

University of California  
Santa Barbara

# Advancing Practicality and Scalability in High-Dimensional Graphical Models

A dissertation submitted in partial satisfaction  
of the requirements for the degree

Doctor of Philosophy  
in  
Statistics and Applied Probability

by

Joshua Bang

Committee in charge:

Professor Sang-Yun Oh, Chair  
Professor Tomoyuki Ichiba  
Professor Yuedong Wang

March 2024

The Dissertation of Joshua Bang is approved.

---

Professor Tomoyuki Ichiba

---

Professor Yuedong Wang

---

Professor Sang-Yun Oh, Committee Chair

December 2023

Advancing Practicality and Scalability in High-Dimensional Graphical Models

Copyright © 2024

by

Joshua Bang

To my parents, Younggil Bang and Myoungsoown Heo

## Acknowledgements

I am grateful to have Professor Sang-Yun Oh as my research advisor. I truly enjoyed working with him throughout my studies at UCSB, and I hope he feels the same way.

I thank the faculty members of the Department of Statistics and Applied Probability. Specifically, I thank my committee members, Professor Tomoyuki Ichiba and Professor Yuedong Wang, for their guidance and support. I also thank Professor S. Rao Jammalamadaka, Professor Wendy Meiring, Professor Raya Feldman for their teaching in the early years of my studies.

I thank my collaborators overseas: Professor Joong-Ho Won at Seoul National University and Professor Hyungwon Choi, Dr. Sungdong Lee, Dr. Youngra Kim at the National University of Singapore.

Lastly, I would also like to thank Dr. Youhong Lee and Dr. Chao Zhang, my colleagues and friends who have always been willing to discuss and provide their insights on various research topics.

# Curriculum Vitæ

Joshua Bang

## Education

|      |                                                                                       |
|------|---------------------------------------------------------------------------------------|
| 2023 | Ph.D. in Statistics and Applied Probability, University of California, Santa Barbara. |
| 2019 | M.A. in Statistics, University of California, Santa Barbara.                          |
| 2017 | B.S. in Statistics, University of California, Santa Barbara.                          |

## Professional Experience

|              |                                                    |
|--------------|----------------------------------------------------|
| Summer, 2022 | <i>Data Science Intern</i> , Electronic Arts (EA). |
| Summer, 2017 | <i>Data Science Intern</i> , Farmers Insurance.    |
| Summer, 2016 | <i>Market Risk Intern</i> , AIG.                   |

## Publications

**Bang, J.**, Oh, S.-Y. (2023). FROSTY: A High-Dimensional Scale-Free Bayesian Network Learning Method. *Journal of Data Science*, 21(2): 354-367.

## Scholarships and Awards

|      |                                |
|------|--------------------------------|
| 2021 | KSEA-KUSCO Scholarship.        |
| 2017 | Rama Thogarati Memorial Award. |

## Abstract

Advancing Practicality and Scalability in High-Dimensional Graphical Models

by

Joshua Bang

Graphical models, which utilize graphs to encode conditional independencies between random variables, have proven to be versatile tools for encoding multivariate distributions. In addressing some practical considerations for using graphical models in applications, we developed two graphical model methods. Firstly, we develop the FROSTY method for directed graphical models focusing on scalability that performs well in simulation. FROSTY only requires observational data and user-specified confidence level as inputs and can estimate networks with thousands of variables. Notably, FROSTY does not require computationally expensive cross-validation for model selection. We learned that initial input and parameter tuning are important in estimation, and empirical evidence illustrates that other methods can be improved by adopting FROSTY’s initial step. Secondly, we develop the ACCORD method for undirected graphical models based on pseudo-likelihood. For optimization, we propose a novel objective splitting that ensures linear convergence with a wide range of step size in both the objective values and the iterates despite the lack of strong convexity in ACCORD’s objective function. Additionally, we implement a high-performance variant of ACCORD, named HP-ACCORD, that can scale up to 1 million variables utilizing communication-avoiding linear algebra on distributed computing environments.

# Contents

|                                                                                           |            |
|-------------------------------------------------------------------------------------------|------------|
| <b>Curriculum Vitae</b>                                                                   | <b>vi</b>  |
| <b>Abstract</b>                                                                           | <b>vii</b> |
| <b>1 Introduction</b>                                                                     | <b>1</b>   |
| 1.1 Graphical Models . . . . .                                                            | 1          |
| 1.2 Gaussian Undirected and Directed Graphs . . . . .                                     | 2          |
| <b>2 FROSTY for Bayesian Network Structure Learning</b>                                   | <b>4</b>   |
| 2.1 Background . . . . .                                                                  | 5          |
| 2.2 Preliminaries . . . . .                                                               | 9          |
| 2.3 Distributionally Robust Optimization for Cholesky Factor . . . . .                    | 12         |
| 2.4 Recovering Variable Ordering . . . . .                                                | 16         |
| 2.5 Computational Complexity . . . . .                                                    | 17         |
| 2.6 Numerical Experiments . . . . .                                                       | 17         |
| 2.7 Discussion . . . . .                                                                  | 23         |
| <b>3 Applications of FROSTY</b>                                                           | <b>25</b>  |
| 3.1 Real Network Analysis with FROSTY . . . . .                                           | 25         |
| 3.2 Missing Data Analysis with FROSTY . . . . .                                           | 28         |
| <b>4 ACCORD for Sparse Precision Matrix Estimation</b>                                    | <b>35</b>  |
| 4.1 Background . . . . .                                                                  | 36         |
| 4.2 Pseudo-Likelihood and CONCORD . . . . .                                               | 37         |
| 4.3 ACCORD, the Asymmetric CONCORD . . . . .                                              | 39         |
| 4.4 Optimization for ACCORD . . . . .                                                     | 41         |
| 4.5 HP-ACCORD, ACCORD with High-Performance Computing . . . . .                           | 46         |
| 4.6 Numerical Experiments . . . . .                                                       | 47         |
| 4.7 Cast Study: Integrated Network Analysis of Multi-Omics Data in Liver Cancer . . . . . | 59         |
| 4.8 Discussion . . . . .                                                                  | 68         |

|          |                                             |           |
|----------|---------------------------------------------|-----------|
| <b>A</b> | <b>Appendix for FROSTY</b>                  | <b>70</b> |
| A.1      | Proofs . . . . .                            | 70        |
| A.2      | Additional Numerical Results . . . . .      | 75        |
| A.3      | Software Package . . . . .                  | 78        |
| <b>B</b> | <b>Appendix for Applications of FROSTY</b>  | <b>80</b> |
| B.1      | Missing Data Generation under MAR . . . . . | 80        |
| <b>C</b> | <b>Appendix for ACCORD</b>                  | <b>82</b> |
| C.1      | Proofs . . . . .                            | 82        |
| C.2      | Software Package . . . . .                  | 87        |
|          | <b>Bibliography</b>                         | <b>89</b> |

# Chapter 1

## Introduction

### 1.1 Graphical Models

Graphical models are statistical models that utilize graphs to represent complex probability distributions. In these models, the random variables are represented by the vertices of the graph, and their conditional dependencies are captured by the edges. Both undirected graphs and directed acyclic graphs (DAGs) can be decomposed into smaller configurations based on their respective graph structures. In the case of undirected graphs, cliques are commonly employed as configurations, which are subgraphs where every pair of vertices is connected. On the other hand, DAGs utilize conditional distributions as configurations. Consider a collection of random variables, denoted as  $X_1, \dots, X_p$ , along with a graph  $G$  that encodes the conditional dependencies between them. The joint distribution, denoted as  $P(X_1, \dots, X_p)$ , can be expressed as follows.

1. Undirected graphical models

$$P(X_1, \dots, X_p) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \psi_c(X_c)$$

where  $\mathcal{C}$  is a set of maximal cliques in a graph  $G$  and  $Z$  is a normalizing constant.  $\psi_c(\cdot)$  is called a *compatibility function* or *potential function* for a clique  $c$ , which defines the relationship between neighboring vertices in  $c$ .

## 2. Directed acyclic graphical models

$$P(X_1, \dots, X_p) = \prod_{i=1}^p P(X_i \mid X_{pa_G(X_i)}).$$

where  $pa_G(X_i)$  is a set of parents of variable  $X_i$ .

In contrast to an undirected graph, a directed acyclic graph (DAG) inherently gives rise to a linear ordering of variables due to the asymmetry of its configurations, which are represented by the conditional distributions. This ordering is commonly referred to as a *topological sort*, where vertex  $i$  precedes vertex  $j$  for every directed edge  $i \rightarrow j$ . It is important to note that a topological sort is not necessarily unique in general, allowing for multiple valid orderings within a given DAG.

## 1.2 Gaussian Undirected and Directed Graphs

In particular, we will focus on Gaussian graphical models, which make up a large proportion of the graphical model literature due to various interesting properties of the Gaussian distribution. For example, not only is the Gaussian distribution closed under marginalization and conditioning, which leads to mathematical simplicity, but it also has the maximum entropy for a specified mean and covariance given that many physical systems tend to move towards maximal entropy over time [1].

The multivariate Gaussian log-likelihood is given by

$$\ell(S; \Theta) = \log |\Theta| - \text{trace}(S\Theta) \tag{1.1}$$

where  $S$  and  $\Theta$  are the sample covariance matrix and the precision matrix, respectively. It is well known that the zero/nonzero entries of the precision matrix correspond to zero/nonzero partial correlations, which implies conditional independence/dependence between random variables. Hence, estimating an undirected graph under multivariate Gaussianity boils down to estimating the precision matrix.

Furthermore, if we order variables according to a reversed topological sort, the adjacency matrix of a DAG can be expressed as a lower triangular matrix, which in fact corresponds to the Cholesky factor of the precision matrix. Let  $\pi$  be a reversed topological sort and  $P_\pi$  be a permutation matrix according to  $\pi$  so that  $P_\pi \Theta P_\pi^T = \Theta_\pi$  is a reverse-topologically sorted precision matrix. Then,  $\Theta_\pi = L_\pi L_\pi^T$  where  $L_\pi$  is the Cholesky factor of  $\Theta_\pi$  whose strictly lower triangular portion represents directed edges.

Note that whether or not knowing a variable ordering apriori does not affect the precision matrix estimation since the multivariate Gaussian log-likelihood in (1.1) is invariant to a variable ordering.

$$\begin{aligned} \ell(S_\pi; \Theta_\pi) &= \log |\Theta_\pi| - \text{trace}(S_\pi \Theta_\pi) \\ &= \log |P_\pi \Theta P_\pi^T| - \text{trace}(P_\pi S P_\pi^T P_\pi \Theta P_\pi^T) \\ &= \log |\Theta| - \text{trace}(S \Theta) = \ell(S; \Theta) \end{aligned} \tag{1.2}$$

where  $S_\pi = P_\pi S P_\pi^T$  and the global maximizer of  $\Theta_\pi$  is simply a permuted inverse sample covariance matrix according to  $\pi$ , i.e.,  $\Theta_\pi^* = S_\pi^{-1} = P_\pi S^{-1} P_\pi^T$ . However, this is not the case if we were to estimate the DAG structure,  $L_\pi$ , because the Cholesky factor is *not* invariant to a variable ordering. In other words,  $L_\pi$  cannot be expressed as a permuted version of the Cholesky factor of an arbitrarily ordered precision matrix. Considering this, we develop a scalable Bayesian network structure learning method in the following chapter.

## Chapter 2

# FROSTY for Bayesian Network Structure Learning

In this chapter, we propose a scalable Bayesian network learning algorithm based on sparse Cholesky decomposition, which we call *scale-Free bayesian network learning with RObust Selection Tuning and sparse choleskY* (FROSTY). Our approach only requires observational data and user-specified confidence level as inputs and can estimate networks with thousands of variables. The computational complexity of the proposed method is  $O(p^3)$  for a graph with  $p$  vertices. Extensive numerical experiments illustrate the usefulness of our method with promising results. In simulation, the initial step in our approach also improves an alternative Bayesian network structure estimation method that uses an undirected graph as an input.

## 2.1 Background

### 2.1.1 Bayesian Networks

Bayesian network encodes conditional independencies between random variables as directed acyclic graphs (DAGs) [2]. Consider a multivariate distribution formed by  $p$ -variate random vector  $X = (X_1, \dots, X_p)^T$ . Then, a Bayesian network can be defined by a pair  $\{G, X\}$  where  $G$  is a DAG that imposes a causal structure over  $X$ .  $G$  is formed by two sets  $(V, E)$  where  $V = \{X_1, \dots, X_p\}$  or simply  $\{1, \dots, p\}$  is a set of vertices and  $E = \{(i, j) : i \rightarrow j \forall i, j \in V\}$  is a set of directed edges.  $pa_G(j) = \{i \in V : i \rightarrow j\}$  denotes the parent set of vertex  $j$  in the DAG  $G$ . Bayesian networks allow decomposition of the joint distribution into smaller configurations (conditional distributions) such that the product of all configurations amounts to the full joint distribution

$$P(X) = \prod_{i=1}^p P(X_i \mid X_{pa_G(X_i)}). \quad (2.1)$$

Not only does the decomposition compactly represent complex relationships, it also provides computational efficiency in sampling and inference that would otherwise require intractable summations [3].

**Definition 1** (Markov Compatibility [2, 4]). *If a probability distribution  $P$  admits the decomposition in (2.1) relative to a DAG  $G$ , we say that  $G$  and  $P$  are compatible, that  $P$  is Markov relative to  $G$ , or that  $G$  is an I-map (independency map) of  $P$ .*

Markov compatibility is a necessary and sufficient condition for us to model  $G$  as a data generating mechanism for a probability distribution  $P$ . The *matching* between  $G$  and  $P$  can be done by comparing *d-separation* in  $G$  and conditional independencies in  $P$ .

**Definition 2** (d-separation [4]). *A sequence of consecutive edges in DAG  $G$  regardless of directionality is called a path. Let  $X, Y, Z$  be disjoint sets of vertices in  $G$ . Then, a path is said to be d-separated by  $Z$  if and only if*

1. *a path contains a chain  $i \rightarrow k \rightarrow j$  or a fork  $i \leftarrow k \rightarrow j$  such that the middle vertex  $k$  is in  $Z$ , or*
2. *a path contains an inverted fork (or collider)  $i \rightarrow k \leftarrow j$  such that the middle vertex  $k$  is not in  $Z$  and such that no descendant of  $k$  is in  $Z$ .*

*$Z$  is said to d-separate  $X$  from  $Y$  if and only if  $Z$  d-separates every path from a vertex in  $X$  to a vertex in  $Y$ .*

The notion of d-separation represents conditional independencies entailed by causal relationships in the graph. Let  $\mathcal{I}(G)$  be the set of conditional independencies that can be read off by d-separation in  $G$  and  $\mathcal{I}(P)$  be the set of conditional independencies in  $P$ . Then, the following theorem connects these sets succinctly.

**Theorem 1** (I-map [4]). *For any three disjoint subsets of vertices  $(X, Y, Z)$  in a DAG  $G$  and for all probability functions  $P$ ,  $\mathcal{I}(G) \subseteq \mathcal{I}(P)$  whenever  $G$  and  $P$  are compatible, i.e.,  $G$  is an I-map of  $P$ .*

Note that multiple DAGs may encode the same conditional independence relationships. We say such DAGs are *Markov equivalent*. Since these DAGs are statistically indistinguishable, identifying the true underlying DAG from observational data alone is not possible in general. Let the skeleton of a DAG  $G$  be the undirected graph obtained from replacing all directed edges of  $G$  with undirected edges. Also, a collider  $i \rightarrow k \leftarrow j$  is called a v-structure if  $i$  and  $j$  are not adjacent. Then, the following theorem characterizes Markov equivalent DAGs.

**Theorem 2** (Markov Equivalence [5]). *Two DAGs  $G_1$  and  $G_2$  are Markov equivalent if and only if they have the same skeleton and the same set of  $v$ -structures.*

However, a class of Markov equivalent DAGs can be identified from observational data alone under the *stability* (also called *faithfulness*) assumption.

**Definition 3** (Stability [2, 6]). *If all and only the conditional independencies in  $P$  are entailed by  $G$ , i.e.,  $\mathcal{I}(P) = \mathcal{I}(G)$ , then we say that  $P$  is a stable distribution of  $G$ , or that  $P$  is faithful to  $G$ .*

Under the stability assumption, if there are no hidden variables, Theorem 2 ensures that every probability distribution has a unique *minimal* DAG up to a Markov equivalence class.

**Definition 4** (Minimality [4]).  *$G$  is minimal if and only if any proper subgraph  $G'$  of  $G$  is not an  $I$ -map of  $P$ . We call such a  $G$  a minimal  $I$ -map of  $P$ .*

### 2.1.2 Challenges

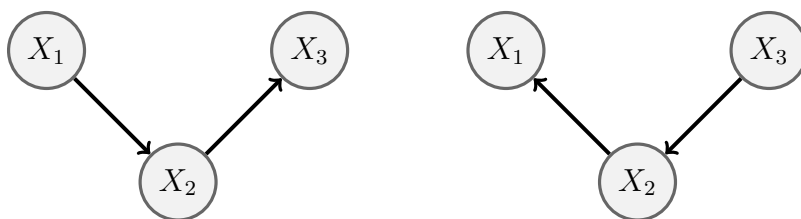


Figure 2.1: Two statistically indistinguishable DAGs

Estimating the structure of a Bayesian network is challenging mainly due to the following three reasons. First, one can identify a Bayesian network only up to a Markov equivalence class with observational data alone [5]. Two DAGs shown in Figure 2.1 are statistically indistinguishable because they share the same set of conditional independence relationships in which there is only one,  $\{X_1 \perp X_3 \mid X_2\}$ . Secondly, the number

of possible Bayesian networks grows super-exponentially as the number of vertices increases [7], which makes exhaustive search infeasible except for very small graphs. For example, a total number of a  $p$ -variate Bayesian network can be recursively computed by  $|B_p| = \sum_{i=1}^p (-1)^{i+1} \binom{p}{i} 2^{i(p-i)} |B_{p-i}|$ , and even for a small Bayesian network of size  $p = 14$ , we have  $|B_{14}| = 1,439,428,141,044,398,334,941,790,719,839,535,103$  DAGs to enumerate. Lastly, the acyclicity of Bayesian networks adds a non-convex constraint to the problem. The aforementioned characteristics of Bayesian network make analyzing high-dimensional data beyond hundreds of variables challenging even for many state-of-the-art methods in the literature.

### 2.1.3 Previous Work

There are mainly three approaches in Bayesian network structure learning.

1. *Constraint-based approach.*

Constraint-based approaches perform a series of conditional independence tests (CI tests) to identify the skeleton of a DAG and v-structures. Then, they proceed to recover as many edge directions as possible by iteratively applying a set of rules that prevent a graph from having cycles or any additional edges in the skeleton other than those already found by CI tests. One of the well-known constraint-based methods is PC algorithm [8].

2. *Score-based approach.*

Score-based approaches optimize a score function to identify a DAG structure. However, since Bayesian network structure learning in the DAG space is known to be NP-hard [9], researchers have proposed various heuristic methods: greedy search methods [10, 11], a method that transforms the combinatorial search space to a continuous space [12], methods that search over the permutation space [13, 14, 15].

### 3. *Hybrid approach.*

Hybrid approaches combine the two approaches above. Commonly, they first apply a constraint-based method to construct the skeleton of a Bayesian network, then apply a score-based method to orient edges in the skeleton. Max-Min Hill-Climbing (MMHC) of [16] is a well-known example of hybrid approach.

## 2.2 Preliminaries

Consider  $p$ -variate Gaussian random vector  $X = (X_1, \dots, X_p)^T \sim N(0, \Sigma)$  with covariance structure  $\Sigma$ . Assuming linear dependencies between variables of  $X$ , structural equation modeling writes

$$X_j = \sum_{i \in pa_G(j)} \beta_{ij} X_i + \epsilon_j, \quad j = 1, \dots, p \quad (2.2)$$

where  $\beta_{ij}$  are regression coefficients and  $\epsilon_j \sim N(0, \omega_j^2)$ ,  $\epsilon_j \perp \epsilon_k \forall k \neq j$ . In matrix form, we can write

$$X = B^T X + \epsilon \quad (2.3)$$

where  $\epsilon = (\epsilon_1, \dots, \epsilon_p)^T$  and  $B$  is the coefficient matrix, i.e.,  $(B)_{ij} = \beta_{ij}$  whose nonzero patterns represent the Bayesian network structure. Let  $\Theta = \Sigma^{-1}$  denote the precision matrix and  $\Omega = \text{diag}(\omega_1^2, \dots, \omega_p^2)$  be the diagonal matrix with error variances. Then it is straightforward to show that  $\Theta$  can be expressed as a function of  $B$  and  $\Omega$

$$\Theta(B, \Omega) = (I - B)\Omega^{-1}(I - B)^T. \quad (2.4)$$

Since zero correlation implies conditional independence under multivariate Gaussianity, the zeros of the precision matrix  $\Theta$  coincides with the missing edges in the correspond-

ing undirected graph. Since zero/nonzero patterns in  $\Theta$  and  $B$  coincide with edges in undirected graph and DAG, respectively, we will use these terms interchangeably for simplicity.

Notice that (2.4) becomes the Cholesky decomposition when arranging variables according to a reversed topological ordering denoted by  $\pi$ . In other words,  $B_\pi = P_\pi B P_\pi^T$  is a strictly lower triangular matrix where  $P_\pi$  represents a permutation matrix corresponding to the ordering  $\pi$ . We refer to [14, Fig. 1] for a visual example. Hence, after ordering variables according to  $\pi$ ,  $\Theta_\pi = P_\pi \Theta P_\pi^T$  can be translated into its canonical candidate DAG by computing its modified Cholesky factors:  $\Theta_\pi = T_\pi D_\pi T_\pi^T$ , where  $T_\pi = I - B_\pi$  is computed from ordinary Cholesky factors  $\Theta_\pi = L_\pi L_\pi^T$ . The lower triangular matrix  $T_\pi$  with unit diagonal entries is a scaled version of  $L_\pi$ , and the diagonal matrix  $D_\pi = \Omega_\pi^{-1}$  has  $(D_\pi)_{ii} = (L_\pi)_{ii}^2$  for  $i \in V$  as the diagonal entries. Also, note that the lower triangular shape of  $B_\pi$  ensures the acyclicity condition since any variable  $j$  is linearly dependent only on preceding variables  $i$  such that  $i \prec j$  where  $i, j \in V$  as in (2.2), and coefficient  $\beta_{ij} \neq 0$  implies  $(i, j) \in E$  [17]. Here,  $\prec$  denotes a linear ordering between vertices such that if  $i$  is a parent of  $j$  in the DAG model or  $i \rightarrow j$ , then  $i \prec j$ .

The Cholesky factor  $L$  is dependent on the ordering of rows and columns of  $\Theta$ , so it is often desired to estimate the DAG representing the distribution with minimal number of edges called *minimal-edge I-MAP* [18]. In fact, in the noiseless Gaussian setting, recovering the minimal-edge I-MAP amounts to finding a variable ordering  $\pi$  so that the Cholesky factor in  $\Theta_\pi = L_\pi L_\pi^T$  is the sparsest [13]. Figure 2.2 illustrates the impact variable ordering of  $\Theta$  has on the zero/nonzero patterns in the resulting Cholesky factor. Recently published methods such as ARCS [14] and RFD [15] perform DAG recovery by estimating the permutation ordering  $\pi$  along with the corresponding Cholesky factor  $L$ .

However, even when the variable ordering is known, the quality of the precision matrix  $\Theta$  greatly affects the performance of DAG estimation. This is due to the sequential

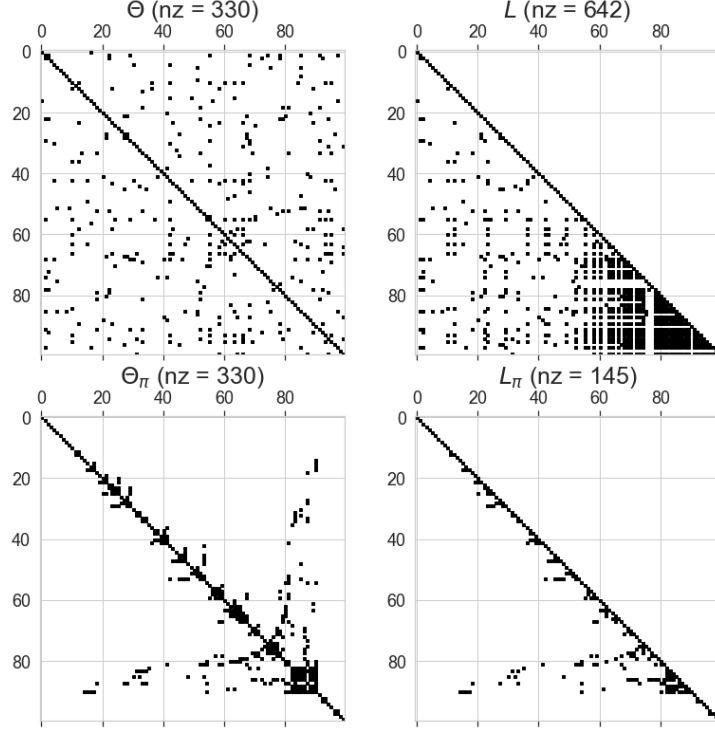


Figure 2.2: The top row represents an unordered sparse undirected graph and its Cholesky factor while the bottom row represents those ordered by sparse Cholesky decomposition. nz indicates the number of nonzero entries.

procedure of Cholesky decomposition that is prone to accumulate errors. We will illustrate empirical evidence of this and introduce distributionally robust optimization to tightly control the number of false edges in  $\Theta$ .

As discussed, the performance of DAG estimation from data hinges on the quality of the precision matrix  $\Theta$  and the permutation ordering  $\pi$ . Based on these observations, we propose a novel algorithm for DAG structure learning called FROSTY whose pseudo-code is given in Algorithm 1. Two distinct features/advantages of FROSTY are its tuning simplicity and scalability to high-dimensional data. In Section 2.6, FROSTY is empirically evaluated in comparison to other recent DAG structure learning methods.

---

**Algorithm 1:** FROSTY algorithm

---

**Input** : Dataset  $X$ , confidence level  $\alpha$ **Output:** Coefficient matrix of Bayesian network  $B$ *Step 1: Estimate undirected graph  $\Theta$* 

- 1 Set  $\lambda_\alpha = \text{RobSel}(X, \alpha)$  using Algorithm 2.
- 2 Estimate  $\Theta = \Theta(\lambda_\alpha)$  with graphical lasso.

