewhere. Then a maximizer $\tilde{Q} = \{\tilde{q}_{ij}\}$ of $\max_{q_{ij}} \sum_{1 \leq i,j \leq p} c_{ij}^k q_{ij}$ subject to (A.8) is attained at some vertices of the polytope, hence that it must be a permutation matrix.

First, suppose $\boldsymbol{A}$ satisfies (2.1). We need to prove that for each k, the maximizer $\tilde{Q}$ satisfies $(p-1) \geq \sum_{1 \leq i,j \leq p} c_{ij}^k \tilde{q}_{ij}$ from (A.8). The permutation matrix $\tilde{Q}$ has its nonzero entries forming dicycle [61]. If the dicycle of $\tilde{Q}$ are trivial dicycle (self loops), with each containing one element, then the maximal of $\sum_{i,j} c_{i,j}^k q_{i,j} = \sum_{i=1}^p c_{i,i}^k$ becomes $(p-1)$, implying (A.8). Otherwise, there are T trivial and S non-trivial dicycle, denoted by $(j_1^1, \ldots, j_{L_1}^1)$, $(j_1^2, \ldots, j_{L_2}^2)$, $\ldots$, $(j_1^S, \ldots, j_{L_S}^S)$, with the length of each dicycle L_s, and $\sum_{s=1}^S L_s = p - T$. The maximal of $\sum_{1 \leq i,j \leq p} c_{ij}^k \tilde{q}_{ij}$ is no greater than $T + \sum_{s=1}^S \sum_{j_1^s = j_{L+1}^s : 1 \leq k \leq L^s} I(A_{j_k j_{k+1}} \neq 0) \leq (T + \sum_{s=1}^S L_s - S) = (p - S) \leq (p-1)$, implying (A.8).

Next we prove the converse by contradiction. Suppose $\{c_{ij}\}$ satisfies (A.8). If there exists a dicycle $(j_1, \ldots, j_L)$ of length L for $\boldsymbol{A}$ such that $\sum_{j_1 = j_{L+1} : 1 \leq k \leq L} I(A_{j_k j_{k+1}} \neq 0) > (L-1)$, then we choose $k^0 \in \{j_2, \ldots, j_L\}$ and construct a permutation matrix using this dicycle in that $q_{j_1 j_2}, \ldots, q_{j_{L-1} j_L}, q_{j_L, j_1} = 1$, $q_{j,j} = 1$ for $j \notin (j_1, \ldots, j_L)$, and $q_{ij} = 0$ otherwise. Then $\sum_{i,j} c_{ij}^{k_0} q_{ij} = \sum_{j_1 = j_{L+1} : 1 \leq k \leq L} A_{j_k j_{k+1}} + (p - L) > (L-1) + (p-L) = (p-1)$. This contradicts to (A.8).

Proof of Theorem 1: Note that (A.7) and (A.8) involve p linear programming problems as follows: for $k = 1, \ldots, p$,

$$(\mathbf{Primal_k}) \quad \max_{q_{ij}} \sum_{1\le i,j\le p} c^k_{ij} q_{ij}$$
$$\text{subj to} \quad \sum_{j=1}^{p} q_{ij} = 1, \sum_{i=1}^{p} q_{ij} = 1, q_{ij} \ge 0, i, j = 1, \dots, p.$$

For $\mathbf{Primal_k}$, consider its dual problem by introducing $2p$ Lagrange multipliers $\{\lambda_{ik} : 1 \le i \le p\}$ for each equality $\sum_{i=1}^p q_{ij} = 1, i = 1, \dots, p$ and $\{\mu_{jk} : 1 \le j \le p\}$ for each equation $\sum_{j=1}^p q_{ij} = 1, j = 1, \dots, p$:

$$(\mathbf{Dual_k}) \quad \min_{(\lambda_{ik},\mu_{jk})} \textstyle\sum_{i=1}^p \lambda_{ik} + \sum_{j=1}^p \mu_{jk}$$
$$\text{subj to } \lambda_{ik} + \mu_{jk} \ge c^k_{ij}, 1 \le i, j \le p.$$

