Journal of Data Science 21 (2), 354–367
April 2023
DOI: 10.6339/23-JDS1097
Statistical Data Science

FROSTY: A High-Dimensional Scale-Free Bayesian Network Learning Method

Joshua Bang[1,*] and Sang-Yun Oh[2]

[1]*Department of Statistics and Applied Probability, University of California, Santa Barbara, Santa Barbara, CA, USA*
[2]*Lawrence Berkeley National Lab, Berkeley, CA, USA*

Abstract

We propose a scalable Bayesian network learning algorithm based on sparse Cholesky decomposition. 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.

Keywords *Bayesian networks; robust selection; sparse Cholesky decomposition; structure learning*

1 Introduction

Bayesian network is a type of graphical model which encodes conditional independencies between random variables as directed acyclic graphs (DAGs) (Pearl, 2014). 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. In addition to compactly representing complex relationships, Bayesian networks also provide computational efficiency in sampling and inference that would otherwise require intractable summations (Goodfellow et al., 2016). Bayesian networks find their applications in a wide range of disciplines: speech recognition (Huang et al., 2001), genetics (Friedman, 2004), computational biology (Sachs et al., 2005), image understanding (Luo et al., 2005), and many more (Wainwright and Jordan, 2008).

Despite the usefulness of these models, learning the DAG structure from observational data is challenging. In general, one can identify a Bayesian network only up to a Markov equivalence class with observational data alone (Verma and Pearl, 2022). Computationally, the number of possible Bayesian networks grows super-exponentially as the number of vertices increases (Robinson, 1977), which makes analyzing high-dimensional data beyond hundreds of variables challenging even for many state-of-the-art methods in the literature. Furthermore, the performance of these algorithms heavily depends on initial input and parameter tuning, which are often not addressed in detail. These challenges necessitate a scalable, simple-to-tune, off-the-shelf learning algorithm for large Bayesian networks.

*Corresponding author. Email: joshuaybang@gmail.com.

© 2023 The Author(s). Published by the School of Statistics and the Center for Applied Statistics, Renmin University of China. Open access article under the CC BY license.
Received July 31, 2022; Accepted March 7, 2023

2 Preliminaries

Consider p-variate Gaussian random vector $X = (X_1, \ldots, X_p)^T \sim N(0, \Sigma)$ with covariance structure Σ. 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 $V = \{X_1, \ldots, X_p\}$ or simply $\{1, \ldots, p\}$ is a set of vertices and $E = \{(i, j) : i \to j \; \forall \, i, j \in V\}$ is a set of directed edges. $pa_G(j) = \{i \in V : i \to j\}$ denotes the parent set of vertex j in the DAG G.

Assuming linear dependencies between variables of X, structural equation modeling writes

$$X_j = \sum_{i \in pa_G(j)} \beta_{ij} X_i + \epsilon_j, \;\; j = 1, \ldots, p \tag{1}$$

where β_{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

$$X = B^T X + \epsilon \tag{2}$$

where $\epsilon = (\epsilon_1, \ldots, \epsilon_p)^T$ 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 $\Omega = \text{diag}(\omega_1^2, \ldots, \omega_p^2)$ be the diagonal matrix with error variances. Then it is straightforward to show that Θ can be expressed as a function of B and Ω

$$\Theta(B, \Omega) = (I - B)\Omega^{-1}(I - B)^T. \tag{3}$$

Since zero correlation implies conditional independence under multivariate Gaussianity, the zeros of the precision matrix Θ coincides with the missing edges in the corresponding undirected graph. Since zero/nonzero patterns in Θ and B coincide with edges in undirected graph and DAG, respectively, we will use these terms interchangeably for simplicity.

Furthermore, Θ can be translated into its canonical candidate DAG by computing its modified Cholesky factors: $\Theta = TDT^T$, where $T = I - B$ is computed from ordinary Cholesky factors $\Theta = LL^T$. The lower triangular matrix T with unit diagonal entries is a scaled version of L, and the diagonal matrix $D = \Omega^{-1}$ has $D_{ii} = L_{ii}^2$ for $i \in V$ as the diagonal entries. Also, notice that the lower triangular shape of B ensures the acyclicity condition since any variable j is linearly dependent only on preceding variables $i < j$, where $i, j \in V$ as in (1), and coefficient $\beta_{ij} \neq 0$ implies $(i, j) \in E$ (Pourahmadi, 1999).