*Step 2: Recover Bayesian network  $B$* 

- 3 Compute Cholesky factor and permutation matrix,  $L_\pi, P_\pi = \text{AMD}(\Theta)$ .
  - 4  $B = P_\pi^T(\text{diag}(L_\pi) - L_\pi)\text{diag}(L_\pi)^{-1}P_\pi$ .
  - 5 **return**  $B$
- 

## 2.3 Distributionally Robust Optimization for Cholesky Factor

First, we introduce a distributionally robust estimate of Cholesky factor. Consider the Gaussian negative log-likelihood function

$$\ell(S; L) = \text{trace}(SLL^T) - 2\log |L|$$

where  $L$  is the Cholesky factor of the precision matrix as in  $\Theta = LL^T$  and  $S$  is the sample covariance matrix. Then, a distributionally robust estimate of  $L$  is obtained by solving

$$\min_{L \in \mathbb{L}_p} \sup_{\mathcal{P} \in \mathcal{S}} E_{\mathcal{P}}[\ell(S; L)]$$

where  $\mathbb{L}_p$  is the set of  $p \times p$  lower triangular matrices with positive diagonal and  $\mathcal{S}$  is called an *ambiguity set*. Let  $\mathcal{S} = \{\mathcal{P} : \mathcal{D}_c(\mathcal{P}, \mathcal{P}_n) \leq \lambda\}$ , the collection of measures such that some notion of distance between plausible distribution  $\mathcal{P}$  for the true probability measure  $\mathcal{P}_0$  and empirical distribution  $\mathcal{P}_n$  is bounded by an *ambiguity size*  $\lambda$ . Let  $\mathcal{D}_c(\cdot, \cdot)$

be the optimal transport cost function between two distributions defined by

$$\mathcal{D}_c(P_1, P_2) = \inf \{E_\pi[c(U, V)] : \pi \in \mathcal{P}(\mathbb{S}_p \times \mathbb{S}_p), \pi_U = P_1, \pi_V = P_2\}$$

where  $\mathbb{S}_p$  is the set of  $p \times p$  symmetric matrices and  $\mathcal{P}(\mathbb{S}_p \times \mathbb{S}_p)$  is the set of joint probability distributions  $\pi$  on  $(U, V)$  supported on  $\mathbb{S}_p \times \mathbb{S}_p$ .  $\pi_U$  and  $\pi_V$  are the marginal distributions of  $U$  and  $V$ , respectively. Let  $\text{vec}(A) \in \mathbb{R}^{p^2}$  be a vectorized form of a matrix  $A \in \mathbb{R}^{p \times p}$  in row-major order, then the following is the cost function of our choice

$$c(U, V) = \|\text{vec}(U) - \text{vec}(V)\|_\infty.$$

**Theorem 3.** *The  $\ell_1$ -regularized optimization problem of  $L$  can be recasted as the Distributionally Robust Optimization (DRO) formulation,*

$$\underbrace{\min_{L \in \mathbb{L}_p} \sup_{\mathcal{P}: D_c(\mathcal{P}, \mathcal{P}_n) \leq \lambda} E_{\mathcal{P}}[\ell(S; L)]}_{\text{DRO formulation}} = \underbrace{\min_{L \in \mathbb{L}_p} \{\ell(S; L) + \lambda \|\text{vec}(LL^T)\|_1\}}_{\ell_1\text{-regularized formulation}}. \quad (2.5)$$

Theorem 3 shows the equivalence of parameter  $\lambda$  between the amount of regularization (right-hand side) of (2.5) and the ambiguity size of the DRO formulation (left-hand side). Utilizing this equivalence, we suggest a selection criterion for  $\lambda$  based on  $(1-\alpha)$ -confidence level for  $L$  by modifying the derivation found in [19]. First, let us construct the confidence region for  $L$ . For any  $L \in \mathbb{L}_p$ ,  $\mathcal{O}(L)$  denotes the set of all probability measures that make the gradient vanish, i.e.,

$$\mathcal{O}(L) = \{\mathcal{P} : E_{\mathcal{P}} \left[ \frac{\partial}{\partial L'} \ell(S; L') \Big|_{L'=L} \right] = 0_{p,p}\}.$$

where  $0_{p,p}$  is a  $p \times p$  zero matrix. Then we consider the set of plausible Cholesky factors

for  $L$  by

$$\mathcal{C}_n(\lambda) = \{L \in \mathbb{L}_p : \exists \mathcal{P} \in \mathcal{O}(L) \cap \{\mathcal{P} : D_c(\mathcal{P}, \mathcal{P}_n) \leq \lambda\}\}.$$

In fact, since  $\mathcal{P}_n$  weakly converges to the true probability measure  $\mathcal{P}_0$  for any  $\lambda$ ,  $\mathcal{C}_n(\lambda)$  is a confidence region for  $L$ . Define the *Robust Wasserstein Profile (RWP) function*  $R_n$  as follows:

$$R_n(L) = \inf \{D_c(\mathcal{P}, \mathcal{P}_n) : \mathcal{P} \in \mathcal{O}(L)\},$$

which represents the minimum distance between the empirical distribution and any plausible distribution that satisfies the first-order optimality condition for  $L$ .

The following theorem demonstrates that the optimal amount of regularization  $\lambda_\alpha$  for  $\ell_1$ -regularized optimization problem of  $L$  can be completely determined by a user-specified error tolerance  $\alpha$  as in  $(1 - \alpha)$ -confidence level for  $L$ .

**Theorem 4.** *The RWP function for  $L$  is*

$$R_n(L) = \|\text{vec}(S - (LL^T)^{-1})\|_\infty,$$

and a robust selection for the ambiguity size  $\lambda$  of the DRO formulation for  $L$  in (2.5) is

$$\lambda_\alpha = \inf \{\lambda > 0 : \mathbb{P}_0(\|\text{vec}(S - (LL^T)^{-1})\|_\infty \leq \lambda) \geq 1 - \alpha\} \quad (2.6)$$

where  $\mathbb{P}_0$  is the measure of the Cholesky factor of the random matrix  $(XX^T)^{-1}$  induced by the true probability law for  $X$ .

The implication of Theorems 3 and 4 is that estimating  $L$  can be achieved in two stages as in Algorithm 1: *Step 1* is the sparse estimation of  $\Theta$ , and *Step 2* is the Cholesky

**Algorithm 2:** RobSel algorithm [19]

---

**Input** : Dataset  $X$ , confidence level  $\alpha$   
**Output**: Graphical lasso regularization parameter  $\lambda_\alpha$

- 1 **for**  $k \leftarrow 1, \dots, b$  **do**
- 2     Obtain bootstrap sample  $X_{1k}^*, \dots, X_{nk}^*$  by sampling uniformly with replacement.
- 3     Compute  $R_{n,k}^* = \|A_{n,k}^* - A_n\|_q$ , with empirical covariance  $A_{n,k}^*$  of bootstrap sample.
- 4 **end**
- 5 Set  $\lambda_\alpha$  as bootstrap order statistic  $R_{n,((b+1)(1-\alpha))}^*$ .
- 6 **return**  $\lambda_\alpha$

---

factorization of  $\Theta = LL^T$ . Since the DAG structure in  $L$  corresponding to an undirected graph structure in  $\Theta$  depends heavily on variable ordering as discussed in Section 2.2, a natural question to ask is whether  $\lambda_\alpha$  is invariant under variable ordering permutations.

In the following lemma, we show that  $\lambda_\alpha$  does not depend on variable ordering.

**Lemma 1.** *Let  $\pi$  be a variable ordering consistent with the minimal-edge I-MAP  $B_\pi$  and  $L_\pi$  be the corresponding Cholesky factor, i.e.,  $B_\pi = (\text{diag}(L_\pi) - L_\pi)\text{diag}(L_\pi)^{-1}$ . For the precision matrix  $\Theta$ , suppose  $P_\pi$  is a permutation matrix such that  $P_\pi \Theta P_\pi^T = \Theta_\pi = L_\pi L_\pi^T$ . Then, the RWP function for  $L_\pi$  is equivalent to the RWP function for  $\Theta$ , i.e.,*

$$R_n(L_\pi) = \|\text{vec}(S_\pi - (L_\pi L_\pi^T)^{-1})\|_\infty = \|\text{vec}(S - \Theta^{-1})\|_\infty = R_n(\Theta). \quad (2.7)$$

A direct consequence of Lemma 1 is that  $\Theta$  only needs to be estimated once independent of the variable ordering  $\pi$  since  $R_n(L_\pi) = R_n(\Theta)$ . Additionally, Lemma 1 justifies estimation  $\lambda_\alpha$  by Robust Selection (RobSel) algorithm of [19] given in Algorithm 2.

RobSel algorithm only requires bootstrapped sample covariance matrices to determine  $\lambda_\alpha$ , without requiring computationally expensive cross-validation. Furthermore, in the recent work of [20], it has been shown that  $\alpha$  in RobSel is the upper bound of the

asymptotic family-wise error rate of estimating at least one erroneous nonzero in  $\Theta$ . Since the DRO formulation is to minimize the worst-case scenario, RobSel tends to be conservative even for large values of  $\alpha$ , and, in fact, a large value of  $\alpha$  is recommended, e.g.,  $\alpha = 0.99$  was routinely used with good results. We remark that using RobSel-tuned  $\Theta$  as inputs might also improve other methods. In our numerical experiments, RFD [15] estimates improve when RobSel-tuned  $\Theta$  is provided as input.

## 2.4 Recovering Variable Ordering

To ultimately recover the minimal-edge I-MAP  $B_\pi$  from the precision matrix  $\Theta$ , we also need to estimate a variable ordering  $\pi$  that leads to the sparsest Cholesky factor  $L_\pi$ . Since recovering such a permutation  $\pi$  is known to be NP-complete [21], a heuristic method called Approximate Minimum Degree ordering (AMD) [22] is used in FROSTY. AMD algorithm is a more efficient variant of the minimum degree ordering [23] method in which the upper bound of the vertex degree is approximated instead of an exact one.

For Gaussian distribution, the relationship between the Cholesky factor of the precision matrix and the ordered conditional independence is well known [24]. Furthermore, the minimum fill-in problem, i.e. sparse Cholesky decomposition, is equivalent to the minimum chordal problem [25], and, coincidentally, scale-free graphs are natural candidates for the problem [26]. In applications, many networks are believed to have scale-free structure, which motivates our selection of AMD as a method to recover the minimal-edge I-MAP from  $\Theta$ .

## 2.5 Computational Complexity

Let  $b$  be the number of bootstrap samples in RobSel. RobSel only requires to compute  $b$  sample covariance matrices, whose computation time is  $O(bnp^2)$ . Then, we estimate undirected graph with graphical lasso with  $O(p^3)$ , which is the computational cost of FROSTY. However, the computation time of graphical lasso is known to be significantly shorter for sparse problems [27].

Lastly, the computational complexity of AMD is

$$O\left(\sum_{k=1}^p |L_{k*}| \cdot |(P\Theta P^T)_{k*}| \right) \approx O(|L|) \ll O(p^2)$$

where  $L$  is the Cholesky factor and the subscript  $k*$  indicates nonzero elements in row  $k$ . AMD is a semi-deterministic algorithm that is efficiently implemented in *sksparse.cholmod*. We observe that the computation time of AMD is almost ignorable compared to graphical lasso. Hence, the computational complexity of FROSTY is  $O(p^3)$ .

## 2.6 Numerical Experiments

### 2.6.1 Accuracy Metrics

As discussed in Introduction, since Markov equivalent DAGs cannot be statistically distinguished based on observational data, we compute all accuracy metrics based on Completed Partially Directed Acyclic Graph (CPDAG). For a DAG  $G$ , CPDAG is obtained by keeping only consistent directed edges among all DAGs in the equivalence class of  $G$  and converting others to undirected edges.

We focus on the following metrics: structural hamming distance (SHD), false discovery rate (FDR), and matthews correlation coefficient (MCC). SHD is the sum of numbers

of false positive (FP), reversed (R), and missing (M) edges. FP and M are counted from the skeleton of CPDAG, and R counts directionally inconsistent edges. However, FDR and MCC are computed on the adjacency matrix of CPDAG. The reasoning behind this decision is that we noticed that SHD is often misleading when the true underlying graph is very sparse, e.g., an empty graph can have a much smaller SHD than any estimated graphs with moderate FP. On the other hand, MCC is robust against class imbalance and invariant for class swapping [28]. Also, another shortcoming of SHD is that it treats FP, R, and M all equally, which is debatable.

Lastly, all accuracy metrics in numerical experiments are averaged over 20 randomly generated datasets unless otherwise noted.

## 2.6.2 Parameter Tuning

Since RFD requires  $\Theta$  as an input, we provide two different estimate schemes: RS and CV indicate estimated  $\Theta$  by RobSel and estimated  $\Theta$  by Cross-Validation, respectively. For RFD (CV), we used a 5-fold CV with a 20 evenly spaced grid of  $\lambda$  on a log-scale where the maximum value of the grid was chosen by the maximum of the absolute values of the sample covariance matrix excluding diagonal entries. The minimum value was chosen to be 0.05 times of the maximum value. It is important to note that, for FROSTY, we simply chose  $\alpha = 0.99$  throughout the entire simulation without any parameter tuning.

### 2.6.3 Empirical Motivation for RobSel

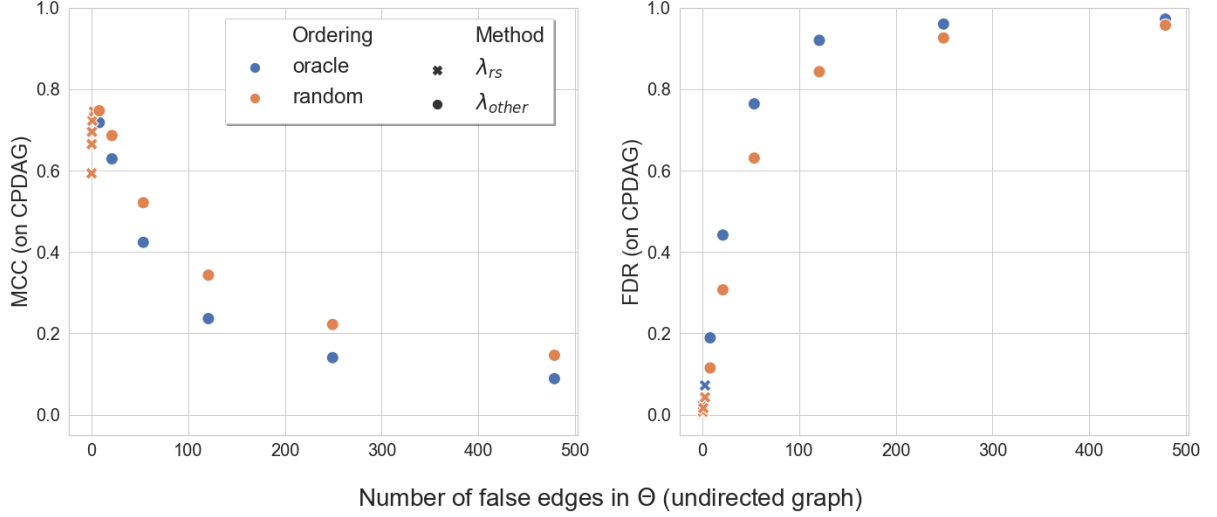


Figure 2.3: The effect of false edge control of RobSel on DAG estimation. For RobSel, a grid of  $\lambda_{rs} = [\lambda_{0.1}, \lambda_{0.3}, \lambda_{0.5}, \lambda_{0.7}, \lambda_{0.9}]$  is used.

Figure 2.3 illustrates the benefits of tightly controlling the number of false edges in  $\Theta$  by RobSel, which in turn leads to greater accuracy in DAG estimation. Based on Gaussian data generated from a scale-free Bayesian network of size  $p = 100$  and sample size  $n = 200$ ,  $\Theta$ 's are estimated with graphical lasso given two grids of  $\lambda$ .  $\lambda_{rs} = [\lambda_{0.1}, \lambda_{0.3}, \lambda_{0.5}, \lambda_{0.7}, \lambda_{0.9}]$  are chosen by RobSel where the subscripts represent the values of  $\alpha$  used, and  $\lambda_{other} = [\lambda_1, \dots, \lambda_{10}]$  are 10 evenly spaced values between  $(0, \lambda_{0.9})$ . However, only the 6 largest values in  $\lambda_{other}$  are shown in Figure 2.3 for visual clarity. For DAG recovery, ordinary Cholesky decomposition is applied when the variable ordering is known while AMD is applied when the ordering is unknown.

When the regularization parameter  $\lambda$  is chosen by RobSel, graphical lasso conservatively estimates  $\Theta$  resulting in low FDR with relatively high MCC on its CPDAG. However, notice that FDR surges and MCC drops rapidly when we select  $\lambda$  smaller than the smallest one chosen by RobSel, i.e.  $\lambda_{0.9}$ . This empirically motivates one to choose

RobSel over cross-validation, which always selects smaller  $\lambda$  than RobSel [19] and has a tendency toward overfitting [29].

### 2.6.4 Simulations

To test FROSTY, we generate two synthetic random graphs, scale-free graphs and Erdos-Renyi graphs, with varying graph sizes  $p \in \{50, 100, 150, 200\}$ . Scale-free graphs are generated by the preferential attachment model [30] with  $m = 1$  where  $m$  is the number of random edges to add for each vertex. The edge direction is chosen from an existing vertex to a newly added vertex. For Erdos-Renyi graphs, we first generate undirected Erdos-Renyi graphs whose expected number of edges are  $p$  and  $2p$ , and take their lower triangular matrix. Then, for each graph, its corresponding precision matrix is obtained by the equation (2.4) whose edge weights are chosen from  $\text{Uniform}(0.25, 1)$  with equal probability of its sign being positive or negative. Symmetry and positive definiteness of the precision matrix are ensured by averaging with its transpose and scaling it to be diagonally dominant. Multivariate Gaussian data are generated from the inverse of the precision matrix with two different sample sizes proportional to the graph size:  $n \in \{0.5p, 2p\}$ . Note that even if a Bayesian network is sparse and follows scale-free degree distribution, its undirected graph can be dense due to a large number of v-structures. We investigate the performance of FROSTY when a scale-free Bayesian network has a significant number of v-structures. Such a scale-free graph is obtained by generating the preferential attachment model with  $m = 1$  and randomly reversing edge directions. For our simulation, we randomly reverse 25% of the edges.

**Scale-free graph** Figure 2.4 shows that FROSTY overall obtained higher scores than ARCS and RFD (CV) when the underlying graph structure has scale-free degree distribution while FROSTY and RFD (RS) were nearly identical. This indicates that RFD

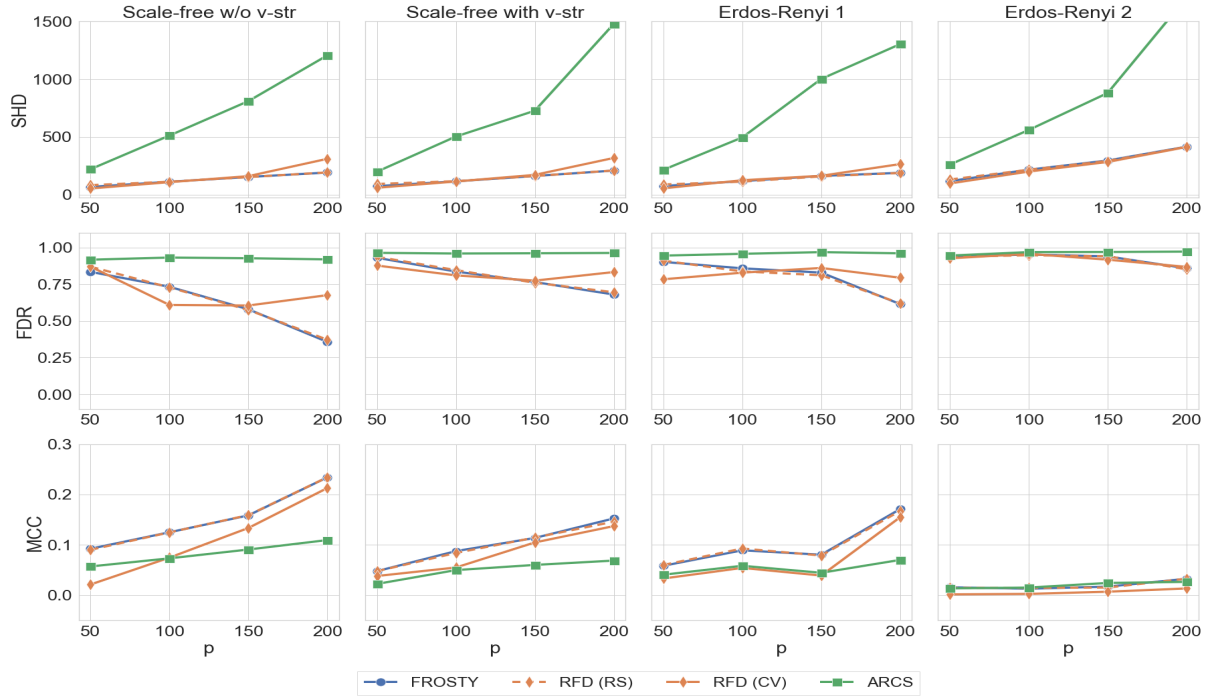
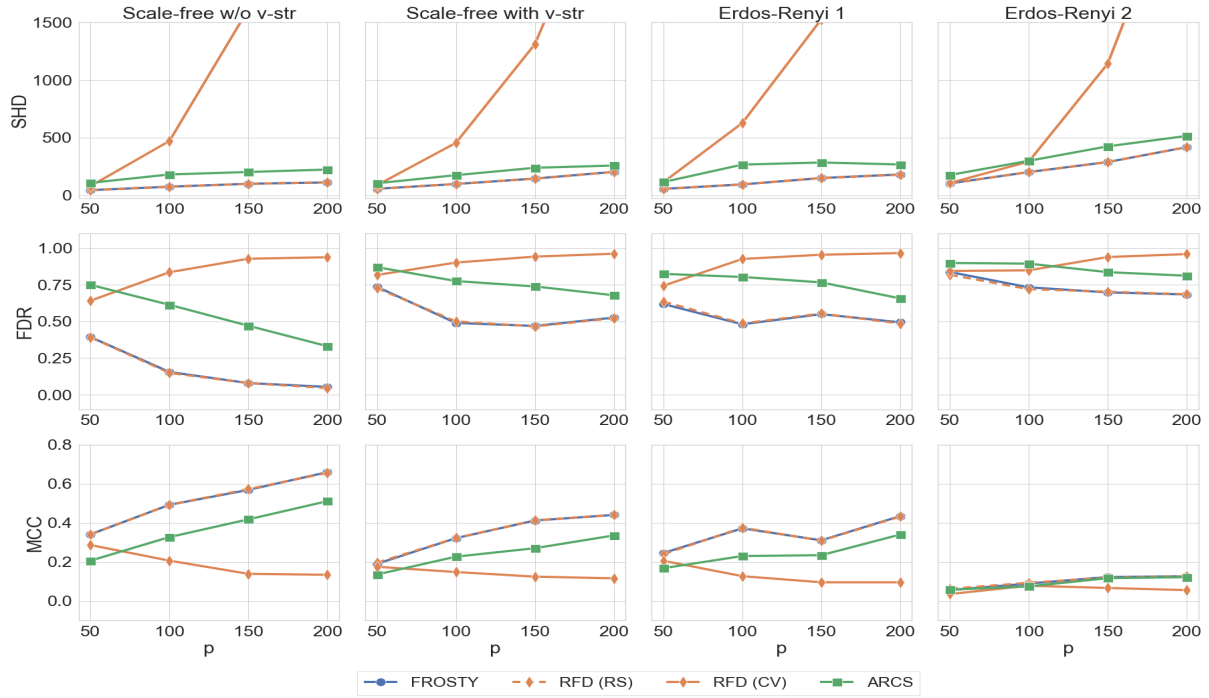
(a) Gaussian data with sample size  $n = 0.5p$ (b) Gaussian data with sample size  $n = 2p$ 

Figure 2.4: Performance comparison for various graph structures

can be significantly improved by carefully selecting undirected graph with RobSel. There was virtually no difference in performance between FROSTY and RFD (RS) even though the RFD algorithm utilizes additional DAG-specific structure on top of vertex degrees for its permutation recovery. This addition increases the computational complexity of RFD resulting in  $O(p^4)$  in the best case scenario (when the depth parameter is chosen to be 1) while AMD, which only utilizes vertex degree information, boasts almost linear complexity. The fact that FROSTY is computationally more efficient than RFD (RS) whilst being able to estimate as accurate CPDAGs demonstrates FROSTY could be a better choice of algorithm under the scale-free assumption on large-scale graphs. Table 2.1 and Figure 2.5 show the superb scalability of FROSTY.

| Method   | Runtime (sec.) |           |           |            |            |
|----------|----------------|-----------|-----------|------------|------------|
|          | $p = 100$      | $p = 200$ | $p = 500$ | $p = 1000$ | $p = 2000$ |
| FROSTY   | 0.12           | 0.28      | 1.39      | 5.52       | 33.69      |
| RFD (RS) | 0.86           | 6.52      | 168.62    | 2180.04    | —          |
| RFD (CV) | 33.59          | 177.88    | 1817.85   | 14250.32   | —          |
| ARCS     | 57.52          | 101.47    | 562.56    | 5721.50    | 14581.00   |

Note: For  $p = 2000$ , RFD did not run in the computing resource available to us (due to out of memory)

Table 2.1: Runtime analysis

**Erdos-Renyi graph** While the deviation from scale-free degree distribution and denser Erdos-Renyi graphs resulted in poorer performance of FROSTY shown in Figure 2.4, FROSTY still showed more favorable performance especially when the underlying graph is sparse. Also, notice that ARCS suffers significantly when the sample size is small and that RFD (CV) shows no improvement in FDR when the sample size increases. However, FROSTY remains competitive in both cases.