By the strong duality theorem, there is no duality gap between the primal and the dual in this case. Then replacing the $\mathbf{Primal_k}$ in (A.7) and (A.8) by the $\mathbf{Dual_k}$ yileds that, for $k = 1, \dots, p$,

$$p - 1 \ge \min_{(\lambda_{ik},\mu_{jk})} \textstyle\sum_{i=1}^p \lambda_{ik} + \sum_{j=1}^p \mu_{jk}$$
$$\text{subj to } \lambda_{ik} + \mu_{jk} \ge c^k_{ij}, 1 \le i, j \le p.$$

Note further that $c^k_{ij} = I(A_{ij} \ne 0)$ if $i \ne j$; $c^k_{ij} = 0$ if $i = j = k$; $c^k_{ij} = 1$ otherwise. Plugging back, for $k = 1, \dots, p$,

$$p - 1 \ge \min_{(\lambda_{ik},\mu_{jk})} \textstyle\sum_{i=1}^p \lambda_{ik} + \sum_{j=1}^p \mu_{jk} \tag{A.9}$$
$$\text{subj to } \lambda_{ik} + \mu_{jk} \ge I(A_{ij} \ne 0), i \ne j; \lambda_{ik} + \mu_{ik} \ge 1, i \ne k; \lambda_{kk} + \mu_{kk} \ge 0. \tag{A.10}$$

Note that (A.9) and (A.10) impose constraints on $\boldsymbol{A}$ because (λ_{ik}, μ_{jk}) are minimized out. In what follows, we prove that (A.9) and (A.10) are equivalent to a set of constraints (A.11) and (A.12) on $(A_{ij}, \lambda_{ik}, \mu_{jk})$. Specifically, there exist a $\{(\lambda_{ik}, \mu_{jk}) : 1 \le i, j \le p\}$ such that

$$(p - 1) \ge \textstyle\sum_{i=1}^p \lambda_{ik} + \sum_{j=1}^p \mu_{jk}, \tag{A.11}$$
$$\lambda_{ik} + \mu_{jk} \ge I(A_{ij} \ne 0), i \ne j; \lambda_{ik} + \mu_{ik} \ge 1, i \ne k; \lambda_{kk} + \mu_{kk} \ge 0. \tag{A.12}$$

In particular, if (A.9) holds, then there exists a $\{\tilde\lambda_{ik}, \tilde\mu_{jk} : 1 \le i, j \le p\}$ at which the min function $\min_{(\lambda_{ik},\mu_{jk})}(\cdot)$ there, then (A.11) is met. Conversely, if there exists $\{(\tilde\lambda_{ik}, \tilde\mu_{jk}) : 1 \le i, j \le p\}$ such that (A.11) holds, then $p - 1 \ge \sum_{i=1}^p \tilde\lambda_{ik} + \sum_{j=1}^p \tilde\mu_{jk} \ge$

$\min_{(\lambda_{ik},\mu_{jk})} \sum_{i=1}^{p} \lambda_{ik} + \sum_{j=1}^{p} \mu_{jk}$, implying thus (A.9) holds.

Finally, taking a summation of (A.11) over $k = 1, \cdots, p$, we have

$$\begin{aligned} p(p-1) &\geq \sum_{k=1}^{p} \left(\sum_{i=1}^{p} \lambda_{ik} + \sum_{j=1}^{p} \mu_{jk} \right) \\ &= \sum_{k=1}^{p} \left(\sum_{i \neq k} (\lambda_{ik} + \mu_{ik}) + (\lambda_{kk} + \mu_{kk}) \right) \geq \sum_{k=1}^{p} ((p-1) + 0) = p(p-1). \end{aligned}$$

The last one inequality uses the fact that $\lambda_{ik} + \mu_{ik} \geq 1$; $i \neq k$, and $\lambda_{kk} + \mu_{kk} \geq 0$, from (A.12). Consequently, these inequalities become equalities. Therefore, $\mu_{ik} = 1 - \lambda_{ik}, i \neq k$ and $\mu_{kk} = -\lambda_{kk}$. Replacing μ_{ik} by $1 - \lambda_{ik}$ and μ_{kk} by $-\lambda_{kk}$; $i \neq k$; $k = 1, \cdots, p$, in (A.12) yields (2.2). This completes the proof. □