The Cholesky factor L is dependent on the ordering of rows and columns of Θ, so it is often desired to estimate the DAG representing the distribution with minimal number of edges called *minimal-edge I-MAP* (Van de Geer et al., 2013). In fact, in the noiseless Gaussian setting, recovering the minimal-edge I-MAP amounts to finding a variable ordering π so that the Cholesky factor in $\Theta_\pi = L_\pi L_\pi^T$ is the sparsest (Raskutti and Uhler, 2018). Figure 1 illustrates the impact variable ordering of Θ has on the zero/nonzero patterns in the resulting Cholesky factor. Recently published methods such as ARCS (Ye et al., 2020) and RFD (Squires et al., 2020) perform DAG recovery by estimating the permutation ordering π along with the corresponding Cholesky factor L.

However, even when the variable ordering is known, the quality of the precision matrix Θ greatly affects the performance of DAG estimation. This is due to the sequential procedure of Cholesky decomposition that is prone to accumulate errors. We will illustrate empirical evidence of this in Section 3 and introduce distributionally robust optimization to tightly control the number of false edges in Θ.

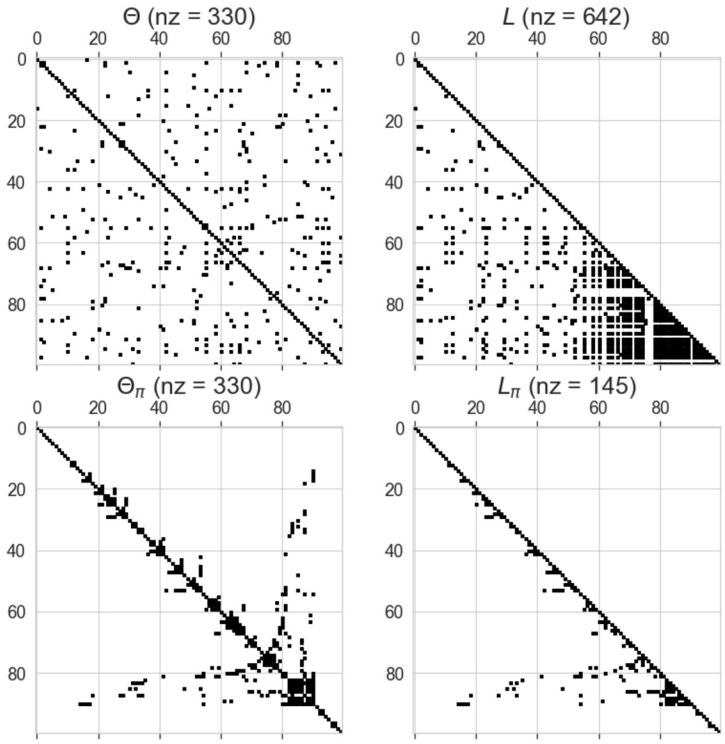


Figure 1: The top row represents an unordered sparse undirected graph and its Cholesky factor while the bottom row represents those ordered by sparse Cholesky decomposition. nz indicates the number of nonzero entries.

As discussed, the performance of DAG estimation from data hinges on the quality of the precision matrix Θ and the permutation ordering π. Based on these observations, we propose a novel algorithm for DAG structure learning that we call FROSTY in Section 3. Two distinct features/advantages of FROSTY are its tuning simplicity and scalability to high-dimensional data. In Section 4, FROSTY is empirically evaluated in comparison to other recent DAG structure learning methods.

3 FROSTY for DAG Structure Learning

We propose a scalable Bayesian network learning algorithm for scale-free graphs, which we call *scale-Free Bayesian network learning with RObust Selection Tuning and sparse cholesky* (FROSTY).[1] At a high-level, FROSTY consists of two steps: input undirected graph Θ estimation step and DAG G recovery step using a sparse Cholesky decomposition. The pseudo-code for FROSTY is given in Algorithm 1. In the remainder of this section, we discuss the components of FROSTY in detail.

[1]The term "scale-free" in the title will be motivated in latter part of this section.

Algorithm 1: FROSTY algorithm.