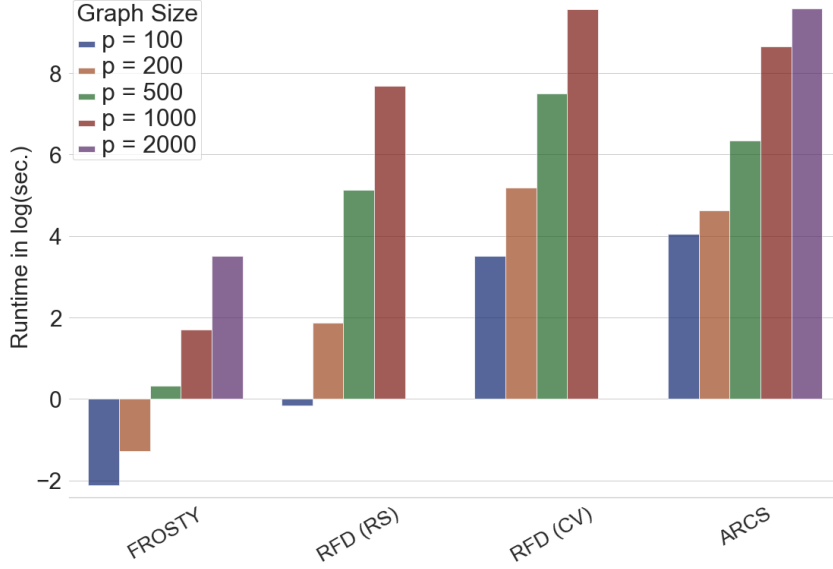


Figure 2.5: Log-scale runtime comparison between methods with varying graph sizes. The results are averaged over 20 runs except for the cases of  $p = 1000$  and  $2000$ , each of which is run only once.

## 2.7 Discussion

The FROSTY algorithm presents an efficient approach for Bayesian network structure learning for high-dimensional data. We summarize the strengths and limitations of FROSTY, and propose potential avenues for future research.

First and foremost, the experimental results demonstrate the competitive performance of FROSTY compared to other state-of-the-art methods. The superior scalability of FROSTY, particularly in the context of high-dimensional data, is a notable advantage. Another notable aspect is the minimal parameter tuning required by FROSTY. Users only need to specify an error tolerance parameter  $\alpha$ , corresponding to a desired  $(1 - \alpha)$ -confidence level for the Cholesky factor of the precision matrix.

However, it is important to acknowledge certain limitations of FROSTY. One limitation is its reliance on a generic variable ordering approach, neglecting the utilization of a DAG-specific structure. While FROSTY aims to maximize the sparsity of the resulting

DAG, this approach may be inappropriate in cases where the true underlying Bayesian network exhibits dense connections. Further research is needed to explore methods that can incorporate specific structural information to enhance variable ordering in FROSTY.

# Chapter 3

## Applications of FROSTY

In the preceding chapter, we showcased the remarkable scalability and competitive performance of FROSTY in comparison to other state-of-the-art methods in the literature, using two widely recognized random graph structures. Nevertheless, practitioners frequently encounter situations where the graph structures of their interest deviate from the characteristics of scale-free graphs or Erdos-Renyi graphs. Additionally, the application of estimation methods can be challenging due to the prevalent issue of missing data, which poses a common hurdle across various domains. Consequently, in this chapter, we present multiple applications of FROSTY, encompassing network estimation in real-world scenarios as well as estimation when confronted with missing data.

### 3.1 Real Network Analysis with FROSTY

In this section, we investigate various real networks. The flow cytometry data [31] is a frequently used real dataset in Bayesian network literature, which contains 11 proteins and phospholipids in human immune system cells ( $p = 11$ ) and 7466 measurements ( $n = 7466$ ). Its network structure is considered to be known from experiments. We

also consider Hailfinder and Pathfinder networks from *bnlearn* repository. Hailfinder is a weather forecast system based on meteorological data with expert judgment [32], and Pathfinder is an expert system that assists in diagnosing lymph-node pathology [33]. Network sizes are  $p = 56$  and 109, respectively. Gaussian data are generated in the same manner as the synthetic graph simulation except that here we try a variety of sample sizes:  $n \in \{0.5p, 2p, 10p\}$ .

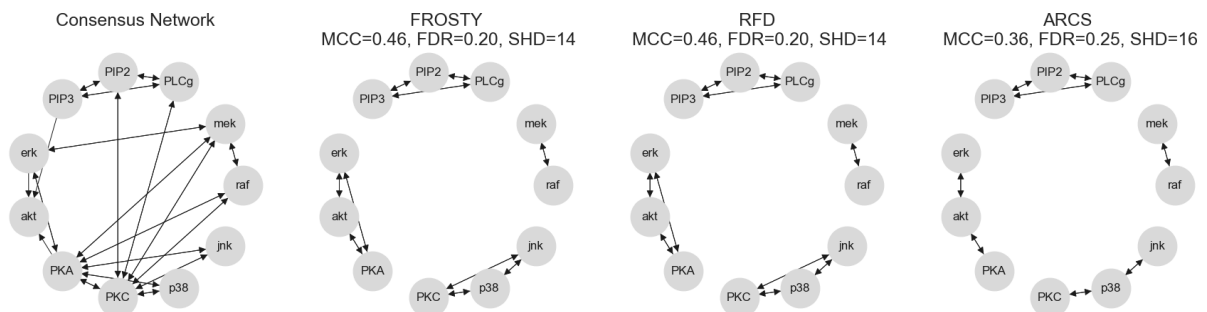


Figure 3.1: Consensus network (as in CPDAG) of flow cytometry data and estimated CPDAGs.

**Flow cytometry network** Since the original flow cytometry dataset hardly follows Gaussian distribution, log transformation was applied. All estimated CPDAGs and corresponding accuracy metrics are given in Figure 3.1. All three methods estimated similar CPDAGs, and, in fact, FROSTY and RFD estimated an identical CPDAG. The CPDAG estimated by ARCS differed by a couple of missing edges, both of which exist in the consensus network.

**Hailfinder & Pathfinder networks** Throughout various sample sizes, FROSTY attained consistently better MCC along with RFD (RS) as shown in Figure 3.2. On the other hand, RFD (CV) suffered from overfitting when the sample size gets large with noticeable increase in FDR, which is more clearly shown in Figure 3.3. Furthermore, Figure 3.3 also shows that FROSTY estimated true edges more consistently compared to

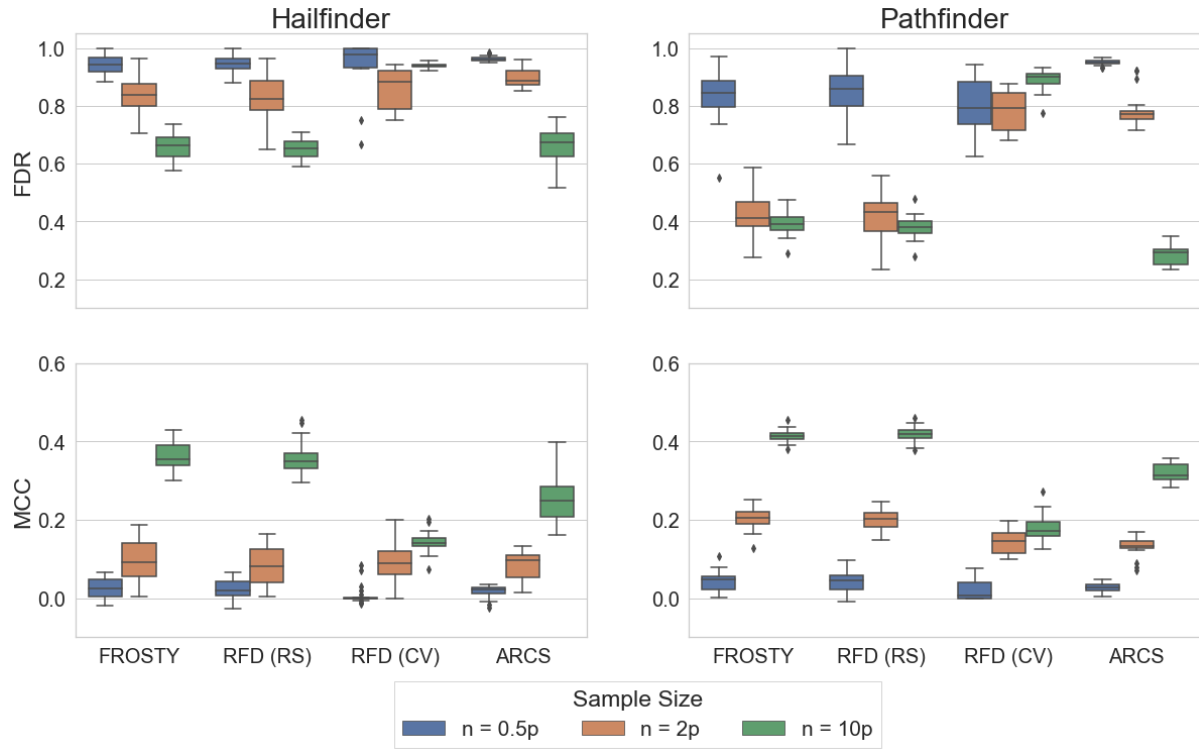


Figure 3.2: Performance comparison for Pathfinder and Hailfinder networks with varying sample sizes.

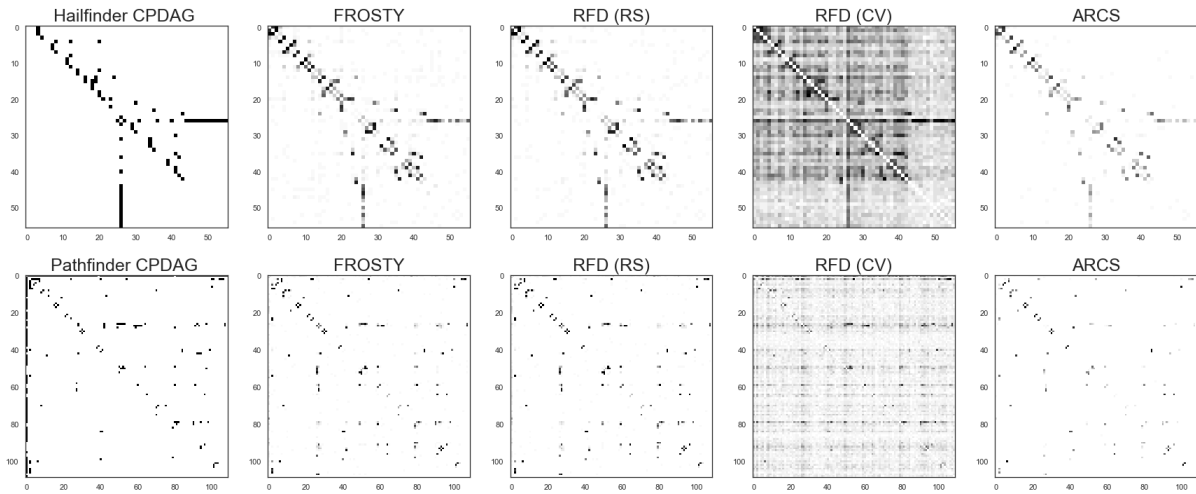


Figure 3.3: True and estimated CPDAGs for Hailfinder and Pathfinder networks. The edge inclusion percentage is represented by the grayscale. The results are based on sample size of  $n = 10p$ .

ARCS. We can observe the estimated edges of ARCS for the true edges are lighter than those of FROSTY.

## 3.2 Missing Data Analysis with FROSTY

### 3.2.1 Missing Mechanisms

The causes of missing data can be broadly classified into the three categories according to Little and Rubin [34]:

1. Missing Completely At Random (MCAR):

MCAR refers to missingness that occurs completely randomly, without any relation to observed or unobserved variables. In this case, the missingness does not introduce bias, allowing for straightforward analyses using only complete cases. However, relying on the MCAR assumption alone can be overly optimistic.

2. Missing At Random (MAR):

MAR denotes missingness that is not random but can be explained by observed variables, even though it may still depend on unobserved data. If the missingness is dependent on observed variables and can be accounted for, it is considered missing at random. Simply excluding observations with missing data, known as complete case analysis, can yield biased results when the MCAR assumption is violated.

3. Missing Not At Random (MNAR):

MNAR occurs when the missingness is associated with the unobserved or missing values themselves. This category presents the greatest challenge among the three. Since the missingness cannot be explained by observed variables, it is nonignorable,

and employing methods designed for MCAR or MAR assumptions may lead to biased outcomes.

### 3.2.2 General EM-based Framework, MissDAG

Recently, [35] proposed a general EM-based framework called *MissDAG* for performing DAG estimation in the presence of missing data where the underlying missing mechanism is independent of the observed information, encompassing MCAR and MAR assumptions.

Let us begin by defining the notations. The data matrix is denoted as  $X \in \mathbb{R}^{n \times p}$ , which may contain missing values. The indicator matrix  $Y \in \mathbb{R}^{n \times p}$  is defined such that  $Y_{ij} = 1$  if  $X_{ij}$  is observed, and  $Y_{ij} = 0$  if  $X_{ij}$  is missing. For each observation  $X_i$ ,  $X_{io} = [X_{ij} : Y_{ij} = 1, j = 1, \dots, p]$  represents the observed positions, and  $X_{im} = [X_{ij} : Y_{ij} = 0, j = 1, \dots, p]$  represents the missing positions. Similarly,  $X_O = [X_{ij} : Y_{ij} = 1, i = 1, \dots, n, j = 1, \dots, p]$  and  $X_M = [X_{ij} : Y_{ij} = 0, i = 1, \dots, n, j = 1, \dots, p]$  denote all observed positions and all missing positions, respectively. Using these notations, the full likelihood of the  $i^{th}$  sample can be written

$$\begin{aligned} \ell_{\text{full}}(X_{io}, Y_i; \theta, \psi) &= \int p(X_{io}, X_{im}; \theta) p(Y_i | X_{io}; \psi) dX_{im} \\ &= p(Y_i | X_{io}; \psi) \int p(X_{io}, X_{im}; \theta) dX_{im}. \end{aligned}$$

Here,  $\theta$  and  $\psi$  represent all model parameters and the parameters governing the missing mechanisms, respectively. Under the assumption of MCAR or MAR, the model parameters  $\theta$  can be inferred independently from the missingness mechanisms. In other words,

the ignorable likelihood  $\ell_{\text{ignorable}}(X_{io}; \theta)$ , defined as

$$\ell_{\text{ignorable}}(X_{io}; \theta) = \int p(X_{io}, X_{im}; \theta) dX_{im}$$

is sufficient to infer the model parameters  $\theta$ .

Under this ignorable missingness assumption, the MissDAG framework consists of two steps: the E-step and the M-step.

- E-step: Impute the missing values, given the current state of the estimated model parameters  $\theta^{(t)}$  and the observed data  $X_O$ .
- M-step: Update the model parameters to  $\theta^{(t+1)}$ .

It should be noted that the M-step is essentially the DAG estimation step, which can be substituted with any eligible DAG estimation method.

### 3.2.3 MissDAG for Gaussian Linear Structural Modeling

In the previous chapter, we introduced Gaussian linear structural equation modeling which writes

$$X = B^T X + \epsilon$$

where  $\epsilon = (\epsilon_1, \dots, \epsilon_p)^T$  with  $\epsilon_j \sim N(0, \omega_j^2)$ ,  $\epsilon_j \perp \epsilon_k \forall k \neq j$  and  $B$  is the coefficient matrix, i.e.,  $(B)_{ij} = \beta_{ij}$ . By defining  $\Omega = \text{diag}(\omega_1^2, \dots, \omega_p^2)$ , the complete log-likelihood can be expressed in terms of the model parameters  $\theta = (B, \Omega)$  as

$$\ell(S; B, \Omega) = \frac{np}{2} \log(2\pi) - \frac{n}{2} \log |(I - B)^{-T} \Omega (I - B)^{-1}| - \frac{1}{2} \text{trace}(S(I - B)\Omega^{-1}(I - B)^T)$$

where  $S = \frac{1}{n} X^T X$  represents the sample covariance matrix.

For the E-step, consider the partition  $X = [Y|Z]$ . Then, the Gaussian conditional distribution  $Y|Z = z$  follows  $N(\mu_y + \Sigma_{yz}\Sigma_{zz}^{-1}(z - \mu_z), \Sigma_{yy} - \Sigma_{yz}\Sigma_{zz}^{-1}\Sigma_{zy})$ . Therefore, assuming centered data, the expectation of missing positions for the  $k^{th}$  sample in the  $t^{th}$  iteration, denoted as  $\hat{X}_{km}^{(t)}$ , can be filled by  $\Sigma_{mo}^{(t)}(\Sigma_{oo}^{(t)})^{-1}X_{ko}$ . This corresponds to the expectation of the conditional distribution of  $X_M$  given  $X_O$ , using the model parameters  $\theta^{(t)}$ . Additionally, the sample covariance matrix  $S^{(t)} = E[S|X_O; \Sigma^{(t)}]$  can be computed by  $S_{ij}^{(t)} = \frac{1}{n} \sum_{k=1}^n \xi_k$  where

$$\xi_k = \begin{cases} \Sigma_{ij}^{(t)} + \hat{X}_{ki}\hat{X}_{kj} & \text{if } X_{ki} \text{ and } X_{kj} \text{ are missing,} \\ \hat{X}_{ki}\hat{X}_{kj} & \text{otherwise.} \end{cases}$$

For the M-step, we apply the FROSTY algorithm to update the model parameters  $(B, \Omega)$  using the imputed data obtained from the E-step. The updates are performed as follows:

$$\begin{aligned} B^{(t+1)} &= \text{FROSTY}(S^{(t)}) \\ \Omega^{(t+1)} &= (I - B^{(t+1)})^T S^{(t)} (I - B^{(t+1)}). \end{aligned}$$

### 3.2.4 Numerical Results

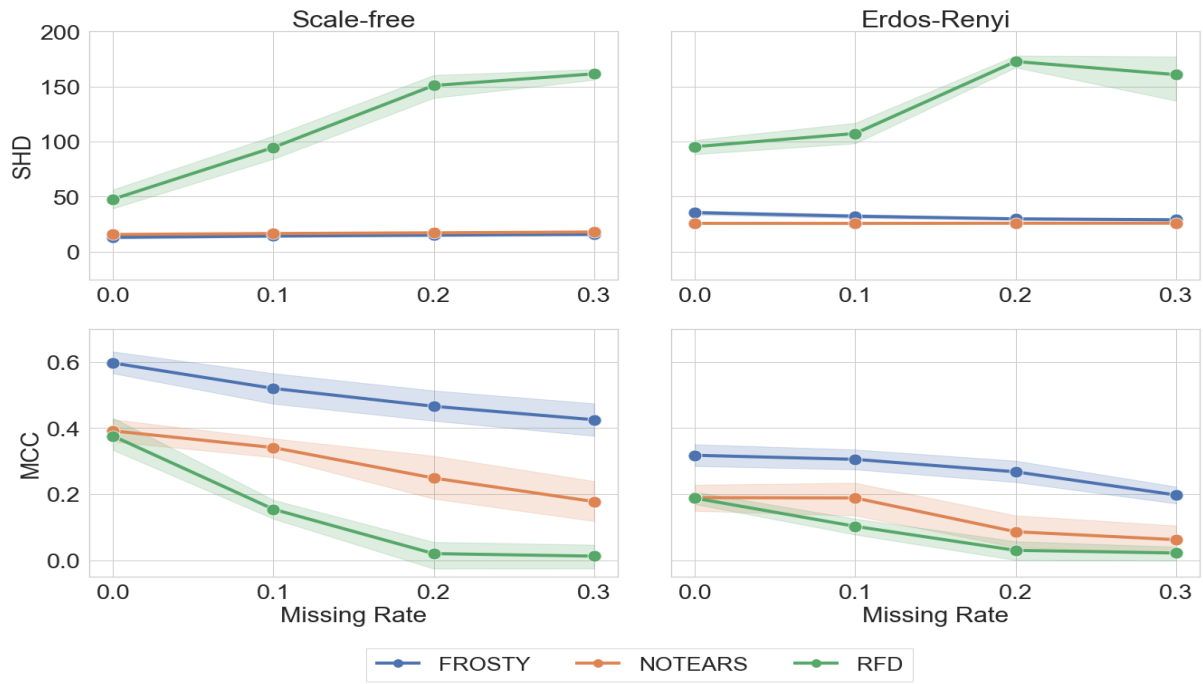
In this section, we present the numerical results of MissDAG with the FROSTY algorithm, and we compare its performance with that of MissDAG using NOTEARS [12] and RFD [15]. To conduct the simulations, we generate scale-free and Erdos-Renyi graph structures according to the procedures outlined in subsection 2.6.4 of the previous chapter. Considering the computational demands of the EM algorithm, we limit the simulations to a graph size of  $p = 20$ . Furthermore, we explore two distinct missing mechanisms, MCAR and MAR, while varying the missing rates from 0% to 30%. To

ensure appropriate sample sizes relative to the missing rates, we choose a sample size of  $n = 10p$ . Results are averaged over 20 randomly generated datasets.

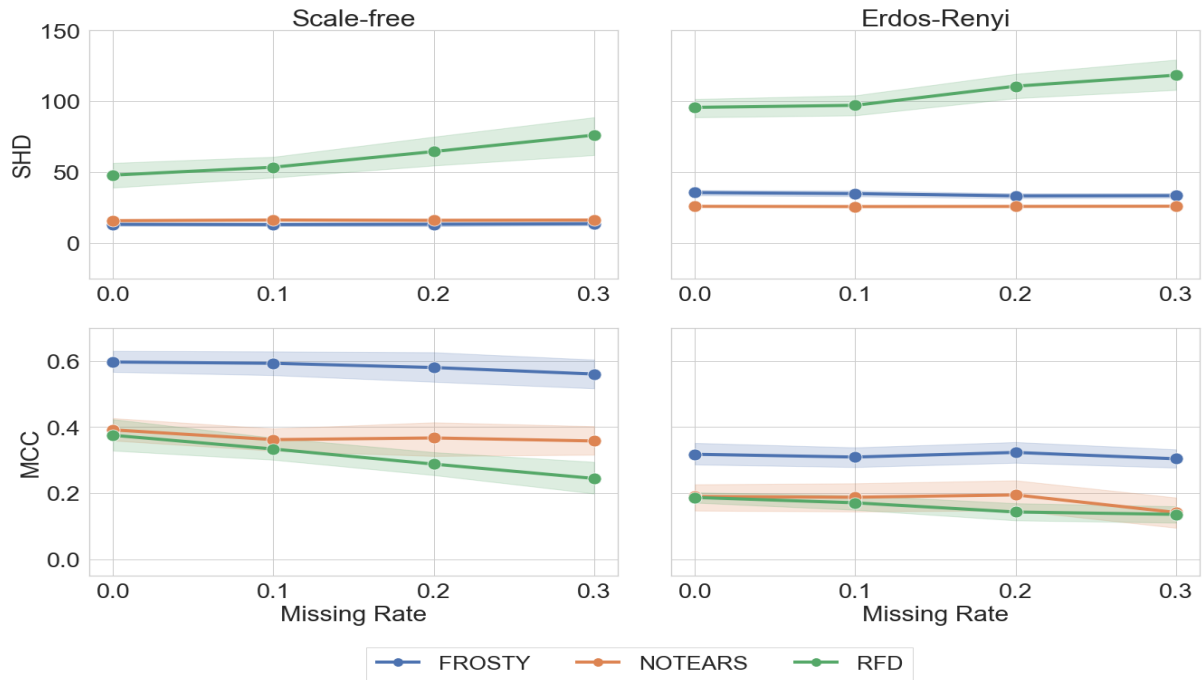
It is important to acknowledge the various challenges in conducting a fair comparison between the methods under consideration. Firstly, FROSTY and RFD estimate an undirected graph and subsequently convert it to a DAG based on variable ordering, while NOTEARS directly estimates a Bayesian network. As discussed in the previous chapter, FROSTY efficiently determines the appropriate level of  $\ell_1$ -regularization using the Rob-Sel method, which relies on the investigator’s confidence level. In the case of RFD, the  $\ell_1$ -regularization parameter is applied during the estimation of the undirected graph, and an appropriate range for the parameter is constructed based on the off-diagonal values of the sample covariance matrix [36]. However, selecting the appropriate range for the regularization parameter in NOTEARS is not straightforward. Additionally, due to the slow convergence of NOTEARS and the computational burden of cross-validation with a wide range of regularization parameters, we conducted multiple runs of NOTEARS on a sparse grid of parameter values and further refined the grid around the parameter value that exhibited the best performance. Another challenge arose from the fact that the estimated Bayesian networks by NOTEARS often contained cycles, making them unsuitable for conversion to CPDAGs. Although NOTEARS provides a parameter called *w\_threshold* that sets small nonzero values to exactly zero in the final step, selecting an appropriate threshold proved to be nontrivial. To ensure a fair comparison, we opted for a threshold value that rendered all estimated Bayesian networks acyclic across the 20 datasets.

The results are presented in Figure 3.4. For the scale-free graph, MissDAG with FROSTY consistently outperformed the other methods in terms of both SHD and MCC. MissDAG with FROSTY also demonstrated favorable performance in terms of MCC for the Erdos-Renyi graph. However, when considering the SHD metric for the Erdos-

Renyi graph, MissDAG with NOTEARS achieved the best performance. It is worth noting that this superior performance was influenced by the extremely sparse estimates obtained by NOTEARS. As mentioned in the previous paragraph, the selection of the `w_threshold` parameter aimed to ensure acyclicity across all 20 datasets. This selection process resulted in some estimates being excessively sparse, which can artificially decrease the SHD metric. This issue is illustrated in Figure A.1 in the Appendix. Overall, the results demonstrate the favorable performance of FROSTY in both the MCAR and MAR missing mechanisms across all levels of missing rates, highlighting its practicality in real-world scenarios.



(a) Accuracy metrics of SHD and MCC with varying missing rates under MCAR mechanism



(b) Accuracy metrics of SHD and MCC with varying missing rates under MAR mechanism

Figure 3.4: Comparison of accuracy metrics for MissDAG with various methods. Each graph is of size  $p = 20$  and of sample size  $n = 200$ .

## Chapter 4

# ACCORD for Sparse Precision Matrix Estimation

In this chapter, we propose *Asymmetric ConCORD* (ACCORD), an efficient method for estimating sparse precision matrices based on pseudo-likelihood. ACCORD not only guarantees descent in the objective values but also exhibits linear convergence in both the objective values and iterates. Additionally, ACCORD can achieve even faster convergence by employing bounded backtracking line search during its substep. To demonstrate the advantages of ACCORD, we conduct numerical experiments comparing it with other pseudo-likelihood based estimation methods such as CONCORD [37] and SPACE [38], and a well-known Gaussian likelihood-based method called graphical lasso [27]. In addition, we implement a high-performance variant of ACCORD, named HP-ACCORD, which allows the method to scale up to 1 million variables utilizing communication-avoiding distributed matrix multiplication. The experimental results highlight the aforementioned characteristics of ACCORD and HP-ACCORD, and we also observe that the bounded backtracking line search enhances the numerical stability of ACCORD.

This chapter is a joint work with Sungdong Lee and Joong-Ho Won from the Depart-

ment of Statistics at Seoul National University, and Youngra Kim and Hyungwon Choi from the Department of Medicine at the National University of Singapore.