Proof of proposition 1: To prove convergence of Algorithm 1, we analyze the DC and ADMM components separately. Given strong convexity of $l(\boldsymbol{A})$ in its arguments, the ADMM converges [38]. For the DC component, note that the Karush-Kuhn-Tucker conditions imply that there exists a vector of nonnegative Lagrange multipliers $\nu \geq 0$ and $\{\zeta_{ijk} \geq 0\}_{i,j,k=1,\ldots,p,j\neq i}$, such that $(\hat{\boldsymbol{A}}^{(m^*)}, \hat{\boldsymbol{\lambda}}^{(m^*)})$ minimizes the Lagrange function, where m^* is the iteration index at termination:

$$\begin{aligned} \bar{S}(\boldsymbol{A}, \boldsymbol{\lambda}) &= l(\boldsymbol{A}) + \nu \left(\sum_{1 \leq j \neq k \leq p} \mathrm{J}_{\tau}(A_{jk}) - K \right) \\ &+ \sum_{i,j,k=1,\ldots,p,j\neq i} \zeta_{ijk} \left(\mathrm{J}_{\tau}(A_{ij}) - \lambda_{ik} - I(j \neq k) + \lambda_{jk} \right) \end{aligned}$$

with respect to $\boldsymbol{A}$.

For the solution of (2.10) subject to (2.11), and (2.12), note that $0 \leq \bar{S}(\hat{\boldsymbol{A}}^{(m)}, \hat{\boldsymbol{\lambda}}^{(m)}) = \bar{S}^{(m+1)}(\hat{\boldsymbol{A}}^{(m)}, \hat{\boldsymbol{\lambda}}^{(m)}) \leq \bar{S}^{(m)}(\hat{\boldsymbol{A}}^{(m)}, \hat{\boldsymbol{\lambda}}^{(m)}) \leq \bar{S}^{(m)}(\hat{\boldsymbol{A}}^{(m-1)}, \hat{\boldsymbol{\lambda}}^{(m-1)}) = \bar{S}(\hat{\boldsymbol{A}}^{(m-1)}, \hat{\boldsymbol{\lambda}}^{(m-1)})$, where m is the DC iteration index and $\bar{S}^{(m)}(\cdot)$ is the DC cost function value at iteration m. By nonincreasingness, $\lim_{m\to\infty} \bar{S}(\hat{\boldsymbol{A}}^{(m)}, \hat{\boldsymbol{\lambda}}^{(m)})$. Finite step convergence follows from strict decreasing-ness of $\bar{S}^{(m)}(\hat{\boldsymbol{A}}^{(m)}, \hat{\boldsymbol{\lambda}}^{(m)})$ in m and finite possible values of the subgradient of the trailing convex function. At termination $\bar{S}(\hat{\boldsymbol{A}}^{(m^*)}, \hat{\boldsymbol{\lambda}}^{(m^*)}) = \bar{S}(\hat{\boldsymbol{A}}^{(m^*-1)}, \hat{\boldsymbol{\lambda}}^{(m^*-1)})$; otherwise the iteration continues. It can be verified that $(\hat{\boldsymbol{A}}^{(m^*)}, \hat{\boldsymbol{\lambda}}^{(m^*)})$ satisfies the local optimality in (2.18), implying that the DAG requirement for

$\boldsymbol{A}$ is met when $\tau > 0$ is sufficiently small, leading to an estimated DAG. Readers may consult [62] for this kind of arguments. This completes the proof.

Proof of Theorem 2: Before proceeding, we define a complexity measure for the size of a space $\mathcal{F}$. The bracketing Hellinger metric entropy of $\mathcal{F}$, denoted by the function $H(\cdot, \mathcal{F})$, is defined by logarithm of the cardinality of the u-bracketing (of $\mathcal{F}$) of the smallest size. That is, for a bracket covering $S(\varepsilon, m) = \{f_1^l, f_1^u, \cdots, f_m^l, f_m^u\} \subset \mathcal{L}_2$ satisfying $\max_{1 \leq j \leq m} \|f_j^u - f_j^l\|_2 \leq \varepsilon$ and for any $f \in \mathcal{F}$, there exists a j such that $f_j^l \leq f \leq f_j^u$, *a.e.* P, then $H(u, \mathcal{F})$ is $\log(min\{m : \ S(u, m)\})$, where $\|f\|_2 = \int f^2(z) d\mu$. For more discussions about metric entropy of this type, see [63].