Input : Dataset X, confidence level α
Output: Coefficient matrix of Bayesian network B

Step 1: Estimate undirected graph Θ

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_π, $P_\pi = \text{AMD}(\Theta)$.

4 $B = P_\pi^T(\text{diag}(L_\pi) - L_\pi)\text{diag}(L_\pi)^{-1}P_\pi$.

5 **return** B

3.1 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 $\mathcal{S} = \{\mathcal{P} : \mathcal{D}_c(\mathcal{P}, \mathcal{P}_n) \leqslant \lambda\}$, 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* λ. 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 π on (U, V) supported on $\mathbb{S}_p \times \mathbb{S}_p$. π_U and π_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 3.1. *The ℓ_1-regularized optimization problem of L can be recasted as the Distributionally Robust Optimization (DRO) formulation,*

$$\underbrace{\min_{L\in\mathbb{L}_p} \sup_{\mathcal{P}:D_c(\mathcal{P},\mathcal{P}_n)\leqslant\lambda} E_{\mathcal{P}}[\ell(S; L)]}_{\text{DRO formulation}} = \underbrace{\min_{L\in\mathbb{L}_p} \{\ell(S; L) + \lambda\|\text{vec}(LL^T)\|_1\}}_{\ell_1\text{-regularized formulation}}. \tag{4}$$

Theorem 3.1 shows the equivalence of parameter λ between the amount of regularization (right-hand side) of (4) and the ambiguity size of the DRO formulation (left-hand side). Utilizing this equivalence, we suggest a selection criterion for λ based on $(1-\alpha)$-confidence level for L by modifying the derivation found in Cisneros et al. (2020). 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.,

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

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

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

In fact, since $\mathcal{P}_n$ weakly converges to the true probability measure $\mathcal{P}_0$ for any λ, $\mathcal{C}_n(\lambda)$ 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 λ_α for ℓ_1-regularized optimization problem of L can be completely determined by a user-specified error tolerance α as in $(1-\alpha)$-confidence level for L.

Theorem 3.2. *The RWP function for L is*

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

and a robust selection for the ambiguity size λ of the DRO formulation for L in (4) is

$$\lambda_\alpha = \inf \{ \lambda > 0 : \mathbb{P}_0(\| \mathrm{vec}(S - (LL^T)^{-1}) \|_\infty \leqslant \lambda) \geqslant 1 - \alpha \} \tag{5}$$

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 3.1 and 3.2 is that estimating L can be achieved in two stages as in Algorithm 1: *Step 1* is the sparse estimation of Θ, and *Step 2* is the Cholesky factorization of $\Theta = LL^T$. Since the DAG structure in L corresponding to an undirected graph structure in Θ depends heavily on variable ordering as discussed in Section 2. A natural question to ask is whether λ_α is invariant under variable ordering permutations.

In the following lemma, we show that λ_α does not depend on variable ordering.

Lemma 3.3. *Let π be a variable ordering consistent with the minimal-edge I-MAP B_π and L_π be the corresponding Cholesky factor, i.e., $B_\pi = (\mathrm{diag}(L_\pi) - L_\pi)\mathrm{diag}(L_\pi)^{-1}$. For the precision matrix Θ, suppose P_π 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_π is equivalent to the RWP function for Θ, i.e,*

$$R_n(L_\pi) = \| \mathrm{vec}(S_\pi - (L_\pi L_\pi^T)^{-1}) \|_\infty = \| \mathrm{vec}(S - \Theta^{-1}) \|_\infty = R_n(\Theta). \tag{6}$$

A direct consequence of Lemma 3.3 is that Θ only needs to be estimated once independent of the variable ordering π since $R_n(L_\pi) = R_n(\Theta)$. Additionally, Lemma 3.3 justifies estimation λ_α by Robust Selection (RobSel) algorithm of Cisneros et al. (2020) given in Algorithm 2.

RobSel algorithm only requires bootstrapped sample covariance matrices to determine λ_α, without requiring computationally expensive cross-validation. Furthermore, in the recent work of Tran et al. (2022), it has been shown that α in RobSel is the upper bound of the asymptotic family-wise error rate of estimating at least one erroneous nonzero in Θ. Since the DRO formulation is to minimize the worst-case scenario, RobSel tends to be conservative even for large values of α, and, in fact, a large value of α is recommended, e.g., $\alpha = 0.99$ was routinely used with good results. We remark that using RobSel-tuned Θ as inputs might also improve other methods. In our numerical experiments, RFD (Squires et al., 2020) estimates improve when RobSel-tuned Θ is provided as input.