## 4.1 Background

The precision matrix, also known as the inverse covariance matrix or concentration matrix, holds significant importance in multivariate statistical analysis. Its entries correspond to partial correlations, which provide a measure of the direct association between two variables while considering the effects of other variables. Eliminating the influence of confounding variables allows for a more accurate assessment of the specific relationship between two entities. This characteristic makes the precision matrix particularly valuable in various domains, including biology, genetics, finance, and many others, where understanding direct relationships is of utmost interest.

In particular, sparse precision matrix estimation is of significant interest because sparsity not only allows computationally efficient estimation but it is also a reasonable assumption for many real-world networks. For instance, gene regulatory networks are widely acknowledged as sparse, where only a subset of genes exert influence on the expression of other genes. Similarly, social networks and brain networks also exhibit sparsity, as only a limited number of connections exist between individuals or brain regions. In the field of sparse precision matrix estimation, several approaches have been proposed, ranging from maximum likelihood methods to regression or pseudo-likelihood approaches [39, 40, 27, 38, 37].

In this study, we focus on the pseudo-likelihood approach, which offers notable advantages over maximum likelihood methods. Firstly, the pseudo-likelihood approach provides computational benefits, especially for large-scale networks, as it avoids the need to compute the joint likelihood. Secondly, the pseudo-likelihood approach relaxes the

assumption of multivariate Gaussianity, enhancing the model's robustness to potential misspecification. In the subsequent sections, we introduce ACCORD, a novel method that builds upon the pseudo-likelihood framework, and establish its connection to CONCORD, another prominent pseudo-likelihood based method in the literature.

## 4.2 Pseudo-Likelihood and CONCORD

Consider a  $p$ -variate Gaussian random vector  $\mathbf{x} \in (x_1, \dots, x_p)^T$  and its precision matrix  $\Theta = (\theta_{ij})$ . It can be demonstrated that the negative pseudo-log-likelihood (excluding the constant term) is expressed as

$$-\prod_{i=1}^p \log p(x_i | \mathbf{x}_{-i}) = -\frac{1}{2} \sum_{i=1}^p \log \theta_{ii} + \frac{1}{2} \sum_{i=1}^p \theta_{ii}^{-1} \left( \sum_{j=1}^p \theta_{ij} x_j \right)^2, \quad \theta_{ij} = \theta_{ji}. \quad (4.1)$$

Here,  $p(x_i | \mathbf{x}_{-i})$  denotes the conditional Gaussian density of  $x_i$  given the remaining variables, i.e.,  $\mathbf{x}_{-i} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_p)^T$  is a vector with the  $i^{\text{th}}$  coordinate omitted. Although the negative pseudo-log-likelihood function in equation (4.1) is convex with respect to  $\Theta$ , optimizing the objective function involving the fractional-quadratic terms is not straightforward.

Now let us consider the objective function of CONCORD given by

$$-\sum_{i=1}^p \log \omega_{ii} + \frac{1}{2} \sum_{i=1}^p \left( \omega_{ii} x_i + \sum_{j \neq i} \omega_{ij} x_j \right)^2, \quad \omega_{ij} = \omega_{ji}, \quad \omega_{ii} > 0. \quad (4.2)$$

If we set  $\omega_{ij} = \frac{\theta_{ij}}{\sqrt{\theta_{ii}}}$ , then

$$\begin{aligned} -\sum_{i=1}^p \log \omega_{ii} + \frac{1}{2} \sum_{i=1}^p (\omega_{ii} x_i + \sum_{j \neq i} \omega_{ij} x_j)^2 &= -\frac{1}{2} \sum_{i=1}^p \log \theta_{ii} + \frac{1}{2} \sum_{i=1}^p (\sqrt{\theta_{ii}} x_i + \sum_{j \neq i} \frac{\theta_{ij}}{\sqrt{\theta_{ii}}} x_j)^2 \\ &= -\frac{1}{2} \sum_{i=1}^p \log \theta_{ii} + \frac{1}{2} \sum_{i=1}^p \theta_{ii} (x_i + \sum_{j \neq i} \frac{\theta_{ij}}{\theta_{ii}} x_j)^2 \\ &= -\frac{1}{2} \sum_{i=1}^p \log \theta_{ii} + \frac{1}{2} \sum_{i=1}^p \theta_{ii}^{-1} (\sum_{j=1}^p \theta_{ij} x_j)^2. \end{aligned}$$

In other words, the objective function of CONCORD is a reparametrization of the negative pseudo-log-likelihood with  $\theta_{ij} = \omega_{ii} \omega_{ij}$ . Note that the symmetry of  $\Theta$  translates into the following hidden nonlinear constraints:

$$\omega_{ii} \omega_{ij} = \omega_{jj} \omega_{ji}. \quad (4.3)$$

Now we can see that there is a one-to-one correspondence between (4.1) and

$$-\sum_{i=1}^p \log \omega_{ii} + \frac{1}{2} \sum_{i=1}^p (\omega_{ii} x_i + \sum_{j \neq i} \omega_{ij} x_j)^2, \quad \omega_{ii} \omega_{ij} = \omega_{jj} \omega_{ji}, \quad \omega_{ii} > 0. \quad (4.4)$$

Thus, the original CONCORD problem relaxes the nonlinear constraints (4.3) to the linear constraints of  $\omega_{ij} = \omega_{ji}$ , which implicitly assumes  $\omega_{ii} = \omega_{jj}$ , or equivalently,  $\theta_{ii} = \theta_{jj}$ .

CONCORD demonstrates consistency as a model selection estimator, indicating its ability to recover the sparsity pattern of the precision matrix in the asymptotic regime. However, due to its linear relaxation, CONCORD does not yield a transformed version of the precision matrix  $\Theta$ . In certain domains, this limitation may be undesirable, particularly when researchers aim to estimate precise partial correlations while also considering the presence of nonzero partial correlations.

### 4.3 ACCORD, the Asymmetric CONCORD

Unlike CONCORD, ACCORD directly tackles the objective function presented in (4.4) while considering the nonlinear constraints (4.3). In matrix form, (4.4) can be expressed as follows:

$$\ell(\mathbf{x}; \Omega) = -\log |\Omega_D| + \frac{1}{2} \text{trace}(\Omega^T \Omega \mathbf{x} \mathbf{x}^T),$$

or, in terms of the population version,

$$\ell(\Sigma, \Omega) = -\log |\Omega_D| + \frac{1}{2} \text{trace}(\Omega^T \Omega \Sigma), \quad (4.5)$$

where  $\Omega_D$  and  $\Sigma$  represent the diagonal matrix of  $\Omega$  and the population covariance matrix, respectively, with  $\Omega_D = \text{diag}(\omega_{11}, \dots, \omega_{pp})$  and  $\Sigma = \Theta^{-1} = E[\mathbf{x} \mathbf{x}^T]$ . (4.5) is clearly convex with respect to  $\Omega$  since the negative log determinant function,  $-\log |X|$ , is convex when  $X$  is a positive definite matrix, and the trace term is quadratic in  $\Omega$ .

The gradient of  $\ell(\Sigma, \Omega)$  is given by

$$\nabla \ell(\Sigma, \Omega) = -\Omega_D^{-1} + \Omega \Sigma,$$

and its minimizer can be derived from the first-order optimality condition,

$$\Omega^* = \Omega_D^{-1} \Sigma^{-1} = \Theta_D^{-\frac{1}{2}} \Theta.$$

If  $\omega_{ii} > 0 \ \forall i$ , then the minimizer is unique since the Hessian matrix is positive definite, i.e.,

$$\nabla^2 \ell(\Sigma, \Omega) = -\Omega_D^{-1} \Omega_D^{-1} + \Sigma \succ \mathbf{0} \quad (4.6)$$

Therefore,  $\Omega^*$  is the unique minimizer of the objective function  $\ell(\Sigma, \Omega)$ . Subsequently, the minimizer  $\Omega^*$  can be scaled to obtain the precision matrix  $\Theta$ , which is *not* satisfied with the original CONCORD objective function.

If  $\hat{\Omega}$  is the consistent minimizer of the sample version of the objective function  $\ell(S; \Omega)$ , where  $S = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^T$  is the sample covariance matrix, then the precision matrix can be consistently estimated as well by using the relation

$$\hat{\theta}_{ij} = \hat{\omega}_{ii} \hat{\omega}_{ij}.$$

Furthermore, since  $\Omega^*$  satisfies the hidden constraints (4.3), specifically  $\omega_{ii}^* \omega_{ij}^* = \omega_{jj}^* \omega_{ji}^*$ , by construction, the consistency of  $\hat{\Omega}$  ensures that  $\hat{\omega}_{ii} \hat{\omega}_{ij} = \hat{\omega}_{jj} \hat{\omega}_{ji}$  holds asymptotically. In the finite case, the symmetry of the estimated precision matrix can be assured by one of the following two rules from [39]:

$$\tilde{\Theta}^\wedge = \frac{1}{2} M^\wedge \circ (\hat{\Omega}_D \hat{\Omega} + \hat{\Omega}^T \hat{\Omega}_D) \quad (4.7)$$

$$\tilde{\Theta}^\vee = \frac{1}{2} M^\vee \circ (\hat{\Omega}_D \hat{\Omega} + \hat{\Omega}^T \hat{\Omega}_D) \quad (4.8)$$

where  $M^\wedge$  and  $M^\vee$  are masking matrices consisting of 1s and 0s.  $M^\wedge$  follows an *and* rule where its  $(i, j)^{\text{th}}$  entry takes 1 if both  $\hat{\omega}_{ij}$  and  $\hat{\omega}_{ji}$  are nonzeros, and 0 otherwise. On the other hand,  $M^\vee$  follows a *or* rule where its  $(i, j)^{\text{th}}$  entry takes 1 if  $\hat{\omega}_{ij}$  or  $\hat{\omega}_{ji}$  is nonzero, and 0 otherwise. Note that both  $\tilde{\Theta}^\wedge$  and  $\tilde{\Theta}^\vee$  are consistent.

## 4.4 Optimization for ACCORD

### 4.4.1 The ACCORD Objective

In this section, we propose an optimization algorithm that minimizes the  $\ell_1$ -regularized objective function

$$\ell(\Omega) = \ell(S; \Omega) = -\log |\Omega_D| + \frac{1}{2} \text{trace}(\Omega^T \Omega S) + \lambda \|\Omega\|_1 \quad (4.9)$$

where  $\|\Omega\|_1 = \sum_{i,j} |\omega_{ij}|$ . The domain of  $\ell(\Omega)$  is defined as  $\{\Omega \in \mathbb{R}^{p \times p} : \omega_{ii} > 0, i = 1, \dots, p\}$ . Outside the domain, we set  $-\log |\Omega_D| = \infty$ . Considering the convexity of the last term in (4.9) alongside the previously established convexity of the sum of the first two terms, we conclude that  $\ell(\Omega)$  is a strictly convex function. Therefore, the ACCORD problem (4.9) can be formulated as a convex optimization problem.

To ensure the validity of our approach, we introduce the following assumption:

**Assumption 1.** *The sample covariance matrix  $S$  of the data  $\mathbf{x}_1, \dots, \mathbf{x}_p$  has strictly positive diagonal entries.*

While the objective function of ACCORD is strictly convex, caution must be exercised due to the strictly decreasing nature of the negative logarithm, which renders  $\ell(\Omega)$  non-coercive in general. However, if all diagonal entries of the sample covariance matrix  $S$  are strictly positive, then  $\ell(\Omega)$  becomes coercive and has a unique minimizer [41]. This condition is almost surely satisfied when none of the  $p$  marginal distributions associated with the  $p$ -variate distribution of the data vector  $\mathbf{x}$  is degenerate [37, Remark 5]. Hence, we adopt Assumption 1 throughout the remainder of this chapter.

### 4.4.2 Proximal Gradient Algorithm with Alternative Splitting

Imposing symmetry on  $\Omega$ , [42] propose the CONCORD-ISTA algorithm as a solution for minimizing (4.9). CONCORD-ISTA, based on the iterative shrinkage-thresholding algorithm (ISTA) [43], is a proximal gradient algorithm designed for minimizing a combination of continuously differentiable convex functions and possibly nondifferentiable closed convex functions. Specifically, CONCORD-ISTA splits  $\ell(\Omega)$  into  $\bar{f}(\Omega) + \bar{g}(\Omega)$  where

$$\bar{f}(\Omega) = -\log |\Omega_D| + \frac{1}{2} \text{trace}(\Omega^T \Omega S), \quad \bar{g}(\Omega) = \lambda \|\Omega\|_1. \quad (4.10)$$

Applying proximal gradient with step size  $\tau_t$ , we obtain an iterative update

$$\begin{aligned} \Omega^{(t+1)} &= \arg \min_{\Omega} \langle \nabla \bar{f}(\Omega^{(t)}), \Omega - \Omega^{(t)} \rangle + \frac{1}{\tau_t} \|\Omega - \Omega^{(t)}\|_F^2 + \bar{g}(\Omega) \\ &= \text{prox}_{\tau_t \bar{g}}(\Omega^{(t)} - \tau_t \nabla \bar{f}(\Omega^{(t)})). \end{aligned}$$

The proximity operator can be computed in a closed form, yielding the following expressions:

$$\begin{aligned} \omega_{ii}^{(t+1)} &= \omega_{ii}^{(t)} - \tau_t \left[ -\frac{1}{\omega_{ii}^{(t)}} + \frac{(S\Omega^{(t)})_{ii}}{2} + \frac{(\Omega^{(t)}S)_{ii}}{2} \right], \\ \omega_{ij}^{(t+1)} &= \mathcal{S}_{\tau_t \lambda}(\omega_{ij}^{(t)} - \tau_t \left[ \frac{(S\Omega^{(t)})_{ij}}{2} + \frac{(\Omega^{(t)}S)_{ij}}{2} \right]), \quad i \neq j. \end{aligned}$$

where  $\mathcal{S}_\lambda(x) = \text{sign}(x) \max\{|x| - \lambda, 0\}$  is the soft-thresholding operator.

However, the splitting of CONCORD-ISTA into a “loss + penalty” formulation poses challenges in selecting an appropriate step size  $\tau_t$  due to the lack of global Lipschitz continuity of  $\nabla \bar{f}$ . Although  $\nabla \bar{f}$  exhibits Lipschitz continuity within a compact domain, particularly within the sublevel set  $\{\Omega : \ell(\Omega) \leq \ell(\Omega^{(0)})\}$  of the initial objective value, the absence of an absolute Lipschitz constant presents two issues: (1) it renders line

search not an option but a necessity, and (2) the step sizes chosen by line search depend on the initial value and can be arbitrarily small. The convergence rate of CONCORD-ISTA for the objective value follows a rate of  $O(1/t)$ , or  $O(1/t^2)$  when utilizing Nesterov acceleration [44] at the expense of losing the descent property. However, the specific convergence rate of the iterates  $\{\Omega^{(t)}\}$  towards the minimizer  $\hat{\Omega}$  is currently unknown and warrants further investigation.

In the case of ACCORD, we employ an alternative splitting approach that uses forward-backward splitting while also dropping the symmetry assumption, which we call ACCORD-FBS. Specifically, we redefine the splitted objective functions as follows:

$$f(\Omega) = \frac{1}{2}\text{trace}(\Omega^T \Omega S), \quad g(\Omega) = -\log |\Omega_D| + \lambda \|\Omega\|_1. \quad (4.11)$$

By applying proximal gradient optimization with a step size denoted as  $\tau_t$ , we derive a closed-form update rule:

$$\begin{aligned} \omega_{ii}^{(t+1)} &= \frac{y + \sqrt{y^2 + 4\tau_t}}{2}, \quad y = \omega_{ii}^{(t)} - \tau_t[\Omega^{(t)}S]_{ii}, \\ \omega_{ij}^{(t+1)} &= \mathcal{S}_{\tau_t\lambda}(\omega_{ij}^{(t)} - \tau_t[\Omega^{(t)}S]_{ij}), \quad i \neq j \end{aligned} \quad (4.12)$$

as described in [45]. Since the gradient  $\nabla f(\Omega) = \Omega S$  is globally Lipschitz continuous with constant  $L = \|S \otimes I\|_2 = \sigma_{\max}(S)$ , we can ensure the descent property and convergence of the algorithm towards the minimizer [46, 41] by selecting the step size in the range of

$$\tau_t \in (0, 2/L). \quad (4.13)$$

Hence, no line search is necessary. Furthermore, the new splitting provides an important practical benefit in that a wide range of the step sizes  $\tau_t$  exhibits a *linear convergence* of the algorithm, in both the objective values and the iterates, despite lack of strong

---

**Algorithm 3:** ACCORD-FBS algorithm
 

---

**Input** : Dataset  $X$ , regularization parameter  $\lambda$ , threshold  $\epsilon$   
**Output**: Precision matrix  $\Theta$

```

1 Initialize  $\Omega^{(0)} = I, \tau_0 = 1, \beta \in (0, 1)$ .
2 Set  $S = (X^T X)/n$  and  $L = \sigma_{\max}(S)$ .
3 for  $t = 0, 1, 2, \dots$  do
4      $\nabla f(\Omega^{(t)}) = \Omega^{(t)} S$ .
5     for  $\tau = \tau_0, \beta\tau_0, \beta^2\tau_0, \dots$  do
6          $y = \text{diag}(\Omega^{(t)}) - \tau \text{diag}(\nabla f(\Omega^{(t)}))$ .
7          $\Omega^{(t+1)} = \mathcal{S}_{\tau\lambda}(\Omega^{(t)} - \tau \nabla f(\Omega^{(t)}))$ .
8         Set the diagonal of  $\Omega^{(t+1)}$  to  $(y + \sqrt{y^2 + 4\tau})/2$ .
9          $\Delta = \Omega^{(t+1)} - \Omega^{(t)}$ .
10        if  $f(\Omega^{(t+1)}) \leq f(\Omega^{(t)}) + \langle \nabla f(\Omega^{(t)}), \Delta \rangle + \frac{1}{2\tau} \|\Delta\|_F^2$  or  $\tau \leq 1/L$  then
11            | break.
12        end
13    end
14    if  $\|\Delta\|_F < \epsilon$  then
15        | break.
16    end
17 end
18  $\Omega = \Omega^{(t+1)}$ .
19  $\Theta = (\Omega_D \Omega + \Omega^T \Omega_D)/2$ .
20 return  $\Theta$ 
    
```

---

convexity.

One potential issue arises when computing  $L = \sigma_{\max}(S)$  becomes challenging, particularly in scenarios with a large number of variables  $p$ . This challenge may result in the selection of an overly small step size. To address this concern, we employ a backtracking line search with a fixed decay rate of  $\beta \in (0, 1)$  that is bounded from below by  $1/L$ . For a detailed description of the ACCORD-FBS algorithm, refer to Algorithm 3. The algorithm ensures that  $\tau_t \geq \min\{\tau_0, \beta/L\}$  and  $\ell(\Omega^{(t+1)}) \leq \ell(\Omega^{(t)})$ . If  $\tau_0 \geq 1/L$ , which holds in typical high-dimensional settings, then  $\tau_t \geq 1/L$ .

**Theorem 5.** *Under Assumption 1, the iterate  $\{\Omega^{(t)}\}$  generated by the proximal gradient*

algorithm 3 with a step size sequence  $\{\tau_t\}$  satisfying either i)  $\tau_t \in [\underline{\tau}, \bar{\tau}]$  for all  $t$ , where  $0 < \underline{\tau} \leq \bar{\tau} < 2/L$  or ii) that  $\tau_t$  is determined by the backtracking line search with  $\tau_0$  chosen independent of  $t$ , converges to the unique minimizer  $\hat{\Omega}$  of the ACCORD objective function 4.9 as  $t \rightarrow \infty$ . The objective value sequence  $\{\ell(\Omega^{(t)})\}$  converges to its minimum  $\ell^* = \ell(\hat{\Omega})$  monotonically. Furthermore, if the initial iterate  $\Omega^{(0)}$  is chosen such that  $\ell(\Omega^{(0)}) \leq \text{trace}(S)/2$ , there holds,

$$\begin{aligned} \ell(\Omega^{(t)}) - \ell^* &\leq \left( \frac{1}{1 + 2a\sigma} \right)^t [\ell(\Omega^{(0)}) - \ell^*], \quad t \geq 0, \\ \|\Omega^{(t)} - \hat{\Omega}\|_F &\leq \left( \frac{1}{\sqrt{1 + 2a\sigma}} \right)^{t-1} \left( 1 + \frac{1}{a\sigma\sqrt{1 + \frac{1}{2a\sigma}}} \right) \sqrt{\frac{\ell(\Omega^{(0)}) - \ell^*}{a}}, \quad t \geq 1 \end{aligned} \quad (4.14)$$

where

$$a = \frac{1}{\bar{\tau}} - \frac{L}{2} > 0, \quad \sigma = \frac{1}{4\kappa(1/\underline{\tau} + L)^2}$$

if  $\tau_t \in [\underline{\tau}, \bar{\tau}]$ , and

$$a = \frac{1}{2\tau_0}, \quad \sigma = \frac{1}{4\kappa(1/\tau_{\min} + L)^2}, \quad \tau_{\min} = \min\{\tau_0, \beta/L\}$$

if backtracking line search is used to determine  $\tau_t$ . The constant  $\kappa$  is explicit and only depends on the data  $S$ .

Several remarks are in order.

**Remark 1.** An explicit formula for  $\kappa$  is given in the proof, as Equation (C.2).

**Remark 2.** If we take a fixed step size  $\tau_t = 1/L$  for all  $t$ , then  $a = L/2$  and  $\sigma = (16\kappa L^2)^{-1}$ .

**Remark 3.** Splitting (4.11) generating Algorithm 3 is crucial in that the Lipschitz constant  $L$  is finite and fixed over  $\text{dom}(\ell)$ . On the other hand, with splitting (4.10), the

gradient of  $\bar{f}$  is Lipschitz only within a compact subset of the domain and the constant varies with the subset.

**Remark 4.** *The proof of Theorem 5 establishes a global version of the Kurdyka-Łojasiewicz (KL) inequality with the Łojasiewicz exponent  $1/2$ . This leads to the nonasymptotic complexity bounds (4.14). As long as the initial iterate  $\Omega^{(0)}$  satisfies  $\ell(\Omega^{(0)}) \leq \text{trace}(S)/2$ , e.g.,  $\Omega^{(0)} = \text{diag}(\text{diag}(S))/2$ , then linear convergence occurs (If  $\ell(\Omega^{(0)}) > \text{trace}(S)/2$ , then the iterates enter the region of linear convergence after a finite number of iterations). This is an improvement of from [47, Proposition 4.1], which establishes a local KL inequality for a more general class of objective functions including  $\ell$ . Only a locally linear rate of convergence, or existence of a linear convergence region, can be deduced from this result.*

## 4.5 HP-ACCORD, ACCORD with High-Performance Computing

We present the extension of ACCORD optimized for high-performance parallel computing in distributed memory settings, referred to as HP-ACCORD. The framework builds upon the communication-avoiding linear algebra algorithms proposed by [48] and implements parallelization strategies for ACCORD. HP-ACCORD comprises two distinct variants: the *Cov variant* and the *Obs variant*.

The Cov variant initiates by computing the sample covariance matrix  $S = (X^T X)/n$  in line 2 of Algorithm 3 once upfront via a distributed dense-dense matrix multiplication. The gradient iterate  $\nabla f(\Omega^{(t)}) = \Omega^{(t)} S$  is computed through a distributed sparse-dense matrix multiplication within every proximal gradient iteration. Then, various other significant computations such as  $f(\Omega^{(t)})$ ,  $f(\Omega^{(t+1)})$ ,  $\langle \nabla f(\Omega^{(t)}), \Delta \rangle$  and the term

$-\log |\Omega_D^{(t+1)}|$  for objective value tracking, are computed in an embarrassingly parallel way since  $\langle A, B \rangle = \text{trace}(A^T B) = \sum_{i,j} A_{ij} B_{ij}$  and  $\log |D| = \sum_i \log D_{ii}$  for any diagonal matrix  $D$ .

However, in scenarios where  $p$  is extremely large, the upfront cost of computing the dense sample covariance matrix  $S$  may not be worth it. In response, the Obs variant avoids explicitly computing  $S$ . Instead, this variant computes  $Y^{(t)} = \frac{1}{n} \Omega^{(t)} X^T$  in each proximal gradient iteration and performs gradient computation through  $\nabla f(\Omega^{(t)}) = Y^{(t)} X$ , utilizing distributed sparse-dense and dense-dense matrix multiplications, respectively. Moreover, by exploiting the relationship  $f(\Omega^{(t)}) = \frac{1}{2n} \text{trace}((\Omega^{(t)})^T \Omega^{(t)} X^T X) = \frac{n}{2} \text{trace}((Y^{(t)})^T Y^{(t)}) = \frac{n}{2} \|Y^{(t)}\|_F^2$ , the computation of  $f(\Omega^{(t)})$  also lends itself to an embarrassingly parallel approach. However, the Obs variant scales poorly in comparison to the Cov variant with respect to the number of proximal gradient iterations. For a more comprehensive cost comparison between the two variants, refer to [48, Lemma 3.1].

## 4.6 Numerical Experiments

### 4.6.1 Linear Convergence

In this section, we provide empirical evidence of the guaranteed descent property and linear convergence exhibited by ACCORD-FBS. To demonstrate the effectiveness of its forward-backward splitting over the “loss + penalty” splitting of the ISTA algorithm, we additionally implement ACCORD-ISTA and compare their convergence behaviors. Firstly, we generate two Erdos-Renyi graphs of size  $p = 1000$  with sparsity levels of 3% and 15%. To construct ground truth precision matrices, we employ an identical procedure for each graph. The edge weights are selected from a  $\text{Uniform}(0.5, 1)$  distribution, where each weight’s sign is equally likely to be positive or negative with a probability of 0.5. To

ensure symmetry and positive definiteness, we add the matrix to its transpose and set each row's diagonal entry to be 1.5 times the sum of magnitudes of off-diagonal entries. Then, we scale the matrix by pre- and post-multiplying it with a diagonal matrix such that all diagonal entries are equal to one. Furthermore, we introduce variation among the diagonal entries by performing another pre- and post-multiplication with a diagonal matrix whose entries follow a uniform distribution between  $[1, \sqrt{3}]$ . This resulting matrix serves as the precision matrix. Subsequently, we generate multivariate Gaussian data with a sample size of  $n = 500$ , given the specified precision matrix.