There are $p(p-1)$ parameters in $\boldsymbol{A}$ since the diagonal elements are set to 0. If $K = |E^0|$, $|\hat{E}^{L_0}| \leq |E^0|$. If $\hat{E}^{L_0} = E^0$, then $\hat{\boldsymbol{A}}^{L_0} = \hat{\boldsymbol{A}}^{OR}$. Therefore, it suffices to prove the case of $\hat{E}^{L_0} \neq E^0$. Let $\{E : E \neq E^0, |E| \leq |E^0|\}$ be a class of candidate subsets consisting of nonzeros of $\boldsymbol{A}$.

We define a $p \times p$ matrix $\boldsymbol{D}$ to be a diagonal matrix with diagonal elements being the error variances. In the equal-variance case, the diagonal elements are all σ^2. Now define $\boldsymbol{\Omega}_E = (\boldsymbol{I} - \boldsymbol{A}_E)^T \boldsymbol{D}^{-1} (\boldsymbol{I} - \boldsymbol{A}_E)$ for for any $E \subset \{(i,j) | 1 \leq i \neq j \leq p\}$. Now partition E into $(E \setminus E^0) \cup (E^0 \cap E) = E$. Let $B_{kj} = \{\boldsymbol{\Omega}_E : E \neq E^0, |E^0 \cap E| = k, |E \setminus E^0| = j, (|E^0| - k) C_{\min}(\boldsymbol{\Omega}^0) \leq h^2(\boldsymbol{\Omega}, \boldsymbol{\Omega}^0)\} \subset \mathcal{F}$; $k = 0, \ldots, |E^0| - 1$, $j = 1, \ldots, |E^0| - k$, where $C_{\min}(\boldsymbol{\Omega}^0) = \inf_{\{\boldsymbol{\Omega}_E : \boldsymbol{\Omega}_E \neq \boldsymbol{\Omega}^{\boldsymbol{0}}, |E| \leq |E^0|\}} \frac{-\log(1 - h^2(\boldsymbol{\Omega}_E, \boldsymbol{\Omega}^0))}{\max(|E^0 \setminus E|, 1)}$, and $h^2(\boldsymbol{\Omega}, \boldsymbol{\Omega}^0) = 1 - \sqrt{\frac{(det(\boldsymbol{\Omega}) det(\boldsymbol{\Omega}^0))^{1/2}}{\det(\frac{\boldsymbol{\Omega} + \boldsymbol{\Omega}^0}{2})}}$ is the Hellinger-distance between $\boldsymbol{\Omega}$ and $\boldsymbol{\Omega}^0$ under (2.4). Note that B_{kj} consists of $\binom{|E^0|}{k} \binom{p(p-1) - |E^0|}{j}$ different elements E's of sizes $|E^0 \cap E| = k$ and $|E \setminus E^0| = j$. By definition, $\{\boldsymbol{\Omega} = \boldsymbol{\Omega}_E = (\boldsymbol{I} - \boldsymbol{A}_E)^T \boldsymbol{D}^{-1} (\boldsymbol{I} - \boldsymbol{A}_E) : E \neq E^0, C_{\min}(\boldsymbol{\Omega}^0) \leq h^2(\boldsymbol{\Omega}, \boldsymbol{\Omega}^0), |E| \leq |E^0|\} \subset \cup_{k=0}^{|E^0|-1} \cup_{j=1}^{|E^0|} B_{kj}$. Let $L(\boldsymbol{\Omega}) = -\log f(\boldsymbol{\Omega}, \boldsymbol{x})$, where $\log f(\boldsymbol{\Omega}, \boldsymbol{x}) = \sum_{j=1}^p \left(\frac{1}{2\sigma^2} \sum_{i=1}^n \left(x_{ij} - \sum_{k \neq j} x_{ik} A_{jk} \right)^2 + \frac{n}{2} \log \sigma^2 \right)$, $\varepsilon_{n,p,|E^0|} = \min(1, (2c_0)^{1/2} c_4^{-1} \log(2^{1/2}/c_3) \log p (\frac{|E^0|}{n})^{1/2})$. Assume, without loss of generality, that $|E^0| > 1$. Then