3.2 Recovering Variable Ordering

To ultimately recover the minimal-edge I-MAP B_π from the precision matrix Θ, we also need to estimate a variable ordering π that leads to the sparsest Cholesky factor L_π. Since recovering such a permutation π is known to be NP-complete (Yannakakis, 1981), a heuristic method called Approximate Minimum Degree ordering (AMD) (Amestoy et al., 1996) is used in FROSTY. AMD algorithm is a more efficient variant of the minimum degree ordering (Rose, 1970) 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 (Rue and Held, 2010). Furthermore, the minimum fill-in problem, i.e. sparse Cholesky decomposition, is equivalent to the minimum chordal problem (Rose, 1972), and, coincidentally, scale-free graphs are natural candidates for the problem (Sioutis and Condotta, 2014). 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 Θ.

3.3 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

Algorithm 2: RobSel algorithm (Cisneros et al., 2020).

Input : Dataset X, confidence level α
Output: Graphical lasso regularization parameter λ_α

for $k \leftarrow 1, \ldots, b$ **do**
  Obtain bootstrap sample $X^*_{1k}, \ldots, X^*_{nk}$ by sampling uniformly with replacement.
  Compute $R^*_{n,k} = \left\| A^*_{n,k} - A_n \right\|_q$, with empirical covariance $A^*_{n,k}$ of bootstrap sample.
end
Set λ_α as bootstrap order statistic $R^*_{n,((b+1)(1-\alpha))}$.
return λ_α

computation time of graphical lasso is known to be significantly shorter for sparse problems (Friedman et al., 2008).

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 $O(p^3)$.

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 (Barabási and Albert, 1999) 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 (3) whose edge weights are chosen from 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.

Comparison metrics – structural hamming distance (SHD), false discovery rate (FDR), Matthews correlation coefficient (MCC) – are evaluated on the adjacency matrix of the Completed Partially Directed Acyclic Graph (CPDAG) computed from the estimated DAGs. We use the CPDAG to avoid favoring a structure over another when they are statistically indistinguishable (Spirtes et al., 2000).

Since RFD requires Θ as an input, we provide two different estimate schemes: RS and CV indicate estimated Θ by RobSel and estimated Θ by Cross-Validation, respectively. For RFD (CV), we used a 5-fold CV with a 20 evenly spaced grid of λ 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. All performance metrics in figures are averaged over 20 randomly generated datasets unless otherwise noted.