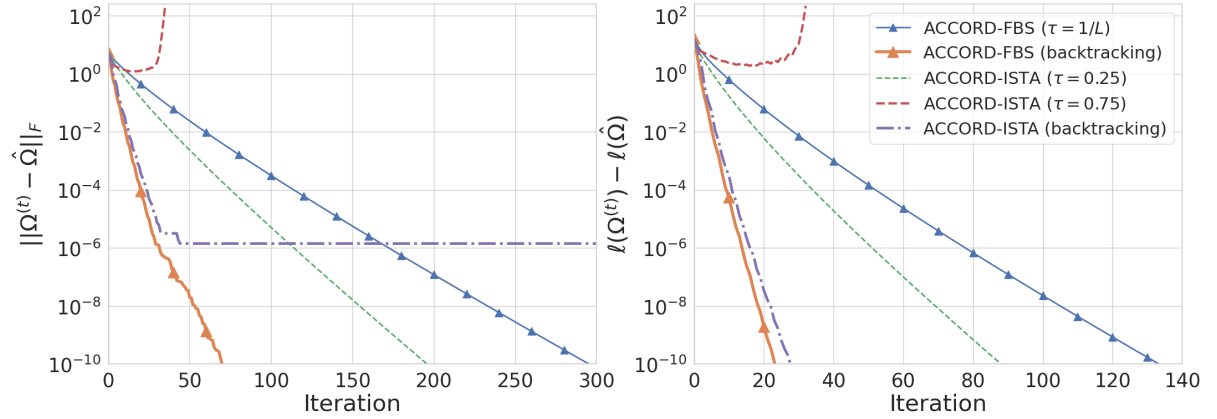
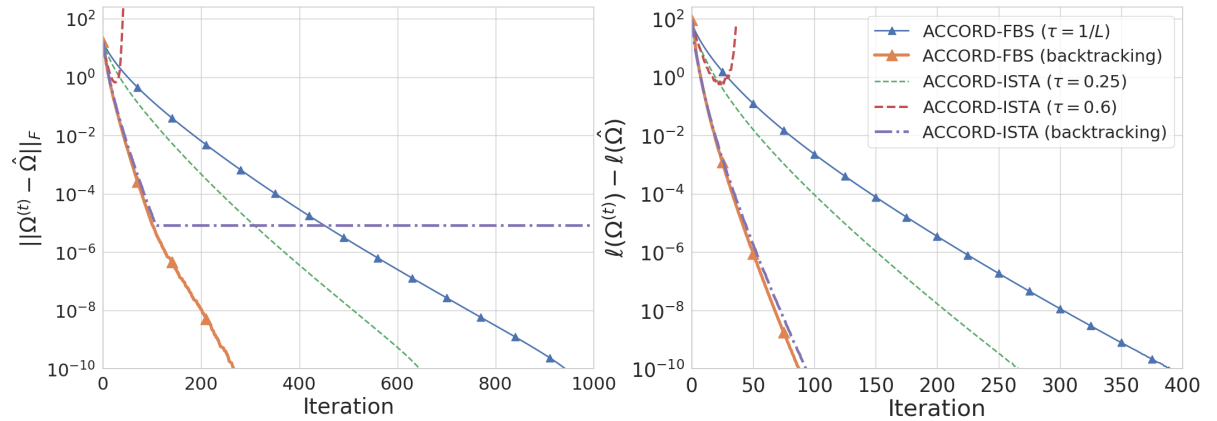

 (a)  $\Theta$  with 3% sparsity level

 (b)  $\Theta$  with 15% sparsity level

Figure 4.1: Convergence comparison between ACCORD-FBS and ACCORD-ISTA

The convergence behavior of  $\|\Omega^{(t)} - \hat{\Omega}\|_F$  and  $\ell(\Omega^{(t)}) - \ell(\hat{\Omega})$  is illustrated in Figure

4.1, with the left-hand and right-hand sides showcasing the results, respectively. Here,  $\hat{\Omega}$  denotes the final estimated solution obtained by executing ACCORD-FBS until convergence, with a threshold criterion of  $\|\Omega^{(t+1)} - \Omega^{(t)}\| < 10^{-15}$  to account for numerical precision. ACCORD-FBS is investigated with two variants: one with a constant step size of  $\tau = \frac{1}{L}$  and the other employing backtracking line search with  $\tau = \frac{1}{L}$  as a lower bound. In contrast, a grid of constant step sizes,  $\tau \in \{0.25, 0.6, 0.75\}$ , along with backtracking, are chosen for ACCORD-ISTA to demonstrate its convergence behavior.

In Figure 4.1, ACCORD-FBS exhibits linear convergence for a constant step size of  $\tau = \frac{1}{L}$ , and even faster convergence when backtracking line search with lower bound is applied. On the other hand, the convergence behavior of ACCORD-ISTA varies significantly across different constant step sizes, ranging from slow convergence to divergence. Moreover, ACCORD-ISTA with backtracking line search encounters numerical instability, leading to a plateau in its convergence at an early stage. These observations underscore the challenges associated with the selection of an appropriate step size, which in turn highlights the advantages of ACCORD-FBS over ACCORD-ISTA.

## 4.6.2 Numerical Stability

Recall that while the ACCORD-FBS algorithm exhibited linear convergence, the ACCORD-ISTA algorithm with backtracking line search suffered from a plateau phenomenon, as depicted in Figure 4.1. This phenomenon stems from the numerical instability involved in checking the descent condition, which necessitates computing  $Q$  in line 10 of Algorithm 3. Let us delve deeper into the underlying dynamics. In Figure 4.2, we observe that the objective value of ACCORD-ISTA experiences its first increment at the 33<sup>rd</sup> outer iteration, represented by the vertical dotted red line. Interestingly, this coincides with a significant surge in the number of inner iterations required by ACCORD-

ISTA's backtracking line search. This presents a paradox: although ACCORD-ISTA successfully escaped the backtracking line search step, indicating that it found a step size satisfying the descent condition, the objective value did not descend. Furthermore, this is when ACCORD-ISTA starts to experience a plateau in its convergence. The computation of  $Q$  involves multiple matrix multiplications, which can lead to the accumulation of numerical errors. While ACCORD-FBS also computes  $Q$  in its backtracking line search, it incorporates a lower bound on the step size, serving as an additional descent condition that helps mitigate numerical instability.

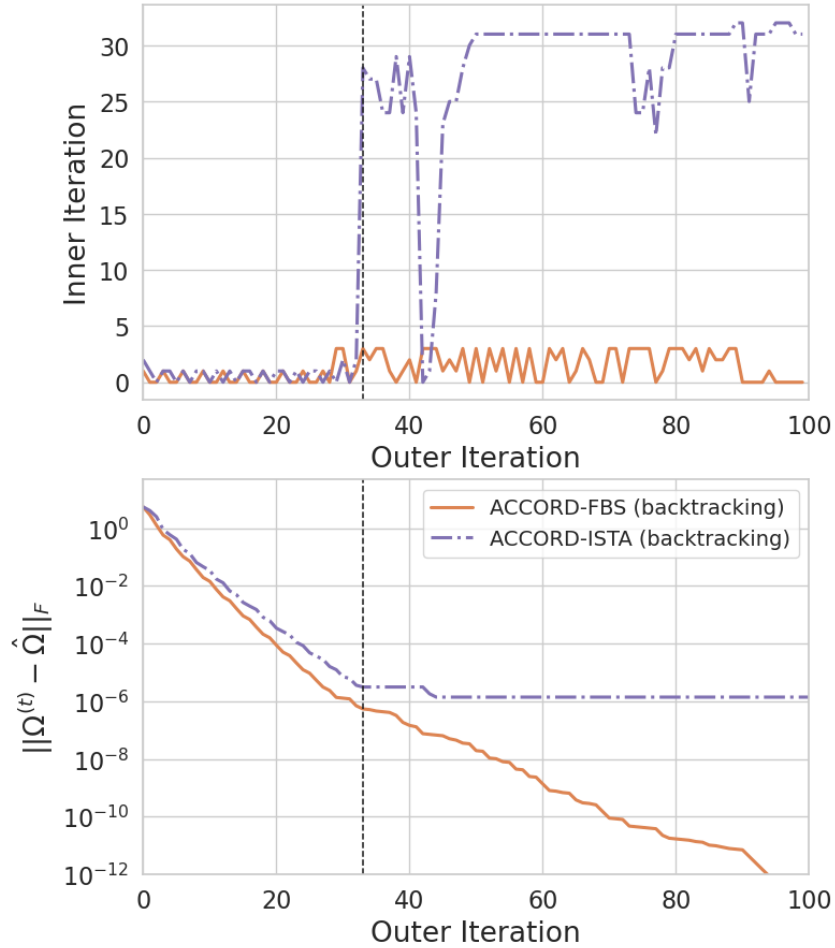


Figure 4.2: Numerically unstable behavior of the ACCORD-ISTA algorithm. The vertical dashed line represents the first outer iteration where the objective value of ACCORD-ISTA increases.

### 4.6.3 Pseudo-Likelihood-Based BIC for Model Selection

Throughout the rest of simulations, we use a pseudo-likelihood-based BIC (pBIC) to select an appropriate level of  $\ell_1$ -regularization for ACCORD-FBS. The complete joint pseudo-likelihood with  $k = 1, \dots, n$  samples can be expressed in terms of the residual sum of squares (RSS) as

$$\begin{aligned}
 \ell(\Theta; x) &= \log \prod_{i=1}^p \prod_{k=1}^n P(X_i = x_i | X_{-i} = x_{-i}) \\
 &= \sum_{i=1}^p \sum_{k=1}^n \log P(X_i = x_i | X_{-i} = x_{-i}) \\
 &= \sum_{i=1}^p \left( \frac{n}{2} \log \theta_{ii} - \frac{\theta_{ii}}{2} \sum_{k=1}^n \left( \sum_{j=1}^p \theta_{ii}^{-1} \theta_{ij} x_j^k \right)^2 \right) \\
 &= \frac{1}{2} \sum_{i=1}^p (n \log \theta_{ii} - \theta_{ii} \text{RSS}_i).
 \end{aligned}$$

Hence, the BIC based on the pseudo-likelihood can be computed by

$$\text{pBIC} = m \log n - 2\ell(\hat{\Theta}; x) = m \log n + \sum_{i=1}^p (\hat{\theta}_{ii} \text{RSS}_i - n \log \hat{\theta}_{ii}) \quad (4.15)$$

where  $m$  is the number of parameters, i.e., the number of edges in  $\hat{\Theta}$ . To introduce enhanced flexibility to ACCORD-FBS, we explore an extended version of the pBIC inspired by [49]. The extended pBIC (epBIC) writes

$$\text{epBIC} = m \log n - 2\ell(\hat{\Theta}; x) + 4m\gamma \log p \quad (4.16)$$

where  $\gamma \in [0, 1]$  serves as a tuning parameter.

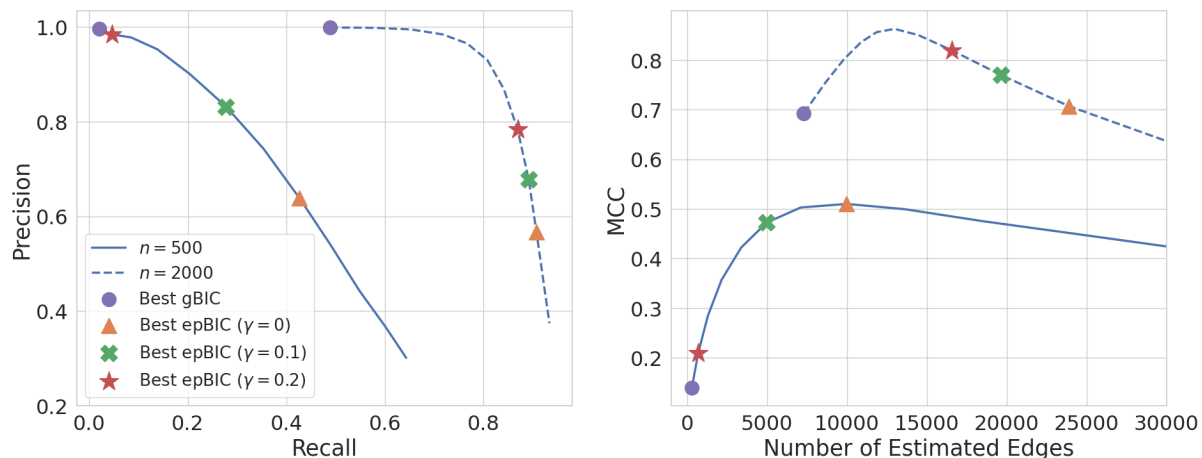


Figure 4.3: Behaviors of different model selection criteria.

Figure 4.3 illustrates the behavior of various model selection criteria when applied to ACCORD-FBS. For this simulation, we employed an Erdos-Renyi random graph with a dimension of  $p = 1000$  and two sample sizes of  $n = 500$  and  $2000$ . Notably, Gaussian BIC, denoted as gBIC, identifies the sparsest graph for both sample sizes with an exceedingly low recall for the small sample size. Conversely, pBIC and epBIC opt for denser graphs while maintaining a reasonable level of precision for both sample sizes. This trend is highlighted in the right-hand plot of Figure 4.3 where pBIC and epBIC achieve higher MCCs, a metric assessing balanced structural accuracy.

#### 4.6.4 Edge Detection and Parameter Estimation

Here, we examine the edge detection and parameter estimation performance of ACCORD, comparing it with other prominent methods including Glasso (graphical lasso), CONCORD, and SPACE. For this simulation, we consider two graph structures, each consisting of  $p = 1000$  nodes: a hub network and an Erdos-Renyi graph. The hub network structure is generated using the following procedure: (1) constructing a Barabasi-Albert scale-free graph, (2) randomly designating 5% of the nodes as hub nodes, and (3) form-

ing complete sub-graphs (cliques) around each hub node. For the Erdos-Renyi graph, we choose an edge probability of 1%. Once the graph structures are generated, the following procedure is applied to construct positive definite precision matrices. Firstly, random edge weights are uniformly selected from the range  $[-1, -0.5] \cup [0.5, 1]$  generating the initial precision matrix  $\Theta^{(0)}$ . Then,  $\Theta^{(0)}$  is iteratively projected onto the intersection of the following two closed convex sets:

$$C_1 = \{\Theta \in \mathbb{S}^p : \lambda_{\min}(\Theta) \geq \epsilon\}, \quad C_2 = \{\Theta \in \mathbb{S}^p : \theta_{ij} = 0 \text{ if } (i, j) \notin E\}$$

where  $\mathbb{S}^p$  represents the set of symmetric matrices and  $\epsilon = 0.4$  is chosen as the eigenvalue threshold for  $C_1$ . We stop the iterative procedure once the condition number of  $\Theta^{(t)}$  is smaller than 20. The required projection is computed using the projection onto convex sets (POCS) algorithm [50].

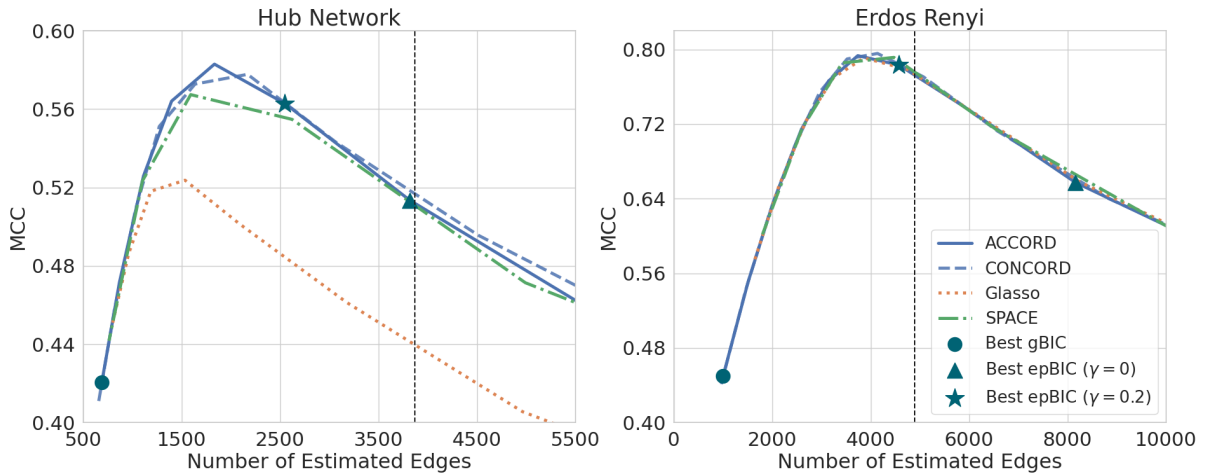


Figure 4.4: Edge detection performance comparison between various methods.

Figure 4.4 presents the MCC curves as the key results for edge detection where the *or* rule symmetrization is applied to ACCORD. In each plot, the vertical dashed line represents the number of edges in the true graph. The methods demonstrate indistin-

guishable performance on the Erdos-Renyi graph. However, when applied to the hub network, a more challenging graph structure to estimate due to its increased maximum degree, graphical lasso yields significantly reduced performance compared to other methods. Additionally, SPACE consistently produced curves that are slightly below those of ACCORD and CONCORD. On the other hand, no significant performance gap is observed between ACCORD and CONCORD for both graph structures.

Table 4.1: Parameter estimation performance comparison between various methods.

| Graph       | Method  | Total Error | Mean Squared Error |              |              |
|-------------|---------|-------------|--------------------|--------------|--------------|
|             |         |             | TP                 | FP           | FN           |
| Hub Network | ACCORD  | 11.01       | 0.195 (1964)       | 0.027 (2536) | 0.153 (1906) |
|             | CONCORD | 12.09       | 0.221 (2063)       | 0.017 (3598) | 0.156 (1807) |
|             | Glasso  | 15.50       | 0.325 (285)        | 0.016 (64)   | 0.242 (3585) |
|             | SPACE   | 15.54       | 0.334 (218)        | 0.015 (10)   | 0.244 (3652) |
| Erdos-Renyi | ACCORD  | 7.60        | 0.090 (2739)       | 0.026 (1301) | 0.126 (2158) |
|             | CONCORD | 7.31        | 0.097 (3577)       | 0.018 (3530) | 0.118 (1320) |
|             | Glasso  | 9.95        | 0.123 (460)        | 0.020 (79)   | 0.144 (4437) |
|             | SPACE   | 10.16       | 0.139 (129)        | 0.016 (1)    | 0.145 (4768) |

Table 4.1 summarizes errors in partial correlation estimation with total squared error and mean squared error across different estimation types. Here, the *and* rule symmetrization is applied to ACCORD. Figures in parentheses represent the number of corresponding estimations. Model selection was carried out using pBIC for regression-based methods such as ACCORD, CONCORD, and SPACE, while gBIC was used for graphical lasso, a Gaussian likelihood-based method. For the hub network, ACCORD attains the smallest total error in addition to significantly lower mean squared error for TP edges. Even for the Erdos-Renyi graph, where ACCORD’s total squared error is marginally higher compared to CONCORD, it remains to achieve the smallest mean squared error for TP edges.

### 4.6.5 Debiasing ACCORD for Unbiased Partial Correlations

The ACCORD estimates are prone to bias towards zero due to the lasso regularization. To address this issue and obtain unbiased partial correlations, we employ a post-selection inference. Firstly, we obtain an initial estimate of the precision matrix where the magnitude of  $\ell_1$ -regularization is selected based on the epBIC. Afterwards, we re-estimate ACCORD in which the nonzero entries in the initial estimate are not regularized while an arbitrarily large regularization is applied to the zero entries to ensure that the resulting estimate has the same zero/nonzero structure as the initial estimate. Finally, we compute the partial correlation estimates by scaling the final estimate of the precision matrix using the formula  $\rho_{ij} = -\frac{\theta_{ij}}{\sqrt{\theta_{ii}\theta_{jj}}}$ . This procedure mitigates the bias introduced by lasso regularization and allows us to obtain more accurate and unbiased partial correlation estimates.

The impact of debiasing ACCORD is demonstrated in Figure 4.5 for a hub network and an Erdos-Renyi graph. Each plot is a histogram showing the true partial correlations (T) and estimated partial correlations. Different colors are used to distinguish the estimated values at correctly detected nonzero locations (true positives or TP) and falsely detected nonzero locations (false positives or FP). The results clearly indicate that debiasing consistently improves the accuracy of the estimates at TP locations while effectively controlling the magnitude of the FP, ensuring it remains relatively small. This observation, as shown in Figure 4.5, encourages the prioritization of larger partial correlation estimates as they are more likely to be at TP locations, while considering the possibility of disregarding smaller estimates as they could correspond to FP locations.

Table 4.2 provides a summary of the structural accuracy and the squared Frobenius norm between the true partial correlation matrix and the estimated one for each graph. Firstly, we observe a significant reduction in the squared Frobenius norm between the

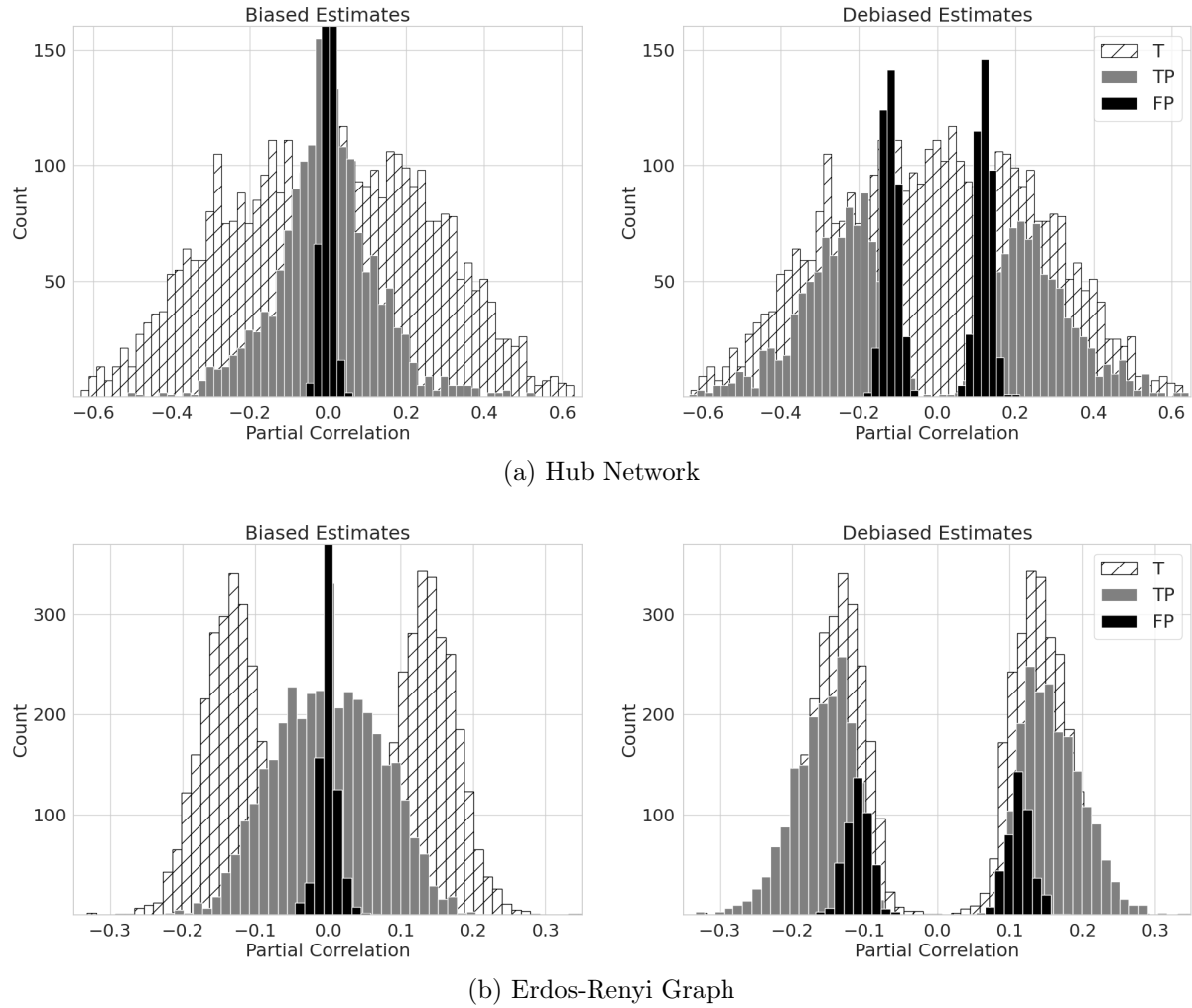


Figure 4.5: Distributions of partial correlation estimates of ACCORD

true partial correlation matrix and the debiased estimated partial correlation matrix, compared to the biased estimate. Note that the magnitude of incorrectly estimated partial correlations also increases with debiasing. However, when examining the TP and FP columns in Table 4.2, we find that the decrease in error for the TP column far exceeds the increase in error for the FP column. Additionally, Table 4.3 provides further insights by presenting the average squared Frobenius norm difference, calculated by dividing the total squared Frobenius norm by the number of nonzero estimates for each corresponding TP and FP. These tables with the histograms provide a comprehensive view of the overall improvement achieved through debiasing.

| Graph       | Structural Accuracy |           |        | $\sum_{i<j}(\rho_{ij} - \hat{\rho}_{ij,\text{biased}})^2$ |      |        | $\sum_{i<j}(\rho_{ij} - \hat{\rho}_{ij,\text{debiased}})^2$ |       |       |
|-------------|---------------------|-----------|--------|-----------------------------------------------------------|------|--------|-------------------------------------------------------------|-------|-------|
|             | MCC                 | Precision | Recall | TP                                                        | FP   | Total  | TP                                                          | FP    | Total |
| Hub Network | 0.55                | 0.68      | 0.45   | 100.37                                                    | 0.12 | 164.84 | 15.44                                                       | 12.01 | 91.80 |
| Erdos-Renyi | 0.78                | 0.81      | 0.76   | 36.65                                                     | 0.13 | 52.06  | 4.10                                                        | 11.32 | 30.71 |

Table 4.2: Accuracy and error metrics for biased and debiased partial correlation estimates of ACCORD. The summations are over estimates for corresponding TP or FP locations.

| Graph       | Structural Accuracy |           |        | $\frac{\sum_{i<j}(\rho_{ij} - \hat{\rho}_{ij,\text{biased}})^2}{ E }$ |      | $\frac{\sum_{i<j}(\rho_{ij} - \hat{\rho}_{ij,\text{debiased}})^2}{ E }$ |      |
|-------------|---------------------|-----------|--------|-----------------------------------------------------------------------|------|-------------------------------------------------------------------------|------|
|             | MCC                 | Precision | Recall | TP                                                                    | FP   | TP                                                                      | FP   |
| Hub Network | 0.55                | 0.68      | 0.45   | 0.06                                                                  | 1e-4 | 0.01                                                                    | 0.01 |
| Erdos-Renyi | 0.78                | 0.81      | 0.76   | 0.01                                                                  | 1e-4 | 1e-3                                                                    | 0.01 |

Table 4.3: Accuracy and average error metrics for biased and debiased partial correlation estimates of ACCORD.  $|E|$  denotes the number of estimated edges for corresponding TP or FP locations.