$$
\begin{aligned}
& P(\hat{\boldsymbol{\Omega}}^{L_0} \neq \hat{\boldsymbol{\Omega}}^{OR}) \leq P^* \Big(\sup_{\boldsymbol{\Omega}_E : E \neq E^0, |E| \leq |E^0|} (L(\boldsymbol{\Omega}_E) - L(\hat{\boldsymbol{\Omega}}^{OR})) \geq 0 \Big) \\
\leq \;& P^* \Big(\sup_{\boldsymbol{\Omega}_E : E \neq E^0, |E| \leq |E^0|} (L(\boldsymbol{\Omega}_E) - L(\boldsymbol{\Omega}^0) \geq 0 \Big) \\
\leq \;& \sum_{E \subset \{(i,j) | 1 \leq i \neq j \leq p\} : |E| \leq |E^0|} P^* \Big(\sup_{\boldsymbol{\Omega}_E \in B_{ij}} (L(\boldsymbol{\Omega}_E) - L(\boldsymbol{\Omega}^0)) 0 \geq \Big) \equiv I,
\end{aligned}
$$

where P^* is the outer measure.

For I, we apply Theorem 1 of [64] to bound each term. Towards this end, we verify the entropy condition (3.1) there for the bracketing entropy over B_{ij}. Let $\mathcal{F}_{ij} = \{f(\boldsymbol{\Omega}, \cdot) : \boldsymbol{\Omega} = (\boldsymbol{I} - \boldsymbol{A})^T \boldsymbol{D}^{-1}(\boldsymbol{I} - \boldsymbol{A}) \in B_{ij}\}$, where $f(\cdot;\cdot)$ is the probability density. For $j = 1, \ldots, p$, $\Omega_{jj} = \frac{1}{\sigma^2} + \sum_{k \neq j} \frac{1}{\sigma^2} A_{jk}^2$; implying that $\frac{1}{\sigma}|A_{kj}| \leq M_2^{1/2}$. Moreover, $\det(\boldsymbol{\Omega})$ is bounded away from zero because $c_{\min}(\boldsymbol{\Omega}) > M_1$. For any $|E| \leq |E^0|$ and some constant $c^{'} > 0$,

$$\begin{aligned} &\int \sup_{\{\tilde{\boldsymbol{\Omega}} \in B_\delta(\boldsymbol{\Omega})\}} (f^{1/2}(\tilde{\boldsymbol{\Omega}}, \boldsymbol{x}) - f^{1/2}(\boldsymbol{\Omega}, \boldsymbol{x}))^2 d\mu \\ \leq & \sup_{\{\tilde{\boldsymbol{\Omega}} \in B_\delta(\boldsymbol{\Omega})\}} c^{'} \|\tilde{\boldsymbol{D}}^{-1/2} \tilde{\boldsymbol{A}} - \boldsymbol{D}^{-1/2} \boldsymbol{A})\|_{F^*}^2, \end{aligned}$$

where $\| \cdot \|_{F*}$ and $\| \cdot \|_{F*}$ are the Frobenius-norm and the L_2-norm whose individual element is taken over $\sup_{(\tilde{\boldsymbol{\Omega}}) \in B_\delta(\boldsymbol{\Omega})}$. Note that the jkth element of $\tilde{D}^{-1/2} \tilde{A}$ is $\tilde{A}_{jk}/\sigma$, which is bounded by $M_2^{1/2}$. By Lemma 1 of [65], it suffices to bound the entropy of $B_\delta(\boldsymbol{\Omega})$. Note that there are $|E|$ nonzero entries of $\boldsymbol{A}$ with $\binom{p(p-1)}{|E|}$ possible locations. By [63], for $u \geq \varepsilon_{0,|\Omega|}^2$,

$$\begin{aligned} H(u, \mathcal{F}_{ij}) &\leq c_0 (\log \binom{p(p-1)}{|E|} + |E| \log (\frac{\min(M_2^{1/2}, 1)}{u})) \\ &\leq c_0 (|E| \log \left(e \frac{p(p-1)}{|E|} \right) + |E| \log (\frac{\min(M_2^{1/2}, 1)}{u})), \end{aligned} \quad (A.13)$$