#### 4.6.6 Scalability of HP-ACCORD

In this section, we assess the scalability of HP-ACCORD, an implementation of the ACCORD-FBS algorithm for distributed-memory high-performance computing systems.

We generated datasets by simulating samples from a zero-mean  $p$ -variate Gaussian distribution, fixing the sample size at 100, while varying the number of variables from ten thousand to one million. Managing these extensive datasets posed challenges as the conventional method of sampling involves computationally intensive Cholesky decomposition. To tackle this issue, we adopted the approach proposed by [51] where a sparse Cholesky factor  $L$  is generated and directly used for sampling data.

The experiments were conducted on the Nurion supercomputer at the Korea Institute of Science and Technology Information (KISTI) National Supercomputing Center (KSC). The benchmarking involved a comparison with BigQUIC [52], highlighting HP-ACCORD’s advantage in utilizing multiple nodes compared to BigQUIC’s single-node limitation.

Our scalability study results, illustrated in Figure 4.6, include two panels, (a) and (b), representing different levels of regularization used for estimation. In (a), we selected  $\lambda$  to produce a diagonal estimate for both methods, whereas in (b), we adjusted  $\lambda$  to ensure the sparsity of the final estimates closely resembled the ground truth precision matrix, allowing a fair comparison. The horizontal red line in each figure indicates the 48-hour time limit as per KSC’s policy for a single job.

Firstly, HP-ACCORD consistently outperformed BigQUIC in all scenarios. BigQUIC faced significant challenges, particularly failing to estimate when  $p = 10^6$  in scenario (a) and for all cases where  $p \geq 10^5$  in scenario (b). We conjecture that BigQUIC majorly suffered in graph clustering due to the relatively large maximum degree of the graphs in these cases. Additionally, the results for the R implementation of FASTCLIME [53] were excluded from the figure, as it could not complete estimations for  $p \geq 30,000$ , taking approximately 30 hours for the smallest precision matrix with  $p = 10,000$ .

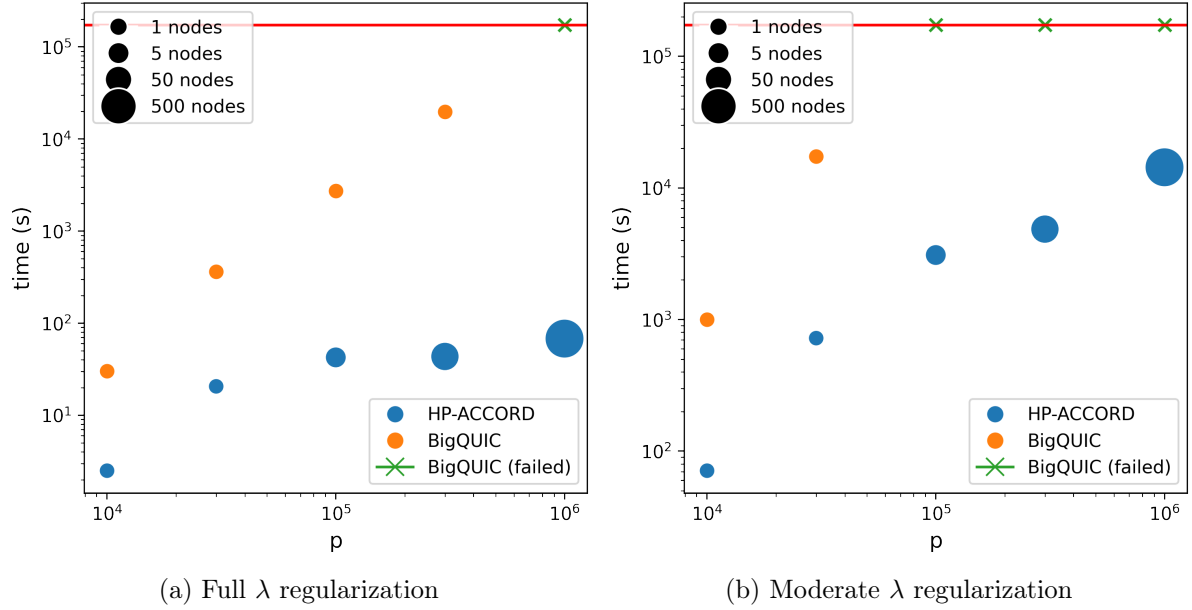


Figure 4.6: Scalability comparison between HP-ACCORD and BigQUIC.

## 4.7 Cast Study: Integrated Network Analysis of Multi-Omics Data in Liver Cancer

In this case study, we estimate a partial correlation network from a multi-omic dataset in Liver Hepatocellular Carcinoma cohort of The Cancer Genome Atlas (TCGA). The dataset consists of 365 samples ( $n = 365$ ) with mRNA-level expression values of 15,598 protein-coding genes (mRNAs), expression levels of 364 microRNAs (miRNAs), and DNA methylation levels ( $\beta$ -values) at 269,396 CpG islands positioned in the upstream regulatory regions of genes, resulting in a total of 285,358 features ( $p = 285,358$ ). These variables are a subset of the processed data available on the TCGA data portal [54], which includes mRNAs, miRNAs, and DNA methylation probes as described in [55]. We apply HP-ACCORD on this dataset to investigate the partial correlation network within and between omics platforms, referred to as intra-modality and inter-modality correlations, respectively.

A primary focus of this analysis is to differentiate the contribution of the transcription factor-driven co-regulations of target genes from that of the gene expression co-repression by concurrent DNA methylations in the promoter regions. Accurate identification of the co-expression network in mRNA expression data is crucial for understanding the key transcription factors, accounting for the presence of other endogenous determinants such as the DNA methylation status in the upstream regulatory regions and gene body. To the best of knowledge, the effort to deconvolute the effects of two or more types of gene expression regulation using graphical models has not been computationally feasible in clinical datasets at this scale.

Following this analysis, we extend our investigation to estimate another network from an expanded dataset, which now includes read counts for 142,417 exons and 72,078 splice junctions as additional features. Furthermore, we include DNA methylation probes that are distantly placed from regulatory regions and gene bodies, resulting in an increase in the dimension of the methylation data from 269,396 to 369,193. Thus, the number of features in the new dataset is 599,648, while the sample size remains unchanged. In this context, the main objective is to identify the conditional dependency structure between exons and splice junctions within each gene and between the sequence units of neighboring genes. This provides a high-resolution correlation map of mRNA transcript units in individual genes and neighbors.

### 4.7.1 Network Estimation in Epigenomic and Transcriptomic Data

In this initial analysis, a debiased partial correlation network is estimated from the first dataset ( $p = 285,358$ ) for which an optimal  $\ell_1$ -regularization parameter  $\lambda$  is chosen to be 0.45 based on epBIC.

As DNA methylation probes outnumber both mRNA- and miRNA-level expression

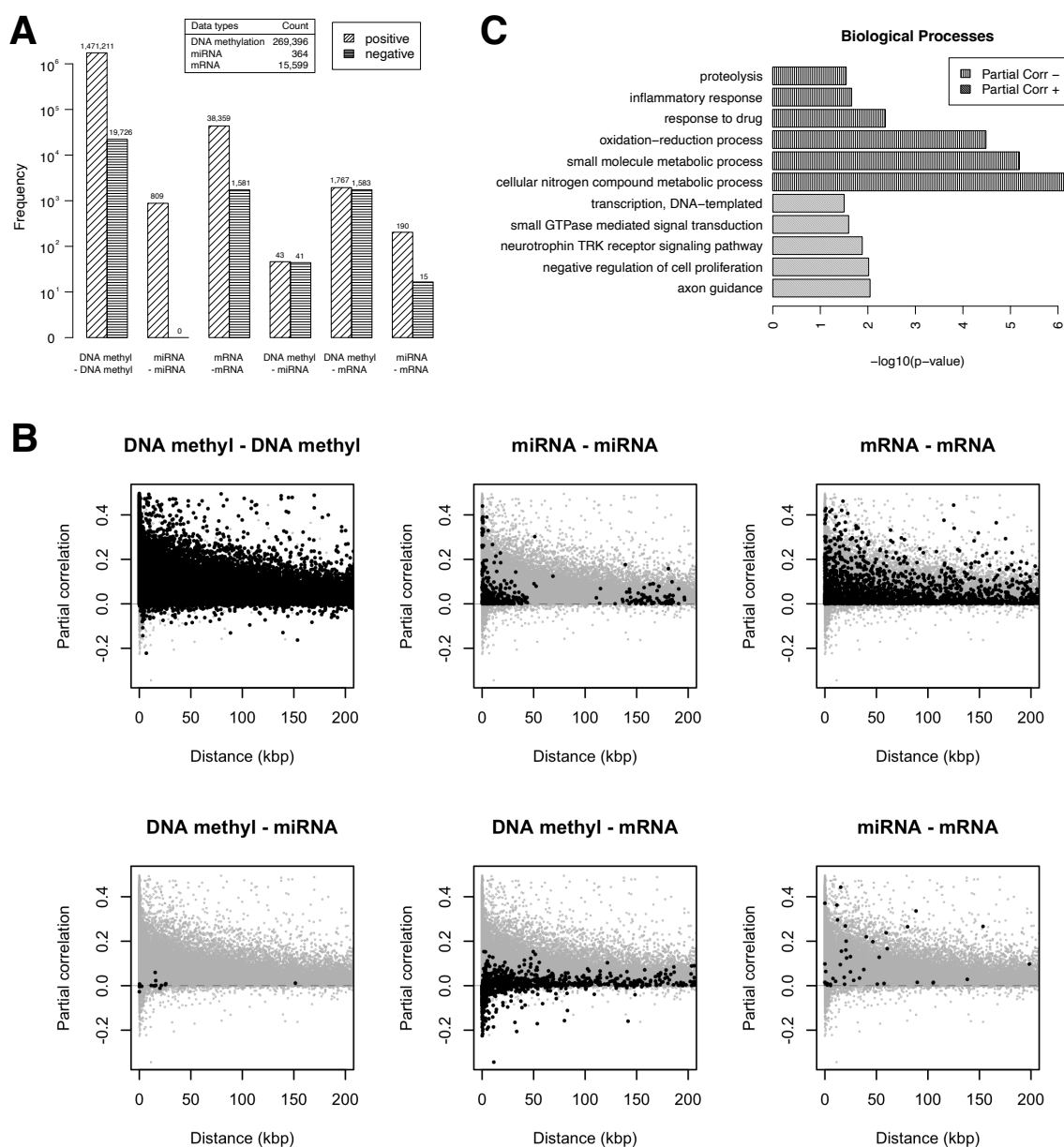


Figure 4.7: Summary of within-modality and between-modality correlations. (A) The barplots show total numbers of edges with positive and negative partial correlations in six categories. (B) Genomic distances between nodes on the same chromosomes. Gray dots are all selected edges and black dots are the edges belonging to each respective category. (C) Gene ontology terms (biological processes) enriched in the nodes (genes) with positive and negative methylation-mRNA partial correlations.

variables in the dataset, the estimated edges were largely represented by nonzero intra-modality correlations among the methylation probes ( $>99\%$ ), in which partial correlations were predominantly positive ( $98\%$ ). The next most frequent edges were from mRNA-mRNA and miRNA-miRNA pairs, where the vast majority showed positive partial correlations ( $96\%$ ), and the magnitude of negative partial correlations was close to zero. This clearly suggests that physically proximal protein coding genes and non-coding RNAs are often co-transcribed. In Figure 4.7A, the frequency of estimated edges (nonzero partial correlations) for intra-modality and inter-modality pairs is shown, categorized by their sign.

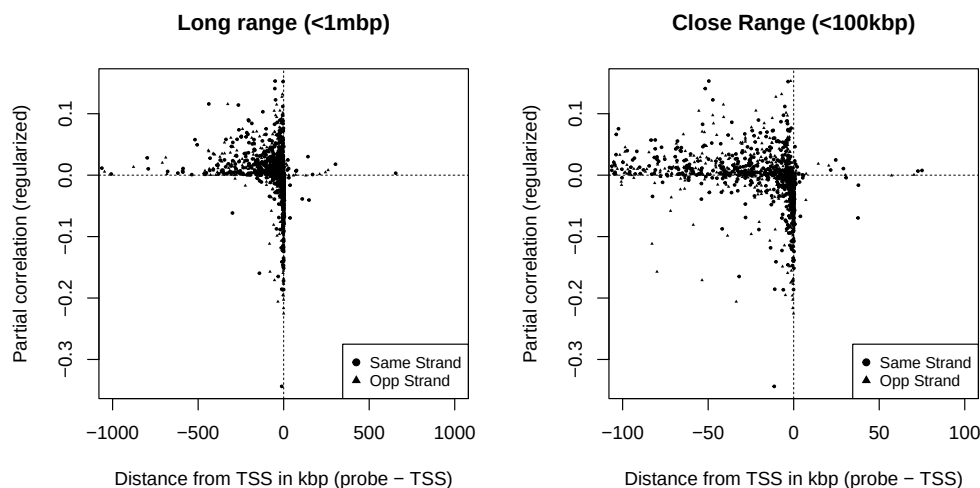


Figure 4.8: Estimated partial correlations between DNA methylation probes and mRNAs for those node pairs on the same chromosomes. The left panel shows the correlations in genomic positions of methylation probes relative to the transcription start site (TSS) of the genes of mRNAs (up to 1 million base distance). The right panel shows the same diagram within 100 kilobase distance.

Additionally, we examine the genomic distances between connected nodes across the six categories. The majority of the positive inter-modality correlations were from the features located on the same chromosome and within 200 kilobase distance from one another 4.7B. The edges connecting mRNAs and miRNAs showed exclusively positive

partial correlations within a 100 kilobase distance only, providing evidence of local specificity in co-transcription events. Last but not least, the selected edges connecting mRNAs and DNA methylation probes showed a mixture of positive and negative correlations. A significant proportion of correlations between mRNAs and DNA methylation probes within a 1 kbp distance from TSS tended to be negative (Figure 4.8), consistent with the hypothesized role of methylation for gene expression repression. When we further investigated the biological functions enriched in the genes showing negatively correlated DNA methylation and mRNA expression, they are mostly the enzymes involved in small molecule metabolism and redox reaction regulation (Figure 4.7C). This suggests that epigenomic regulation actively co-represses the expression levels of metabolic enzymes in liver tumors.

### 4.7.2 Resolving Epigenomic Influence From Co-expression Network

We evaluate the impact of DNA methylation on the estimated networks of mRNA variables by estimating a graph of mRNA variables using only the mRNA expression data and comparing the resulting mRNA-mRNA network with the one estimated with DNA methylation data in the input data.

From this experiment, the estimated mRNA-mRNA network contained 104,599 edges at  $\lambda = 0.30$  based on the epBIC. We observed that the graph estimated without the DNA methylation data, called unadjusted network hereafter, was largely a superset of the graph estimated with the DNA methylation data, called adjusted network (Figure 4.9A). In those edges appearing in both networks, the magnitude of the partial correlation estimates between mRNA pairs was greater in the unadjusted network than in the adjusted network as expected (Figure 4.9B).

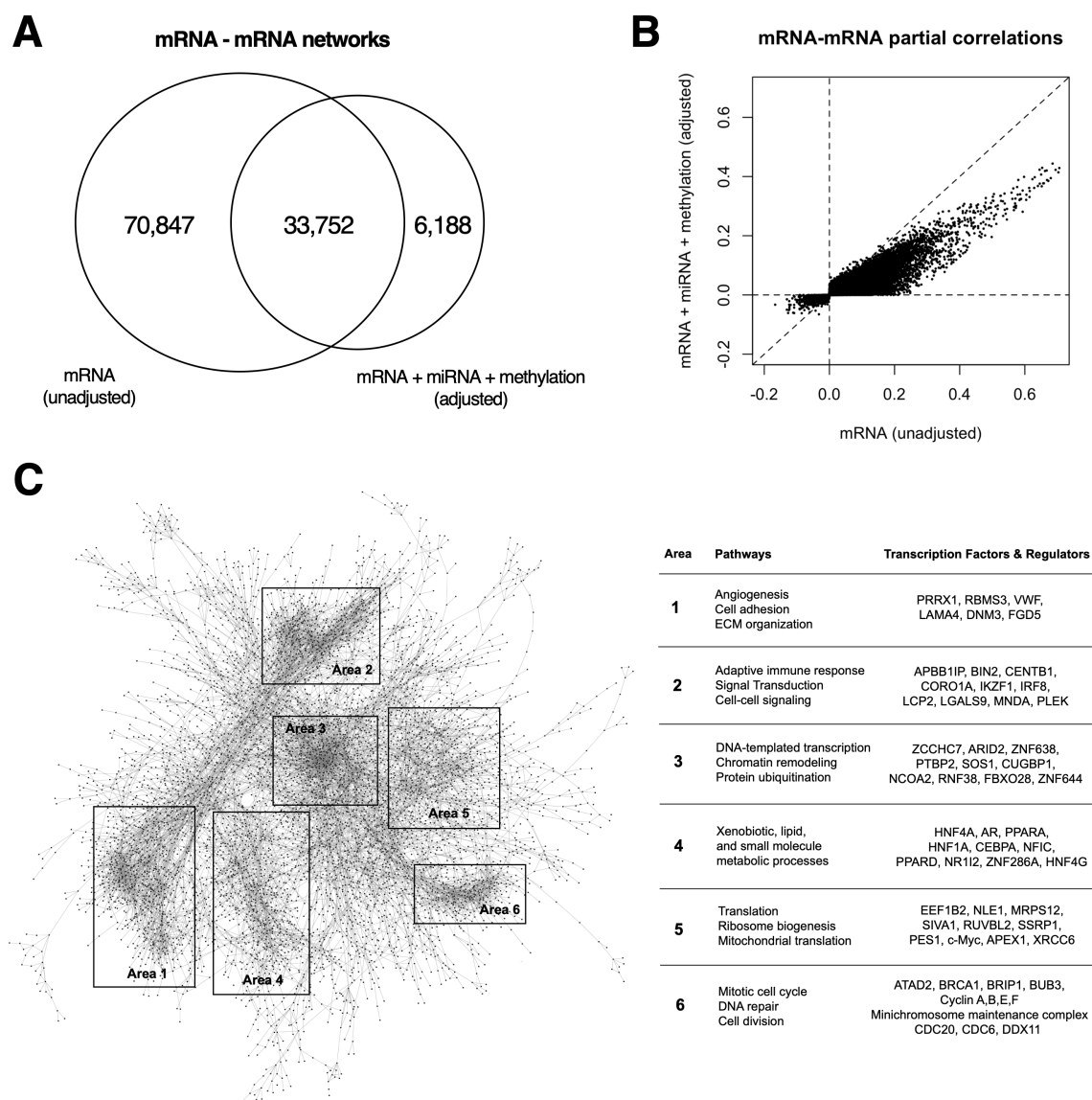


Figure 4.9: (A) Comparison of the two partial correlation networks of mRNA expression data estimated with and without miRNA and DNA methylation data. (B) Comparison of the penalized partial correlation estimates for 110,787 edges appearing in either of the two mRNA-mRNA networks. (C) The network estimated with DNA methylation data. The nodes shown in the graph are mRNAs with at least one edge having a partial correlation above 0.2 or below -0.2 only.

The results above clearly shows that a significant portion of the conditional dependencies between two mRNA nodes could be explained away by the DNA methylation

data, which highlights the contribution of epigenomic elements in the gene expression co-regulation in tumor samples.

Since the impact of methylation-driven co-repression was accounted for, the adjusted network data allows us to infer the active co-regulation driven by transcription factors and co-activators without potential confounding by epigenomic regulation. We thus visualized the adjusted network using the Cytoscape software [56] and selected six subnetworks of high connectivity (Figure 4.9C). We performed hypergeometric probability-based tests for the enrichment of biological functions and transcription factors potentially implicated in the subnetworks. Indeed, the six subnetworks showed highly specific enrichment of widely-known cancer associated biological processes, ranging from the extracellular matrix remodeling in the parenchymal environment, immune response and signal transduction, cell cycle and protein translation, and small molecular metabolism with previously well characterized transcription factors in various cellular processes.

### 4.7.3 Fine-Resolution Network Analysis of Exons and Splice Junctions

In this final experiment, we estimate a graph from the expanded dataset ( $p = 599,648$ ) carrying exons and splice junction-level expressions at  $\lambda = 0.8$  based on epBIC. The estimated graph consisted of 340,611 nodes and 6,894,841 edges. Prior to the analysis, it was expected that the total mRNA levels of a gene, its exon and splice junction values will show the highest degree of partial correlations across the patients, and that the correlations between exons/splice junctions and DNA methylation probes will be smaller. Indeed, all variables physically located within the same gene showed greater correlations than those located in different genes (Figure 4.10A). For this reason, exon-exon, exon-splice junction, and splice junction-splice junction correlations accounted for

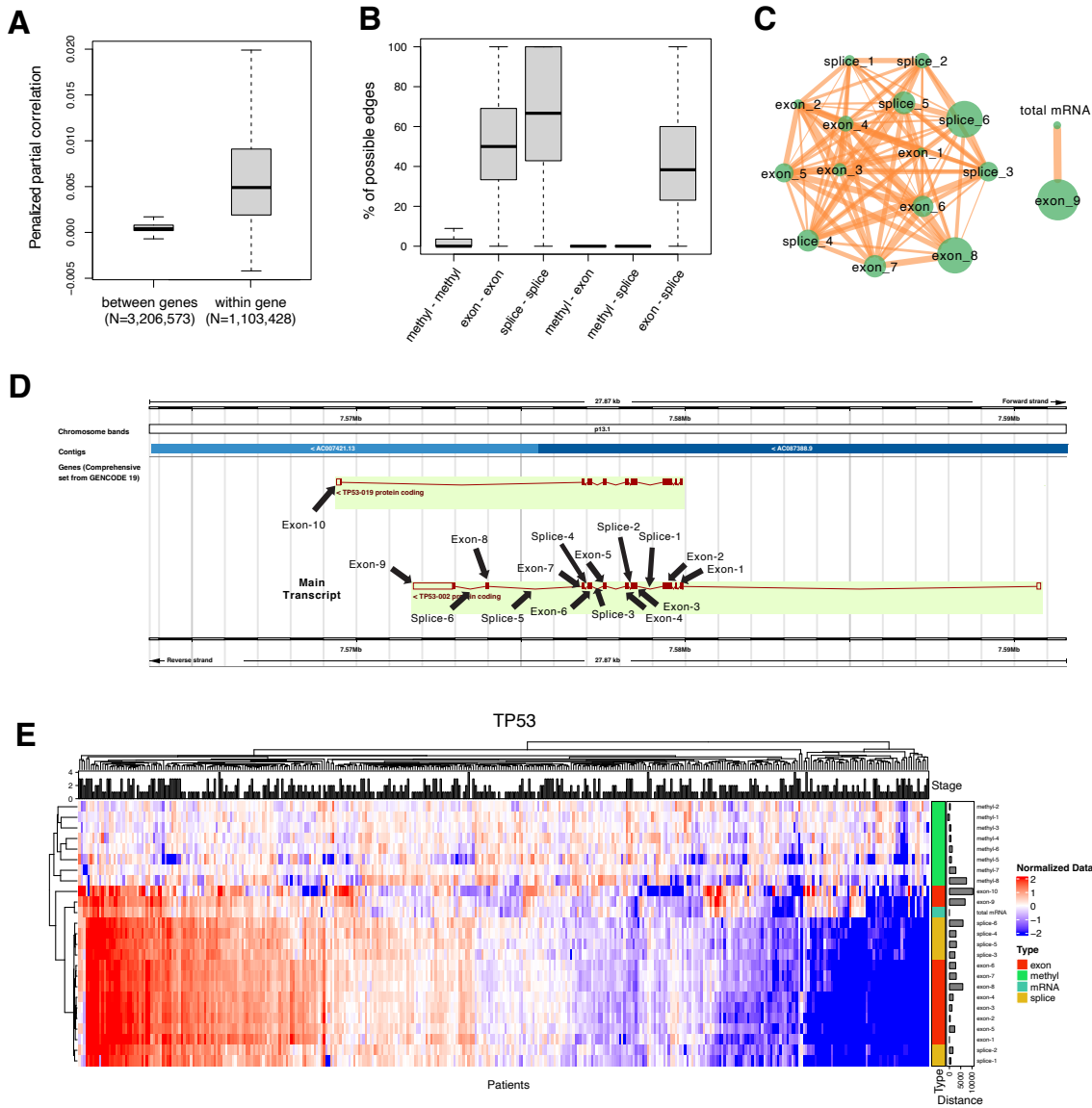


Figure 4.10: (A) The distribution of partial correlation estimates from HP-ACCORD for exon, splice junctions and DNA methylation probes within the same gene and between genes. (B) The number of selected edges between node pairs of indicated types normalized by the number of all possible number of edges between them. (C) Estimated graph for an example gene (TP53) with consistently high partial correlations among the exons and the splice junctions. The node size is proportional to the distance from TSS. (D) TP53 gene structure on chromosome 17 displayed on the Ensembl genome browser (genome version GRCh37). Two protein coding transcripts from GENCODE are shown with respective splice assemblies. Others with extra exonic units not covered by the data were omitted. (E) Mean-centred log2-scale transcripts-per-million (TPM) values across 365 patients for the variables related to TP53.

the majority of the selected edges (Figure 4.10B).

Given that large partial correlations were observed mostly from intra-genic pairs, we investigated the connectivity patterns in well-known cancer associated genes. In many cases, the whole gene-level mRNA expression node was more connected to the exons and splice junctions closer to the 3' end of the genes, which may reflect the influence of poly(A) enrichment during the RNA-seq library preparation stage in this dataset. Even after accounting for the potential experimental bias, however, exons and junctions physically closer to each other showed greater positive partial correlations than distant ones, and this pattern was more frequently observed in the genes whose exon positions span a long range on the chromosome. Overall, the observation suggests that there may exist varying degrees of incomplete transcripts or alternative splicing events occurring in liver tumor tissues.

For example, Figure 4.10C shows that the estimated network of the variables associated with the TP53 gene, the most mutated gene across all cancer types, which is located on the reverse strand of chromosome 17. In this graph, the node size is proportional to the genomic distance to the 3' end untranslated regions, i.e., the end of the gene, and the edge thickness reflects the magnitude of partial correlations. The graph shows that the connectivity is tighter among the early exons (1-5) and begins to weaken as the genomic location reaches the middle of the gene (exons 6-7) and reaches the last exon (9) of the main transcript containing the untranslated region (UTR) (hollow box in Figure 4.10D). Splice junction (6) was connected to junction 5 and exon 8 but not exon 9. Exon 10 was not connected to any other variables within the gene, reflecting that the transcript containing the exon is a minor transcript not correlated with total gene expression levels across the tumors (top, Figure 4.10D). In the Ensembl database [57], there were a few other transcript assemblies with exons containing only up to exon 6 (not shown), and the data above shows that those are likely minor transcripts not associated with the total

gene expression level, either.

From the gene structure diagram in Figure 4.10D and the normalized data in Figure 4.10E, we see that TP53 is strongly associated with the longest isoform spanning from exon 1 through exon 9, but it is plausible that there exist multiple transcript assemblies based on the composition of three exonic groups (1-5, 6-8, and 9). The presence of specific assemblies and their translation potential may need to be verified in the sequencing data of individual patients ultimately. However, our graph estimation approach is clearly indicative of major differential splicing patterns and provides clear data evidence for prioritizing genes for further elucidation of exon usage patterns and transcript diversity in proliferating tumors.

## 4.8 Discussion

In this chapter, we introduced the ACCORD algorithm for sparse precision matrix estimation based on pseudo-likelihood. For optimization, we proposed a novel objective splitting for a proximal gradient method that exhibits guaranteed descent property for a wide range of step size and linear convergence in both the objective values and the iterates. Additionally, we demonstrated that the convergence of the algorithm can be significantly improved by incorporating backtracking line search using a reciprocal of global Lipschitz constant as a lower bound, which in turn enhances numerical stability. For ultra-high-dimensional data, specifically multi-omics data analyses in mind, we developed a high-performance variant of ACCORD, named HP-ACCORD, that can scale up to over 1 million variables utilizing parallelization and communication-avoiding linear algebra algorithms.

In discovery-driven fields such as clinical multi-omics, technological advances in molecular and imaging modalities have steadily driven the number of variables to analyze

upward. The typical dimensionality of datasets easily exceeds tens or hundreds of thousands, as seen in the motivating TCGA dataset produced a decade ago. In analyzing and making inferences from these truly big data, the computational feasibility of the methodology outweighs the likelihood of a slight loss in statistical efficiency. Our proposed method, ACCORD, takes this avenue. The objective function of ACCORD possesses computationally attractive “quadratic plus logarithms” structure inherited from its predecessor, CONCORD. This simple structure with the carefully designed optimization algorithm admits massively large-scale computation of  $\ell_1$ -regularized graphical model selection suitable for the aforementioned applications.

# Appendix A

## Appendix for FROSTY

### A.1 Proofs

#### A.1.1 Preliminaries

Recall that our loss function parametrized with  $L \in \mathbb{L}_p$  is

$$\ell(S; L) = \text{trace}(SLL^T) - 2 \log |L|$$

and its gradient is

$$\frac{\partial}{\partial L} \ell(S; L) = 2SL - 2(L^{-1})^T.$$

Let  $\mathcal{P}(\mathbb{S}_p)$  be the set of all probability distributions supported on  $\mathbb{S}_p$  and  $S_1, \dots, S_n$  be an independent and identically distributed random sample from  $S$  coming from a distribution on  $\mathcal{P}(\mathbb{S}_p)$ . Now, we restate the following two propositions, suitable to our context, that are the key ingredients for proofs of the theorems.

**Proposition 1** (Proposition C.2 of [19]). *For  $\gamma \geq 0$  and loss functions  $\ell(S'; L)$  that are*

upper semi-continuous in  $S' \in \mathbb{S}_p$  for each  $L \in \mathbb{L}_p$ , let

$$\phi_\gamma(S_i; L) = \sup_{S \in \mathbb{S}_p} \{\ell(S; L) - \gamma c(S, S_i)\}. \quad (\text{A.1})$$

Then,

$$\sup_{\mathcal{P}: D_c(\mathcal{P}, \mathcal{P}_n) \leq \lambda} E_{\mathcal{P}}[\ell(S; L)] = \min_{\gamma \geq 0} \left\{ \gamma \lambda + \frac{1}{n} \sum_{i=1}^n \phi_\gamma(S_i; L) \right\}. \quad (\text{A.2})$$

**Proposition 2** (Proposition C.1 of [19]). *Consider  $L \in \mathbb{L}_p$ . Let  $h(\cdot, L)$  be Borel measurable. Also, suppose that  $\mathbb{O}_{p \times p}$  lies in the interior of the convex hull of  $\{h(S', C) : S' \in \mathbb{S}_p\}$ .*

Then,

$$R_n(L) = \sup_{\Lambda \in \mathbb{S}_p} \left\{ -\frac{1}{n} \sum_{i=1}^n \sup_{S \in \mathbb{S}_p} \{\text{trace}(\Lambda^T h(S; L)) - c(S, S_i)\} \right\}. \quad (\text{A.3})$$

### A.1.2 Proof of Theorem 3

*Proof:* Using the duality from Proposition 1, we can rewrite the left-hand side of the equation (4) from the theorem

$$\begin{aligned} \min_{L \in \mathbb{L}_p} \sup_{\mathcal{P}: D_c(\mathcal{P}, \mathcal{P}_n) \leq \lambda} E_{\mathcal{P}}[\ell(S; L)] &= \min_{L \in \mathbb{L}_p} \left[ \min_{\gamma \geq 0} \left\{ \gamma \lambda + \frac{1}{n} \sum_{i=1}^n \phi_\gamma(S_i; L) \right\} \right] \\ &= \min_{L \in \mathbb{L}_p} \left[ \min_{\gamma \geq 0} \left\{ \gamma \lambda + \frac{1}{n} \sum_{i=1}^n \left( \sup_{S \in \mathbb{S}_p} \{\ell(S; L) - \gamma c(S, S_i)\} \right) \right\} \right]. \end{aligned} \quad (\text{A.4})$$

Let  $\Delta = S - S_i$ . Then,

$$\begin{aligned}
& \sup_{S \in \mathbb{S}_p} \{\ell(S; L) - \gamma c(S, S_i)\} \tag{A.5} \\
&= \sup_{S \in \mathbb{S}_p} \{\text{trace}(SLL^T) - 2 \log |L| - \gamma \|\text{vec}(S) - \text{vec}(S_i)\|_\infty\} \\
&= \sup_{\Delta \in \mathbb{S}_p} \{\text{trace}((\Delta + S_i)LL^T) - \gamma \|\text{vec}(\Delta)\|_\infty\} - 2 \log |L| \\
&= \sup_{\Delta \in \mathbb{S}_p} \{\text{trace}(\Delta LL^T) - \gamma \|\text{vec}(\Delta)\|_\infty\} + \text{trace}(S_i LL^T) - 2 \log |L| \\
&= \sup_{\Delta \in \mathcal{M}(L)} \{\|\text{vec}(\Delta)\|_\infty \|\text{vec}(LL^T)\|_1 - \gamma \|\text{vec}(\Delta)\|_\infty\} + \text{trace}(S_i LL^T) - 2 \log |L| \\
&= \sup_{\Delta \in \mathcal{M}(L)} \{\|\text{vec}(\Delta)\|_\infty (\|\text{vec}(LL^T)\|_1 - \gamma)\} + \text{trace}(S_i LL^T) - 2 \log |L| \tag{A.6}
\end{aligned}$$

where  $\mathcal{M}(L) = \{\Delta \in \mathbb{S}_p : \text{trace}(\Delta LL^T) > 0, |(\Delta)_{ij}|^q = |(LL^T)_{ij}|^p\}$  with  $\frac{1}{p} + \frac{1}{q} = 1$ , which is the if and only if condition for Holder's inequality to hold tightly [58]. Since this condition also holds for the limiting case, we choose  $p = 1$  and  $q = \infty$ .

Notice that if  $\gamma < \|\text{vec}(LL^T)\|_1$ , the supremum term in the last line of (A.6) can be arbitrarily large. Since we have the minimum over  $\gamma \geq 0$  in (A.4), we must take  $\gamma \geq \|\text{vec}(LL^T)\|_1$ , which leads to the supremum term to be zero. Then, we have

$$\begin{aligned}
\min_{L \in \mathbb{L}_p} \sup_{\mathcal{P}: D_c(\mathcal{P}, \mathcal{P}_n) \leq \lambda} E_{\mathcal{P}}[\ell(S; L)] &= \min_{L \in \mathbb{L}_p} \left[ \min_{\gamma \geq \|\text{vec}(LL^T)\|_1} \left\{ \gamma \lambda + \frac{1}{n} \sum_{i=1}^n (\text{trace}(S_i LL^T) - 2 \log |L|) \right\} \right] \\
&= \min_{L \in \mathbb{L}_p} \{\text{trace}(SLL^T) - 2 \log |L| + \lambda \|\text{vec}(LL^T)\|_1\}.
\end{aligned}$$

### A.1.3 Proof of Theorem 4

*Proof:* Consider the RWP function with  $h(S; L) = 2SL - 2(L^{-1})^T$ . Applying Proposition 2, we have

$$\begin{aligned} R_n(L) &= \sup_{\Lambda \in \mathbb{S}_p} \left\{ -\frac{1}{n} \sum_{i=1}^n \sup_{S \in \mathbb{S}_p} \{ \text{trace}(\Lambda^T h(S; L)) - c(S, S_i) \} \right\} \\ &= \sup_{\Lambda \in \mathbb{S}_p} \left\{ -\frac{1}{n} \sum_{i=1}^n \sup_{S \in \mathbb{S}_p} \{ \text{trace}(2\Lambda^T (SL - (L^{-1})^T)) - \|\text{vec}(S) - \text{vec}(S_i)\|_\infty \} \right\}. \end{aligned}$$

Let  $\Delta = S - S_i$ . Then, following the procedure of (A.6) in the proof of Theorem 3, we obtain

$$\begin{aligned} & \sup_{S \in \mathbb{S}_p} \{ \text{trace}(2\Lambda^T (SL - (L^{-1})^T)) - \|\text{vec}(S) - \text{vec}(S_i)\|_\infty \} \\ &= \sup_{\Delta \in \mathbb{S}_p} \{ \text{trace}(2\Lambda^T ((\Delta + S_i)L - (L^{-1})^T)) - \|\text{vec}(\Delta)\|_\infty \} \\ &= \sup_{\Delta \in \mathbb{S}_p} \{ \text{trace}(2\Lambda^T \Delta L) - \|\text{vec}(\Delta)\|_\infty \} + \text{trace}(2\Lambda^T (S_i L - (L^{-1})^T)) \\ &= \sup_{\Delta \in \mathcal{M}(\Lambda)} \{ \|\text{vec}(\Delta)\|_\infty \|\text{vec}(2L\Lambda^T)\|_1 - \|\text{vec}(\Delta)\|_\infty \} + \text{trace}(2\Lambda^T (S_i L - (L^{-1})^T)) \\ &= \sup_{\Delta \in \mathcal{M}(\Lambda)} \{ \|\text{vec}(\Delta)\|_\infty (\|\text{vec}(2L\Lambda^T)\|_1 - 1) \} + \text{trace}(2\Lambda^T (S_i L - (L^{-1})^T)). \quad (\text{A.7}) \end{aligned}$$

Similarly, we must consider  $\|\text{vec}(2L\Lambda^T)\|_1 \leq 1$  to prevent the supremum term in the last line of (A.7) from being taken arbitrarily large. Then, we have

$$\begin{aligned}
R_n(L) &= \sup_{\Lambda \in \mathbb{S}_p: \|\text{vec}(2L\Lambda^T)\|_1 \leq 1} \left\{ -\frac{1}{n} \sum_{i=1}^n \text{trace}(2\Lambda^T(S_i L - (L^{-1})^T)) \right\} \\
&= \sup_{\Lambda \in \mathbb{S}_p: \|\text{vec}(2L\Lambda^T)\|_1 \leq 1} \left\{ -\text{trace}(2\Lambda^T(SL - (L^{-1})^T)) \right\} \\
&= \sup_{\Lambda \in \mathbb{S}_p: \|\text{vec}(2L\Lambda^T)\|_1 \leq 1} \left\{ -\text{trace}((SLL^{-1} - (L^{-1})^T L^{-1})2L\Lambda^T) \right\} \\
&= \sup_{\Lambda \in \mathbb{S}_p: \|\text{vec}(2L\Lambda^T)\|_1 \leq 1} \left\{ -\text{trace}((S - (LL^T)^{-1})2L\Lambda^T) \right\} \\
&= \sup_{\Lambda \in \mathbb{S}_p: \|\text{vec}(2L\Lambda^T)\|_1 \leq 1} \left\{ \text{vec}(2L\Lambda^T)^T \text{vec}(S - (LL^T)^{-1}) \right\} \\
&= \|\text{vec}(S - (LL^T)^{-1})\|_\infty.
\end{aligned}$$

Finally, since

$$\{L \in \mathcal{C}_n(\lambda)\} = \{\mathcal{O}(L) \cap \{\mathcal{P} : D_c(\mathcal{P}, \mathcal{P}_n) \leq \lambda\} \neq \emptyset\} = \{R_n(L) \leq \lambda\},$$

one can, given a  $(1 - \alpha)$ -confidence level, optimally choose the ambiguity size  $\lambda$  by

$$\begin{aligned}
\lambda_\alpha &= \inf \{\lambda > 0 : \mathbb{P}_0(L \in \mathcal{C}_n(\lambda)) \geq 1 - \alpha\} \\
&= \inf \{\lambda > 0 : \mathbb{P}_0(R_n(L) \leq \lambda) \geq 1 - \alpha\} \\
&= \inf \{\lambda > 0 : \mathbb{P}_0(\|\text{vec}(S - (LL^T)^{-1})\|_\infty \leq \lambda) \geq 1 - \alpha\}.
\end{aligned}$$

### A.1.4 Proof of Lemma 3.3

*Proof:*

$$\begin{aligned}
R_n(\Theta) &= \|\text{vec}(S - \Theta^{-1})\|_\infty \\
&= \|\text{vec}(S - (P_\pi^T L_\pi L_\pi^T P_\pi)^{-1})\|_\infty \\
&= \|\text{vec}(S - P_\pi^T (L_\pi L_\pi^T)^{-1} P_\pi)\|_\infty \\
&= \|\text{vec}(P_\pi^T [P_\pi S P_\pi^T - (L_\pi L_\pi^T)^{-1}] P_\pi)\|_\infty \\
&= \|\text{vec}(P_\pi^T [S_\pi - (L_\pi L_\pi^T)^{-1}] P_\pi)\|_\infty \\
&= \|\text{vec}(S_\pi - (L_\pi L_\pi^T)^{-1})\|_\infty = R_n(L_\pi).
\end{aligned}$$

## A.2 Additional Numerical Results

In this section, we present numerical experiments on a variety of real networks to evaluate further investigate the performance of our proposed method, FROSTY. We introduce additional accuracy metrics, namely, True Positives (TP), False Positives (FP), Reversed edges (R), and Missing edges (M). TP represents the number of correctly estimated edges, taking into account both the presence and orientation of the edges. FP corresponds to the number of estimated edges that are not present in the skeleton of the true graph. R represents the number of estimated edges that have an inconsistent orientation compared to the corresponding true edges. Lastly, M denotes the number of missing edges in the skeleton of the true graph that were not identified by our method. In addition, we introduce two novel accuracy metrics: Edge Detection Rate (EDR) and Edge Orientation Rate (EOR). These metrics can be conveniently derived from the previously defined accuracy metrics, i.e.,  $\text{EDR} = (\text{TP} + \text{R}) / \text{T}$  and  $\text{EOR} = \text{TP} / (\text{TP} + \text{R})$  where T is the total number of edges in the true graph.

During our experiments, we observed a tendency of the SHD metric to favor sparse estimates, often to an extent that seems unreasonable. As an illustration, we consider the insurance network, where the RFD algorithm predominantly produced empty graph estimates; nonetheless, it achieved the lowest SHD score. To demonstrate this phenomenon, Figure A.1 depicts a comparison of the CPDAG for the insurance network alongside two estimated CPDAGs obtained from the FROSTY and RFD algorithms, respectively.

| Network                                | Method  | Accuracy Metrics |       |       |        |               |             |
|----------------------------------------|---------|------------------|-------|-------|--------|---------------|-------------|
|                                        |         | TP               | FP    | R     | M      | SHD (std)     | MCC (std)   |
| Sachs<br>$p = 11$<br>$ E  = 20$        | FROSTY  | 6.10             | 3.65  | 1.30  | 12.60  | 17.55 (2.42)  | 0.30 (0.11) |
|                                        | RFD     | 0.00             | 0.00  | 0.00  | 20.00  | 20.00 (0.00)  | 0.00 (0.00) |
|                                        | NOTEARS | 3.30             | 2.10  | 2.10  | 14.60  | 18.80 (3.24)  | 0.26 (0.12) |
|                                        | ARCS    | 1.85             | 2.25  | 3.75  | 14.40  | 20.40 (3.41)  | 0.22 (0.12) |
| Insurance<br>$p = 27$<br>$ E  = 52$    | FROSTY  | 4.75             | 11.60 | 11.70 | 35.55  | 58.85 (5.46)  | 0.28 (0.05) |
|                                        | RFD     | 0.50             | 0.65  | 1.40  | 50.10  | 52.15 (0.75)  | 0.04 (0.10) |
|                                        | NOTEARS | 2.30             | 3.20  | 5.95  | 43.75  | 52.90 (1.59)  | 0.21 (0.06) |
|                                        | ARCS    | 5.15             | 6.95  | 8.95  | 37.90  | 53.80 (2.67)  | 0.24 (0.04) |
| Alarm<br>$p = 37$<br>$ E  = 37$        | FROSTY  | 3.75             | 18.75 | 23.70 | 18.55  | 61.00 (6.99)  | 0.43 (0.03) |
|                                        | RFD     | 2.50             | 6.55  | 12.05 | 31.45  | 50.05 (7.97)  | 0.35 (0.11) |
|                                        | NOTEARS | 4.85             | 6.40  | 13.35 | 27.80  | 47.55 (2.48)  | 0.40 (0.04) |
|                                        | ARCS    | 8.50             | 9.95  | 14.35 | 23.15  | 47.45 (3.07)  | 0.38 (0.06) |
| Hailfinder<br>$p = 56$<br>$ E  = 66$   | FROSTY  | 7.25             | 23.95 | 20.50 | 38.25  | 82.70 (6.04)  | 0.36 (0.04) |
|                                        | RFD     | 1.60             | 6.30  | 5.95  | 58.45  | 70.70 (4.80)  | 0.13 (0.14) |
|                                        | NOTEARS | 4.50             | 10.75 | 10.70 | 50.80  | 72.25 (3.19)  | 0.26 (0.06) |
|                                        | ARCS    | 6.15             | 14.10 | 15.45 | 44.40  | 73.95 (4.63)  | 0.25 (0.06) |
| Pathfinder<br>$p = 109$<br>$ E  = 195$ | FROSTY  | 32.55            | 16.55 | 27.05 | 135.40 | 179.00 (4.82) | 0.41 (0.02) |
|                                        | RFD     | 24.35            | 8.20  | 14.75 | 155.90 | 178.85 (5.74) | 0.34 (0.10) |
|                                        | NOTEARS | 16.70            | 3.55  | 8.10  | 170.20 | 181.85 (3.15) | 0.30 (0.02) |
|                                        | ARCS    | 16.90            | 5.30  | 20.75 | 157.35 | 183.40 (3.91) | 0.32 (0.02) |

Note: TP and MCC are the larger the better. FP, R, M, and SHD are the smaller the better where  $SHD = FP + R + M$ .

Table A.1: Accuracy metrics for various real networks with sample size of  $n = 10p$ .

| Network                                | Method  | Accuracy Metrics |      |               |             |
|----------------------------------------|---------|------------------|------|---------------|-------------|
|                                        |         | EDR              | EOR  | SHD (std)     | MCC (std)   |
| Sachs<br>$p = 11$<br>$ E  = 20$        | FROSTY  | 0.37             | 0.83 | 17.55 (2.42)  | 0.30 (0.11) |
|                                        | RFD     | 0.00             | NA   | 20.00 (0.00)  | 0.00 (0.00) |
|                                        | NOTEARS | 0.27             | 0.66 | 18.80 (3.24)  | 0.26 (0.12) |
|                                        | ARCS    | 0.28             | 0.39 | 20.40 (3.41)  | 0.22 (0.12) |
| Insurance<br>$p = 27$<br>$ E  = 52$    | FROSTY  | 0.32             | 0.29 | 58.85 (5.46)  | 0.28 (0.05) |
|                                        | RFD     | 0.04             | 0.26 | 52.15 (0.75)  | 0.04 (0.10) |
|                                        | NOTEARS | 0.16             | 0.28 | 52.90 (1.59)  | 0.21 (0.06) |
|                                        | ARCS    | 0.27             | 0.36 | 53.80 (2.67)  | 0.24 (0.04) |
| Alarm<br>$p = 37$<br>$ E  = 37$        | FROSTY  | 0.60             | 0.14 | 61.00 (6.99)  | 0.43 (0.03) |
|                                        | RFD     | 0.32             | 0.21 | 50.05 (7.97)  | 0.35 (0.11) |
|                                        | NOTEARS | 0.40             | 0.27 | 47.55 (2.48)  | 0.40 (0.04) |
|                                        | ARCS    | 0.50             | 0.37 | 47.45 (3.07)  | 0.38 (0.06) |
| Hailfinder<br>$p = 56$<br>$ E  = 66$   | FROSTY  | 0.42             | 0.26 | 82.70 (6.04)  | 0.36 (0.04) |
|                                        | RFD     | 0.11             | 0.18 | 70.70 (4.80)  | 0.13 (0.14) |
|                                        | NOTEARS | 0.23             | 0.28 | 72.25 (3.19)  | 0.26 (0.06) |
|                                        | ARCS    | 0.33             | 0.28 | 73.95 (4.63)  | 0.25 (0.06) |
| Pathfinder<br>$p = 109$<br>$ E  = 195$ | FROSTY  | 0.31             | 0.55 | 179.00 (4.82) | 0.41 (0.02) |
|                                        | RFD     | 0.20             | 0.65 | 179.05 (5.74) | 0.34 (0.10) |
|                                        | NOTEARS | 0.13             | 0.67 | 181.85 (3.15) | 0.30 (0.02) |
|                                        | ARCS    | 0.19             | 0.45 | 183.40 (3.91) | 0.32 (0.02) |

Note: EDR, EOR, and MCC are the larger the better. SHD is the smaller the better where  $\text{EDR} = (\text{TP} + \text{R}) / \text{T}$ ,  $\text{EOR} = \text{TP} / (\text{TP} + \text{R})$ ,  $\text{SHD} = \text{FP} + \text{R} + \text{M}$ .

Table A.2: Accuracy metrics for various real networks with sample size of  $n = 10p$ .

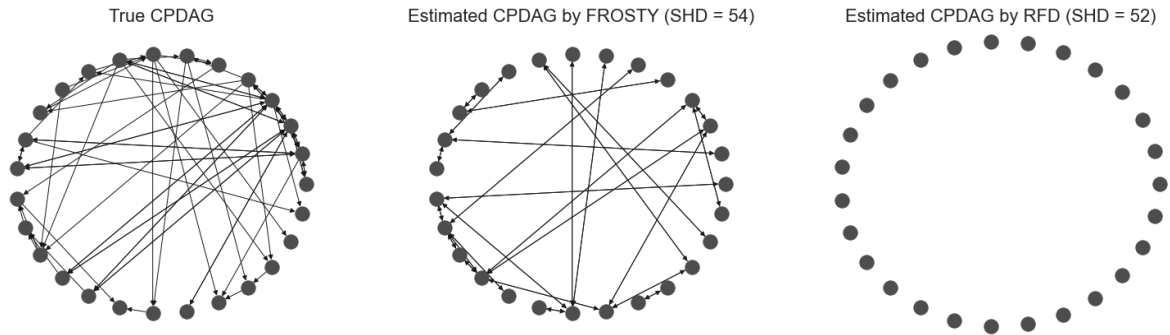


Figure A.1: An instance of the unreasonable behavior of the SHD metric.

### A.3 Software Package

We provide a simple guide for using the FROSTY algorithm, which is available as a software package called *frosty-dag* on PyPI. To utilize *frosty-dag*, it is necessary to first install *scikit-sparse*, which depends on *suite-sparse*. This can be accomplished with the following command lines

```
# for mac
brew install suite-sparse
# for debian
sudo apt-get install libsuitesparse-dev
```

After installing the required dependencies, the FROSTY algorithm can be installed with the following command

```
pip install frosty-dag
```

Given a standardized dataset  $X$ , where rows and columns represent samples and features, respectively, a Bayesian network can be estimated with a single line of code

```
1 from frosty import frosty
```

2

```
3 B, perm = frosty(X, alpha=0.99, b=200, diag=True).
```

Here,  $B$  is the estimated coefficient matrix of the Bayesian network, and  $perm$  represents the recovered variable ordering. As demonstrated in Chapter 2, FROSTY requires minimal parameter tuning. The primary parameter to be tuned is  $alpha$ , corresponding to the  $(1 - \alpha)$ -confidence level. We recommend using a high value such as 0.9 or 0.99, unless an extremely sparse estimate is desired. The remaining parameters are associated with the RobSel algorithm and are relatively insensitive to tuning. Detailed documentation can be found in `frosty.py`.

# Appendix B

## Appendix for Applications of FROSTY

### B.1 Missing Data Generation under MAR

In this section, we describe the details of generating missing data under the MAR assumption. Given a dataset  $X \in \mathbb{R}^{n \times p}$  of  $n$  observations and  $p$  variables, let  $p_{\text{na}}$  denote the number of variables containing missing values, and  $p_{\text{obs}}$  represent the number of variables with no missing values; thus,  $p = p_{\text{na}} + p_{\text{obs}}$ . To generate missing values in  $X$  under the MAR assumption with a desired missing rate  $m$ , we use a logistic function to determine the missing probability for the  $i^{\text{th}}$  observation of each of the pre-selected variables that will contain missing values. For each of these variables, consider the following logistic function

$$\text{logit}_i(\beta_0, \beta) = \frac{1}{1 + \exp(-(\beta_0 + \beta^T x_i))}.$$

Here,  $x_i$  is the  $i^{\text{th}}$  observation of the dataset  $X$ , excluding variables intended to have missing values. The parameters  $\beta_0$  and  $\beta$  correspond to the intercept and a coefficient vector of length  $p_{\text{obs}}$ , respectively. We want to find the coefficients  $\beta_0$  and  $\beta$  such that

$\frac{1}{n} \sum_{i=1}^n \text{logit}_i(\beta_0, \beta) = m$ . If we generate  $\beta$  from a standard normal distribution, then we are confronted with a series of  $p_{\text{na}}$  univariate optimization problems each of which can be easily solved using the bisection method. Once  $\beta_0$  and  $\beta$  are determined, say  $\hat{\beta}_0$  and  $\hat{\beta}$ , we mask values in the pre-selected variables according to the probabilities given by  $\text{logit}_i(\hat{\beta}_0, \hat{\beta})$ .