where $H_2(\cdot, \mathcal{F}_{ij})$ is the ℓ_2-metric entropy $\mathcal{F}_{ij}$ and inequality $\binom{n}{m} \leq \left(e \frac{n}{m}\right)^m$ has been used, c.f., Theorem 2.6 of [66]. Hence $H(u, \mathcal{F}_{ij}) \leq c_0 (|E| \log p \log(\frac{1}{u}))$. Then $\varepsilon = \varepsilon_{n,p,|E^0|}$ satisfies

$$\sup_{\{0 \leq |E| \leq |E^0|\}} \int_{2^{-8}\varepsilon^2}^{2^{1/2}\varepsilon} H^{1/2}(t/c_3, \mathcal{F}_{ij}) dt \leq |E^0|^{1/2} 2^{1/2} \varepsilon \log(2/2^{1/2} c_3) \leq c_4 n^{1/2} \varepsilon^2. \quad (A.14)$$

for some constant $c_3 > 0$ and c_4, say $c_3 = 10$ and $c_4 = \frac{(2/3)^{5/2}}{512}$. By Assumption B, $C_{\min}(\boldsymbol{\Omega}^0) \geq \varepsilon_{n,p,|E^0|}^2$. Moreover, by Theorem 2.6 of [66], $\binom{b}{a} \leq \frac{b^{b+1/2}}{\sqrt{2\pi} a^{a+1/2} (b-a)^{b-a+1/2}} \leq \exp\big((a+1/2) \log(b/a) + a\big)$ for any integers $a < b$. Note that $C_{\min}(\boldsymbol{\Omega}^0) \geq \varepsilon_{n,|E^0|,p}^2$ implies (A.14), provided that $2C_2^{-1} > (2c_0)^{1/2} c_4^{-1} \log(2^{1/2}/c_3)$. Using the facts about binomial coefficients: $\sum_{j=0}^{|E^0|-k} \binom{p(p-1)-|E^0|}{j} \leq (p(p-1) - |E^0| + 1)^{|E^0|-k}$ and $\binom{|E^0|}{i} \leq |E^0|^i$, we obtain, by Theorem 1 of [64], that for a constant $c_2 > 0$, say $c_2 = \frac{4}{27} \frac{1}{1926}$. Then

$$\begin{aligned} I &= \sum_{E\subset\{(i,j)|1\leq i\neq j\leq p\}:E\neq E^0,|E|\leq|E^0|} P^*\Big(\sup_{\boldsymbol{\Omega}_E\in B_{ij}}(L(\boldsymbol{\Omega}_E)-L(\boldsymbol{\Omega}^0))\geq 0\Big) \\ &\leq \sum_{k=0}^{|E^0|-1}\sum_{j=0}^{|E^0|-k} P^*\big(\sup_{\boldsymbol{\Omega}_E\in B_{kj}}(L(\boldsymbol{\Omega}_E)-L(\boldsymbol{\Omega}^0))\geq 0\big) \\ &\leq 4\sum_{k=0}^{|E^0|-1}\binom{|E^0|}{k}\exp(-c_2n(|E^0|-k)C_{\min}(\boldsymbol{\Omega}^0))\sum_{j=0}^{|E^0|-k}\binom{p(p-1)-|E^0|}{j} \\ &\leq 4\sum_{i=1}^{|E^0|}\exp\Big(-i\big(c_2nC_{\min}(\boldsymbol{\Omega}^0)-\log(p(p-1)-|E^0|+1)-\log|E^0|\big)\Big) \\ &\leq R\Big(\exp\big(-\big(c_2nC_{\min}(\boldsymbol{\Omega}^0)-\log(p(p-1)-|E^0|+1)-\log|E^0|\big)\big)\Big), \end{aligned}$$

where $R(x)=x/(1-x)$ is the exponentiated logistic function. Note, moreover, that $I\leq 1$ and $\log\big(p(p-1)-|E^0|+1\big)+\log|E^0|\leq 2\log\frac{p(p-1)+1}{2}$.

$$\begin{aligned} I &\leq 5\exp\big(-c_2nC_{\min}(\boldsymbol{\Omega}^0)+2\log\frac{p(p-1)+1}{2}\big) \\ &\leq \exp\big(-c_2nC_{\min}(\boldsymbol{\Omega}^0)+2\log(p(p-1)+1)+1\big). \end{aligned}$$

Consequently, under Assumption B or (2.19), that $P\Big(\hat{\boldsymbol{\Omega}}^{L_0}\neq\hat{\boldsymbol{\Omega}}^{OR}\Big)\to 0$ as $n,p,|E^0|\to\infty$, implying consistent reconstruction. For parameter estimation,

$$Eh^2(\hat{\boldsymbol{\Omega}}^{L_0},\boldsymbol{\Omega}^0)=Eh^2(\hat{\boldsymbol{\Omega}}^{OR},\boldsymbol{\Omega}^0)I(\hat{\boldsymbol{\Omega}}^{L_0}=\hat{\boldsymbol{\Omega}}^{OR})+P(\hat{\boldsymbol{\Omega}}^{L_0}\neq\hat{\boldsymbol{\Omega}}^{OR})\leq(1+o(1))Eh^2(\hat{\boldsymbol{\Omega}}^{OR},\boldsymbol{\Omega}^0).$$

This completes the proof. □

Proof of Theorem 3: The proof is basically the same as that in Theorem 2 with a modification that E is replaced by $E^\tau\equiv\{(i,j):|A_{i,j}|\geq\tau)$. Note that $h^2(\boldsymbol{\Omega},\boldsymbol{\Omega}^0)=1-\sqrt{\frac{(det(\boldsymbol{\Omega})det(\boldsymbol{\Omega}^0))^{1/2}}{det(\frac{\boldsymbol{\Omega}+\boldsymbol{\Omega}^0}{2})}}$. Moreover, for $j\neq k=1,\cdots,p$, for any $\boldsymbol{\theta}\in\mathcal{F}$,

$$\Big|\frac{\partial h^2(\boldsymbol{\Omega},\boldsymbol{\Omega}^0)}{\partial\Omega_{jk}}\Big|=\frac{1}{4}\Big|\big(1-h^2(\boldsymbol{\Omega},\boldsymbol{\Omega}^0)\big)tr\big(\big(2(\frac{\boldsymbol{\Omega}+\boldsymbol{\Omega}^0}{2})^{-1}-\boldsymbol{\Omega}^{-1}\big)\boldsymbol{\Delta}_{jk}\big)\Big|,$$

which is upper bounded by $\Big|\frac{1}{c_{\min}\big(\boldsymbol{\Omega}\big)+c_{\min}\big(\boldsymbol{\Omega}^0\big)}+\frac{1}{4c_{\min}\big(\boldsymbol{\Omega}\big)}\Big|\leq\frac{2}{M_1}$; $j\neq k=1,\cdots,p$. Let $\boldsymbol{\theta}_\tau=(\boldsymbol{A}\cdot I(|\boldsymbol{A}|\geq\tau),\boldsymbol{\phi})$ and $E^\tau\equiv\{(i,j):|A_{i,j}|\geq\tau)$. Then $|h^2(\boldsymbol{\Omega},\boldsymbol{\Omega}^0)-h^2(\boldsymbol{\Omega}_\tau,\boldsymbol{\Omega}^0)|=\tau\Big|\sum_{j\in A^{\tau^-}}\frac{\partial h^2(\boldsymbol{\Omega},\boldsymbol{\Omega}^0)}{\partial\Omega_{jk}}\Big|_{\boldsymbol{\Omega}=\boldsymbol{\Omega}^\star}\Big|$. This implies $|h^2(\boldsymbol{\Omega},\boldsymbol{\Omega}^0)-h^2(\boldsymbol{\Omega}_\tau,\boldsymbol{\Omega}^0)|\leq 2\tau p(p-1)/M_1$.