# Appendix C

## Appendix for ACCORD

### C.1 Proofs

#### C.1.1 Proof of Theorem 5

*Proof:* Convergence of  $\{\Omega^{(t)}\}$  and  $\hat{\Omega}$  is a standard result in the operator splitting literature; see, e.g, [46, 59, 45]. Monotone convergence of  $\ell(\Omega^{(t)})$  to  $\ell^*$  follows from continuity of  $\ell$  and the descent property of the algorithm.

In order to establish the rate of convergence, we first prove a Łojasiewicz error bound [60]. Consider the splitting (4.10). Let  $y = \text{vec}(\Omega) \in \mathbb{R}^{p^2}$  and write

$$\begin{aligned}\bar{f}(\Omega) &= \bar{f}(y) = -\phi((I - P)y) + \frac{1}{2}y^T Q y, \quad \bar{g}(\Omega) = \bar{g}(y) = \lambda \|Py\|_1, \\ \ell(\Omega) &= \ell(y) = \bar{f}(y) + \bar{g}(y)\end{aligned}$$

where  $Q = S \otimes I$ ,  $P$  is a projection matrix such that  $Py = \text{vec}(\Omega - \Omega_D)$ , and  $\phi(\omega) = -\sum_i \log[\text{mat}(\omega)_{ii}]$ . It is clear that the smooth part  $\bar{f}$  is a proper closed convex function with an open domain, is strongly convex on any compact convex subset of  $\text{dom}(\bar{f}) = \{y \in$

$\mathbb{R}^{p^2} : (\text{mat}(y))_{ii} > 0\}$  and is twice continuously differentiable on  $\text{dom}(\bar{f})$ . The nonsmooth part  $\bar{g}$ , with  $\text{dom}(\bar{g}) = \mathbb{R}^{p^2}$ , is piecewise linear, thus its epigraph is polyhedral.

Since the objective function  $\ell$  is strictly convex and coercive, all of its sublevel sets are compact and convex. Let  $y^* = \text{vec}(\hat{\Omega})$ , the unique minimizer of  $\ell$ . Then,  $\ell(y^*) \leq \ell(\text{vec}(I)) = \text{trace}(S)/2$  and  $L_0 = \{y \in \mathbb{R}^{p^2} : \ell(y) \leq \text{trace}(S)/2\}$  is nonempty, compact, and convex. Obviously  $y^* \in L_0$ . Now let

$$R = \max \{\|y\|_1 : y \in L_0\} \in (0, \infty)$$

so that  $\|y^*\|_1 \leq R$  and  $\|y\|_1 \leq R$  whenever  $y \in L_0$ . Then,

$$\begin{aligned} \min_{y \in \mathbb{R}^{p^2}} \ell(y) &= \min \{\bar{f}(y) + \bar{g}(y) : y \in L_0\} \\ &= \min \{\bar{f}(y) + \lambda \|Py\|_1 : \|y\|_1 \leq R\} \\ &= \min \{\bar{f}(y) + \lambda(\mathbf{1}^T z^+ + \mathbf{1}^T z^-) : \|y\|_1 \leq R, z^+ - z^- = Py, z^+ \geq \mathbf{0}, z^- \geq \mathbf{0}\} \\ &= \min \{\bar{f}(y^+ - y^-) + \lambda(\mathbf{1}^T z^+ + \mathbf{1}^T z^-) : \mathbf{1}^T y^+ + \mathbf{1}^T y^- \leq R, y^+ \geq \mathbf{0}, y^- \geq \mathbf{0}, \\ &\quad z^+ - z^- = P(y^+ - y^-), z^+ \geq \mathbf{0}, z^- \geq \mathbf{0}\} \\ &= \min \{\bar{f}(E\tilde{y}) + b^T \tilde{y} : A\tilde{y} \leq a\} \end{aligned}$$

where  $\mathbf{1}$  and  $\mathbf{0}$  are the vector of ones and zeros, respectively,  $E = [I, -I, \mathbf{0}, \mathbf{0}]$ , and

$$\tilde{y} = \begin{pmatrix} y^+ \\ y^- \\ z^+ \\ z^- \end{pmatrix}, b = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \lambda \mathbf{1} \\ \lambda \mathbf{1} \end{pmatrix}, A = \begin{pmatrix} \mathbf{1}^T & \mathbf{1}^T & \mathbf{0} & \mathbf{0} \\ -I & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -I & \mathbf{0} & \mathbf{0} \\ P & -P & -I & I \\ -P & P & I & -I \\ \mathbf{0} & \mathbf{0} & -I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -I \end{pmatrix}, a = \begin{pmatrix} R \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}.$$

Note the constraint set  $Y = \{\tilde{y} \in \mathbb{R}^{4p^2} : A\tilde{y} \leq a\}$  is a compact polyhedral ( $\|z\|_2 = \|Py\|_2 \leq \|y\|_2 \leq B$  for some  $B < \infty$  since  $\|y\|_1 \leq R$ ), in which  $\bar{f}$  is strongly convex with modulus  $1/R^2$  because from the vectorized form of the sample version of (4.6)

$$\nabla^2 \bar{f}(y) = \Omega_D^{-1} \otimes \Omega_D^{-1} + S \otimes I \succeq \Omega_D^{-1} \otimes \Omega_D^{-1} = \text{diag}[(1/(\omega_{ii}\omega_{jj}))]$$

and

$$\omega_{ii} = |\omega_{ii}| \leq \sum_{k,l} |w_{kl}| = \|\Omega\|_1 = \|y\|_1 \leq R, \quad i = 1, \dots, p.$$

Define the diameters and radius of the sets

$$D = \max \{\|\tilde{y}_1 - \tilde{y}_2\|_2 : \tilde{y}_1, \tilde{y}_2 \in Y\} < \infty$$

$$D_E = \max \{\|E\tilde{y}_1 - E\tilde{y}_2\|_2 : \tilde{y}_1, \tilde{y}_2 \in Y\} \leq \|E\|_{op,2} D,$$

$$G = \max \{\|E\tilde{y}\|_2 : \tilde{y} \in Y\} \leq \|E\|_{op,2} D$$

where  $\|E\|_{op,r} = \sup_{v \neq \mathbf{0}} \|Ev\|_r / \|v\|_r$  is the operator  $r$ -norm. Let  $Y^* = \arg \min_{A\tilde{y} \leq a} \{\bar{f}(E\tilde{y}) + b^T \tilde{y}\}$ ,

which is nonempty and compact. Then, by [61, Lemma 2.5], for any  $\tilde{y} \in Y$ ,

$$\text{dist}^2(\tilde{y}, Y^*) \leq \kappa[\bar{f}(E\tilde{y} + b^T\tilde{y}) - \min_{\bar{y} \in Y} \{\bar{f}(E\bar{y}) + b^T\bar{y}\})] \quad (\text{C.1})$$

where  $\text{dist}(p, S) = \inf_{s \in S} \|p - s\|_2$  and

$$\kappa = \theta^2(\|b\|_2 D + 3GD_E + 2R^2(G^2 + 1)) \quad (\text{C.2})$$

for the Hoffman constant  $\theta$  that only depends on  $A$  and  $E$  [62]. In fact,  $\theta$  is characterized as

$$\theta = \max_{B \in \mathcal{B}} \{1/\lambda_{\min}(BB^T)\}$$

where  $\lambda_{\min}$  denotes the smallest eigenvalue and  $\mathcal{B}$  is the set of matrices constructed by linearly independent rows of  $[E^T, A^T]^T$ , and can be estimated from  $A$  and  $E$  [63].

Going back to the original variable, observe that for any  $y \in \mathbb{R}^{p^2}$  such that  $\|y\|_1 \leq R$ ,

$$y = E\tilde{y}, \quad \tilde{y} \in Y$$

and vice versa. We also see that  $\ell(y) = \bar{f}(E\tilde{y}) + b^T\tilde{y}$ . Let  $\tilde{y}^*$  be the projection of  $\tilde{y}$  onto  $Y^*$ . Since the minimizer  $y^*$  of  $\ell$  is unique, we also have  $y^* = E\tilde{y}^*$ . It also holds that

$$\ell^* \triangleq \min_{y \in \mathbb{R}^{p^2}} \ell(y) = \ell(y^*) = \bar{f}(E\tilde{y}^*) + b^T\tilde{y}^* = \min_{\bar{y} \in Y} \{\bar{f}(E\bar{y}) + b^T\bar{y}\}.$$

It follows from inequality (C.1.1) that for any  $\Omega \in \mathbb{R}^{p \times p}$  such that  $\|\Omega\|_1 \leq R$ ,

$$\text{dist}^2(\Omega, X^*) \leq 2\kappa[\ell(\Omega) - \ell^*],$$

i.e., a Łojasiewicz error bound inequality holds.

Then, from [64, Theorem 5], the Kurdyka-Łojasiewicz (KL) inequality

$$\varphi'(\ell(\Omega) - \ell^*) \text{dist}(0, \partial \ell(\Omega)) \geq 1 \quad (\text{C.3})$$

holds with  $\varphi(s) = 2\sqrt{2\kappa s}$  for all  $\Omega$  such that  $\|\Omega\|_1 \leq R$  and  $\ell(\Omega) > \ell^*$ . The inverse  $\psi : [0, \infty) \ni y \mapsto y^2/(8\kappa)$  of  $\varphi$  satisfies Assumption (A) of [64].

If  $\tau_t \in [\underline{\tau}, \bar{\tau}]$  with  $0 < \underline{\tau} \leq \bar{\tau} < 2/L$ , then [64, Proposition 13] asserts that Assumptions (H1) and (H2) of [64] are satisfied with  $a = 1/\bar{\tau} - L/2$  and  $b = 1/\underline{\tau} + L$ . The nonasymptotic complexity bound (4.14) follows immediately from [64, Corollary 20].

If  $\tau_t$  is chosen by the line search, then we have

$$f(\Omega^{(t+1)}) \leq f(\Omega^{(t)}) + \langle \nabla \bar{f}(\Omega^{(t)}), \Omega^{(t+1)} - \Omega^{(t)} \rangle + \frac{1}{2\tau_t} \|\Omega^{(t+1)} - \Omega^{(t)}\|_F^2$$

where

$$\Omega^{(t+1)} = \arg \min_{\Omega} \langle \nabla f(\Omega^{(t)}), \Omega - \Omega^{(t)} \rangle + \frac{1}{2\tau_t} \|\Omega - \Omega^{(t)}\|_F^2 + g(\Omega).$$

From the convexity of  $g$ , the latter is equivalent to

$$-\nabla f(\Omega^{(t)}) - \tau_t^{-1}(\Omega^{(t+1)} - \Omega^{(t)}) \in \partial g(\Omega^{(t+1)}), \quad (\text{C.4})$$

which implies

$$\begin{aligned} g(\Omega^{(t)}) &\leq g(\Omega^{(t+1)}) + \langle -\nabla f(\Omega^{(t)}) - \tau_t^{-1}(\Omega^{(t+1)} - \Omega^{(t)}), \Omega^{(t)} - \Omega^{(t+1)} \rangle \\ &= g(\Omega^{(t+1)}) + \langle \nabla f(\Omega^{(t)}), \Omega^{(t+1)} - \Omega^{(t)} \rangle + \tau_t^{-1} \|\Omega^{(t+1)} - \Omega^{(t)}\|_F^2. \end{aligned}$$

Combining the above two inequalities yields

$$f(\Omega^{(t+1)}) + g(\Omega^{(t+1)}) + \frac{1}{2\tau_t} \|\Omega^{(t+1)} - \Omega^{(t)}\|_F^2 \leq f(\Omega^{(t)}) + g(\Omega^{(t)}).$$

Since  $\tau_t \leq \tau_0$  for all  $t$ , Assumption (H1) of [64] is satisfied with  $a = 1/(2\tau_0)$ . Now from the condition (C.4) there is  $s^{(t+1)} \in \partial g(\Omega^{(t+1)})$  such that  $s^{(t+1)} + \nabla f(\Omega^{(t)}) + \tau_t^{-1}(\Omega^{(t+1)} - \Omega^{(t)}) = 0$ . This entails

$$\begin{aligned} \|s^{(t+1)} + \nabla f(\Omega^{(t+1)})\|_F &\leq \|s^{(t+1)} + \nabla f(\Omega^{(t)})\|_F + \|\nabla f(\Omega^{(t)}) - \nabla f(\Omega^{(t+1)})\|_F \\ &\leq \tau_t^{-1} \|\Omega^{(t+1)} - \Omega^{(t)}\|_F + L \|\Omega^{(t)} - \Omega^{(t+1)}\|_F \\ &\leq (\tau_{\min}^{-1} + L) \|\Omega^{(t+1)} - \Omega^{(t)}\|_F \end{aligned}$$

since  $\tau_t \geq \tau_{\min}$ . Invoking that  $s^{(t+1)} + \nabla f(\Omega^{(t+1)}) \in \partial \ell(\Omega^{(t+1)})$ , we see that Assumption (H2) of [64] is satisfied with  $b = \tau_{\min}^{-1} + L$ . The nonasymptotic complexity bound (4.14) follows again from [64, Corollary 20].

## C.2 Software Package

We present a user-friendly guide for utilizing the ACCORD algorithm, which will be made available as a software package called *gaccord* on PyPI. *gaccord* is implemented in C++ and Python, using *pybind11* for seamless binding. Additionally, for ease of development, we offer access to a pure Python version known as *pyaccord* within the package. Furthermore, the package's API closely aligns with that of the well-known machine learning library *scikit-learn*, ensuring convenient accessibility. Installation can be completed with the following command

```
pip install gaccord
```

Given a standardized dataset  $X$ , where rows and columns correspond to samples and features, respectively, the precision matrix can be estimated with the following code snippet

```
1 from gaccord import GraphicalAccord
2
3 model = GraphicalAccord(lam1=0.1, lam2=0, epstol=1e-5)
4 model.fit(X)
5 Theta = model.precision_
```

In this code, *Theta* represents the estimated precision matrix, with its nonzero elements stored in compressed sparse column format. Moreover, we provide cross-validation based on pBIC through the *GraphicalAccordCV* class.

```
1 from gaccord import GraphicalAccordCV
2
3 model_cv = GraphicalAccord(lam1=[0.1, 0.2, 0.5], lam2=0, epstol=1e-5)
4 model_cv.fit(X)
5 Theta, cv_results = model_cv.precision_, model_cv.cv_results_
```

# Bibliography

- [1] G. I. Allen, *Handbook of graphical models: by marloes maathuis, mathias drton, steffen lauritzen, and martin wainwright, eds. boca raton, fl: Chapman & hall/crc press, 2018, xviii+ 536 pp., 140.00(h), isbn : 978 – 1 – 49 – 878862 – 5.*, 2020.
- [2] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988.
- [3] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1. MIT press Cambridge, 2016.
- [4] J. Pearl *et. al.*, *Models, reasoning and inference*, Cambridge, UK: CambridgeUniversityPress **19** (2000), no. 2.
- [5] T. S. Verma and J. Pearl, *Equivalence and synthesis of causal models*, in *Probabilistic and Causal Inference: The Works of Judea Pearl*, pp. 221–236. 2022.
- [6] P. Spirtes, C. N. Glymour, R. Scheines, and D. Heckerman, *Causation, prediction, and search*. MIT press, 2000.
- [7] R. W. Robinson, *Counting unlabeled acyclic digraphs*, in *Combinatorial mathematics V*, pp. 28–43. Springer, 1977.
- [8] P. Spirtes and C. Glymour, *An algorithm for fast recovery of sparse causal graphs*, *Social science computer review* **9** (1991), no. 1 62–72.
- [9] M. Chickering, D. Heckerman, and C. Meek, *Large-sample learning of bayesian networks is np-hard*, *Journal of Machine Learning Research* **5** (2004) 1287–1330.
- [10] D. M. Chickering, *Optimal structure identification with greedy search*, *Journal of machine learning research* **3** (2002), no. Nov 507–554.
- [11] J. A. Gámez, J. L. Mateo, and J. M. Puerta, *Learning bayesian networks by hill climbing: efficient methods based on progressive restriction of the neighborhood*, *Data Mining and Knowledge Discovery* **22** (2011) 106–148.

- [12] X. Zheng, B. Aragam, P. K. Ravikumar, and E. P. Xing, *Dags with no tears: Continuous optimization for structure learning*, *Advances in neural information processing systems* **31** (2018).
- [13] G. Raskutti and C. Uhler, *Learning directed acyclic graph models based on sparsest permutations*, *Stat* **7** (2018), no. 1 e183.
- [14] Q. Ye, A. Amini, and Q. Zhou, *Optimizing regularized cholesky score for order-based learning of bayesian networks*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).
- [15] C. Squires, J. Amaniampong, and C. Uhler, *Efficient permutation discovery in causal dags*, *arXiv preprint arXiv:2011.03610* (2020).
- [16] I. Tsamardinos, L. E. Brown, and C. F. Aliferis, *The max-min hill-climbing bayesian network structure learning algorithm*, *Machine learning* **65** (2006) 31–78.
- [17] M. Pourahmadi, *Joint mean-covariance models with applications to longitudinal data: Unconstrained parameterisation*, *Biometrika* **86** (1999), no. 3 677–690.
- [18] S. Van de Geer, P. Bühlmann, *et. al.*,  $\ell_0$ -penalized maximum likelihood for sparse directed acyclic graphs, *Annals of Statistics* **41** (2013), no. 2 536–567.
- [19] P. Cisneros-Velarde, A. Petersen, and S.-Y. Oh, *Distributionally robust formulation and model selection for the graphical lasso*, in *International Conference on Artificial Intelligence and Statistics*, pp. 756–765, PMLR, 2020.
- [20] C. Tran, P. Cisneros-Velarde, S.-Y. Oh, and A. Petersen, *Family-wise error rate control in gaussian graphical model selection via distributionally robust optimization*, *Stat* **11** (2022), no. 1 e477.
- [21] M. Yannakakis, *Computing the minimum fill-in is np-complete*, *SIAM Journal on Algebraic Discrete Methods* **2** (1981), no. 1 77–79.
- [22] P. R. Amestoy, T. A. Davis, and I. S. Duff, *An approximate minimum degree ordering algorithm*, *SIAM Journal on Matrix Analysis and Applications* **17** (1996), no. 4 886–905.
- [23] D. J. Rose, *Symmetric elimination on sparse positive definite systems and the potential flow network problem*. PhD thesis, Harvard University, 1970.
- [24] H. Rue and L. Held, *Discrete spatial variation*, *Handbook of spatial statistics* (2010) 171–200.
- [25] D. J. Rose, *A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations*, in *Graph theory and computing*, pp. 183–217. Elsevier, 1972.

- [26] M. Sioutis and J.-F. Condotta, *Tackling large qualitative spatial networks of scale-free-like structure*, in *Artificial Intelligence: Methods and Applications: 8th Hellenic Conference on AI, SETN 2014, Ioannina, Greece, May 15-17, 2014. Proceedings 8*, pp. 178–191, Springer, 2014.
- [27] J. Friedman, T. Hastie, and R. Tibshirani, *Sparse inverse covariance estimation with the graphical lasso*, *Biostatistics* **9** (2008), no. 3 432–441.
- [28] D. Chicco and G. Jurman, *The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation*, *BMC genomics* **21** (2020) 1–13.
- [29] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, vol. 2. Springer, 2009.
- [30] A.-L. Barabási and R. Albert, *Emergence of scaling in random networks*, *science* **286** (1999), no. 5439 509–512.
- [31] 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** (2005), no. 5721 523–529.
- [32] B. Abramson, J. Brown, W. Edwards, A. Murphy, and R. L. Winkler, *Hailfinder: A bayesian system for forecasting severe weather*, *International Journal of Forecasting* **12** (1996), no. 1 57–71.
- [33] D. E. Heckerman and B. N. Nathwani, *An evaluation of the diagnostic accuracy of pathfinder*, *Computers and Biomedical Research* **25** (1992), no. 1 56–74.
- [34] R. J. Little and D. B. Rubin, *Statistical analysis with missing data*, vol. 793. John Wiley & Sons, 2019.
- [35] E. Gao, I. Ng, M. Gong, L. Shen, W. Huang, T. Liu, K. Zhang, and H. Bondell, *Missdag: Causal discovery in the presence of missing data with continuous additive noise models*, *arXiv preprint arXiv:2205.13869* (2022).
- [36] J. Friedman, T. Hastie, and R. Tibshirani, *Regularization paths for generalized linear models via coordinate descent*, *Journal of statistical software* **33** (2010), no. 1 1.
- [37] K. Khare, S.-Y. Oh, and B. Rajaratnam, *A convex pseudolikelihood framework for high dimensional partial correlation estimation with convergence guarantees*, *Journal of the Royal Statistical Society: Series B: Statistical Methodology* (2015) 803–825.

- [38] J. Peng, P. Wang, N. Zhou, and J. Zhu, *Partial correlation estimation by joint sparse regression models*, *Journal of the American Statistical Association* **104** (2009), no. 486 735–746.
- [39] N. Meinshausen and P. Bühlmann, *High-dimensional graphs and variable selection with the lasso*, *The Annals of Statistics* **34** (2006), no. 3 1436–1462.
- [40] O. Banerjee, L. El Ghaoui, and A. d’Aspremont, *Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data*, *The Journal of Machine Learning Research* **9** (2008) 485–516.
- [41] K. Lange, *MM optimization algorithms*. SIAM, 2016.
- [42] S. Oh, O. Dalal, K. Khare, and B. Rajaratnam, *Optimization methods for sparse pseudo-likelihood graphical model selection*, *Advances in Neural Information Processing Systems* **27** (2014).
- [43] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, *SIAM journal on imaging sciences* **2** (2009), no. 1 183–202.
- [44] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*, vol. 87. Springer Science & Business Media, 2003.
- [45] S. Ko, H. Zhou, J. J. Zhou, and J.-H. Won, *High-performance statistical computing in the computing environments of the 2020s*, *Statistical Science* **37** (2022), no. 4 494–518.
- [46] H. H. Bauschke, P. L. Combettes, *et. al.*, *Convex analysis and monotone operator theory in Hilbert spaces*, vol. 408. Springer, 2011.
- [47] G. Li and T. K. Pong, *Calculus of the exponent of kurdyka–łojasiewicz inequality and its applications to linear convergence of first-order methods*, *Foundations of computational mathematics* **18** (2018), no. 5 1199–1232.
- [48] P. Koanantakool, A. Ali, A. Azad, A. Buluc, D. Morozov, L. Olikar, K. Yelick, and S.-Y. Oh, *Communication-avoiding optimization methods for distributed massive-scale sparse inverse covariance estimation*, in *International Conference on Artificial Intelligence and Statistics*, pp. 1376–1386, PMLR, 2018.
- [49] R. Foygel and M. Drton, *Extended bayesian information criteria for gaussian graphical models*, *Advances in neural information processing systems* **23** (2010).
- [50] H. H. Bauschke and J. M. Borwein, *On projection algorithms for solving convex feasibility problems*, *SIAM review* **38** (1996), no. 3 367–426.
- [51] L. Li and K.-C. Toh, *An inexact interior point method for  $l_1$ -regularized sparse covariance selection*, *Mathematical Programming Computation* **2** (2010) 291–315.

- [52] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, P. K. Ravikumar, and R. Poldrack, *Big  $\mathcal{E}$  quic: Sparse inverse covariance estimation for a million variables*, *Advances in neural information processing systems* **26** (2013).
- [53] H. Pang, H. Liu, and R. J. Vanderbei, *The fastclime package for linear programming and large-scale precision matrix estimation in r.*, *Journal of Machine Learning Research* (2014).
- [54] A. Ally, M. Balasundaram, R. Carlsen, E. Chuah, A. Clarke, N. Dhalla, R. A. Holt, S. J. Jones, D. Lee, Y. Ma, *et. al.*, *Comprehensive and integrative genomic characterization of hepatocellular carcinoma*, *Cell* **169** (2017), no. 7 1327–1341.
- [55] S. Ko, G. X. Li, H. Choi, and J.-H. Won, *Computationally scalable regression modeling for ultrahigh-dimensional omics data with parprox*, *Briefings in Bioinformatics* **22** (2021), no. 6 bbab256.
- [56] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker, *Cytoscape: a software environment for integrated models of biomolecular interaction networks*, *Genome research* **13** (2003), no. 11 2498–2504.
- [57] F. J. Martin, M. R. Amode, A. Aneja, O. Austine-Orimoloye, A. G. Azov, I. Barnes, A. Becker, R. Bennett, A. Berry, J. Bhai, *et. al.*, *Ensembl 2023*, *Nucleic acids research* **51** (2023), no. D1 D933–D941.
- [58] J. M. Steele, *The Cauchy-Schwarz master class: an introduction to the art of mathematical inequalities*. Cambridge University Press, 2004.
- [59] S. Ko, D. Yu, and J.-H. Won, *Easily parallelizable and distributable class of algorithms for structured sparsity, with optimal acceleration*, *Journal of Computational and Graphical Statistics* **28** (2019), no. 4 821–833.
- [60] S. Łojasiewicz, *Division d’une distribution par une fonction analytique de variables réelles*, *Comptes Rendus Hebdomadaires Des Seances de l’Académie Des Sciences* **246** (1958), no. 5 683–686.
- [61] A. Beck and S. Shtern, *Linearly convergent away-step conditional gradient for non-strongly convex functions*, *Mathematical Programming* **164** (2017) 1–27.
- [62] A. J. Hoffman, *On approximate solutions of systems of linear inequalities*, in *Selected Papers Of Alan J Hoffman: With Commentary*, pp. 174–176. World Scientific, 2003.
- [63] C. Zualinescu, *Sharp estimates for hoffman’s constant for systems of linear inequalities and equalities*, *SIAM Journal on Optimization* **14** (2003), no. 2 517–533.

- [64] J. Bolte, T. P. Nguyen, J. Peypouquet, and B. W. Suter, *From error bounds to the complexity of first-order descent methods for convex functions*, *Mathematical Programming* **165** (2017) 471–507.