Now $B_{kj}=\{\boldsymbol{\theta}_{\tau^+}:E^\tau\neq E^0,|E^0\cap E^\tau|=k,|E^\tau\setminus E^0|=j,(d_1(|E^0|-k)C_{\min}(\boldsymbol{\Omega}^0)-d_3q\tau^{d_2})\leq h^2(\boldsymbol{\Omega}_\tau,\boldsymbol{\Omega}^0)\}$; $j=1,\ldots,|E^0|$. Then $\{\boldsymbol{\Omega}=(\boldsymbol{I}-\boldsymbol{A})^T\boldsymbol{D}^{-1}(\boldsymbol{I}-\boldsymbol{A}):E^\tau\neq$

$E^0, |E^\tau| \le |E^0|, C_{\min}(\boldsymbol{\Omega}^0) \le h^2(\boldsymbol{\Omega}, \boldsymbol{\Omega}^0)\} \subset \cup_{k=0}^{|E^0|-1} \cup_{j=0}^{|E^0|-k} B_{kj}$.

When $K = |E^0|$, $\sum_{1 \le j \ne i \le p} \mathrm{J}_\tau(A_{ij}) \le |E^0|$, implying that $|\hat{E}^\tau| \le |E^0|$. If $\hat{E}^\tau = E^0$, then $\sum_{1 \le j \ne i \le p} |A_{ij}| I(|A_{ij}| \le \tau) = 0$, implying that $\hat{\boldsymbol{\theta}}^T = \hat{\boldsymbol{\theta}}^{OR}$. Then we focus our attention to the case of $E^{\tau+} \ne E^0$.

$$\begin{aligned} & P^*\Big(\sup_{\boldsymbol{\Omega}_{E^\tau}: E^\tau \ne E^0, |E^\tau| \le |E^0|} \big(L(\boldsymbol{\Omega}_{E^\tau}) - L(\boldsymbol{\Omega}^0)\big) \ge 0 \Big) \\ \le \;& \sum_{k=0}^{|E^0|-1} \sum_{j=0}^{|E^0|-k} P^*\Big(\sup_{\boldsymbol{\Omega}_{E^\tau} \in B_{kj}} \big(L(\boldsymbol{\Omega}_{E^\tau})) - L(\boldsymbol{\Omega}^0)\big) \ge 0 \Big) \\ \le \;& 4 \sum_{k=0}^{|E^0|-1} \sum_{j=0}^{|E^0|-k} \binom{p(p-1) - |E^0|}{j} \binom{|E^0|}{k} \exp(-c_2 n (d_1 C_{\min}(\boldsymbol{\Omega}^0) - d_3 q \tau^{d_2})) \\ \le \;& 5 \exp\big(-(c_2 d_1/2) n C_{\min}(\boldsymbol{\Omega}^0) + 2 \log \frac{p(p-1)+1}{2} \big) \\ \le \;& \exp\big(-c_2 n C_{\min}(\boldsymbol{\Omega}^0) + 2\log(p(p-1)+1) + 3 \big), \end{aligned}$$

provided that $\tau \le (\gamma_{\min}^2 c_{\min}(H) M_1 / 4q)$ by Proportion 2 of [7]. The rest of the proof proceeds as in the proof of Theorem 2. This completes the proof. □

A.2 Technical Proofs for Chapter 3

Proof of Theorem 1: Let $\boldsymbol{\beta}^m$ be the solution at iteration m. Convergence follows directly from the fact that $f(\boldsymbol{\beta}^m)$ is decreasing in m. To prove the limit of β^* satisfies the subdifferential condition, that is, there exists $\lambda > 0$, s.t. $-\mathrm{sign}(\beta_j) \frac{\partial f(\beta)}{\partial \beta_j} = \lambda$ for $\beta_j \ne 0$, and $\left| \frac{\partial f(\beta)}{\partial \beta_j} \right| < \lambda$ for $\beta_j = 0$, suppose the subdifferential condition does not hold for the convergence point β^*. Let $i = \mathrm{argmax}_{k \in 1,\ldots,p} |\frac{\partial f(\beta^*)}{\partial \beta_k}|$, $j = \mathrm{argmin}_{\{k | \beta_k^* \ne 0\}} |\frac{\partial f(\beta^*)}{\partial \beta_k}|$. Then $|\frac{\partial f(\beta^*)}{\partial \beta_i}| > |\frac{\partial f(\beta^*)}{\partial \beta_j}|$. This implies that we can further reduce the value of the objection function, which contradicts to convergence. From [67], this subdifferential condition is sufficient and necessary for β^* to be the minimizer of the Lasso. This completes the proof.