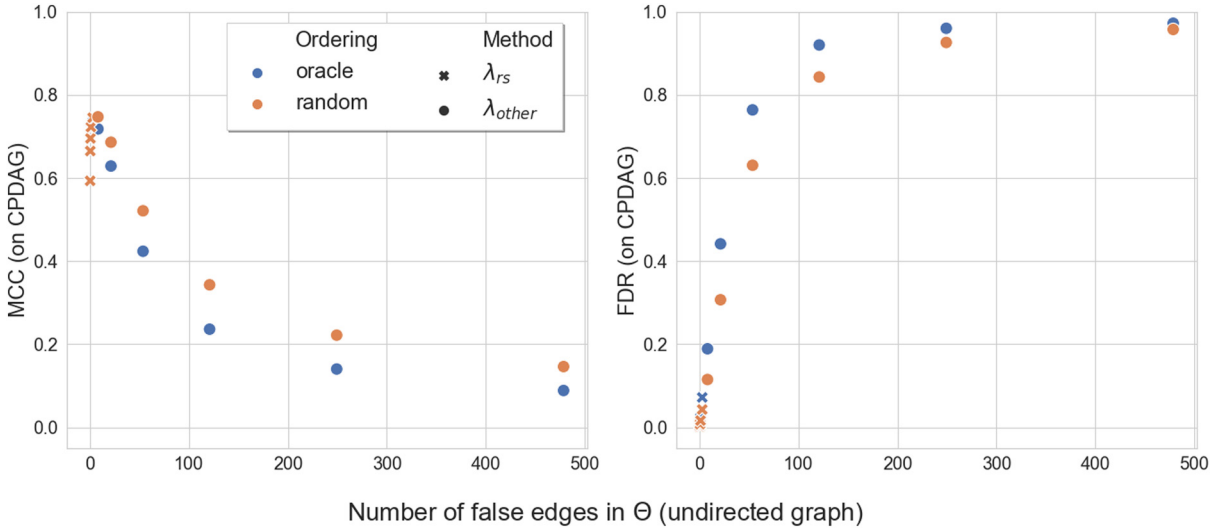


Figure 2: The effect of false edge control of RobSel on DAG estimation. For RobSel, a grid of $\lambda_{\text{rs}} = [\lambda_{0.1}, \lambda_{0.3}, \lambda_{0.5}, \lambda_{0.7}, \lambda_{0.9}]$ is used.

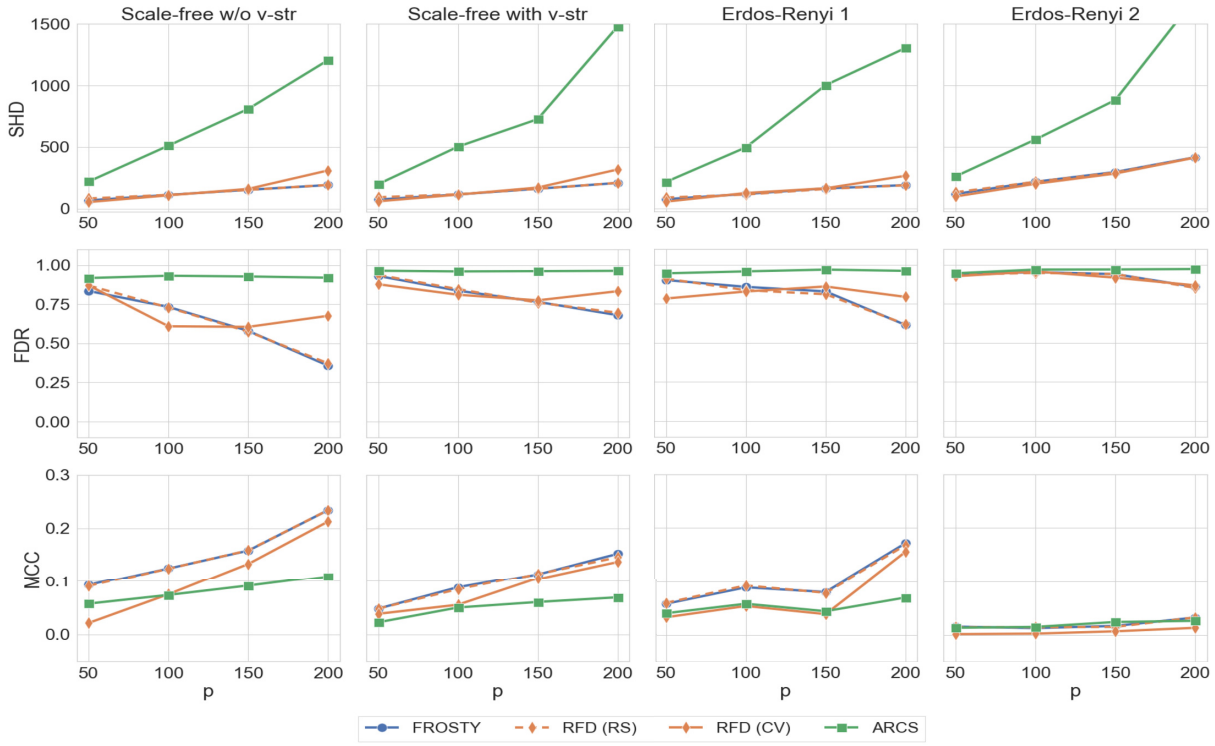
4.1 Empirical Motivation for RobSel

Figure 2 illustrates the benefits of tightly controlling the number of false edges in Θ 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 $n = 200$, Θ's are estimated with graphical lasso given two grids of λ. $\lambda_{\text{rs}} = [\lambda_{0.1}, \lambda_{0.3}, \lambda_{0.5}, \lambda_{0.7}, \lambda_{0.9}]$ are chosen by RobSel where the subscripts represent the values of α used, and $\lambda_{\text{other}} = [\lambda_1, \ldots, \lambda_{10}]$ are 10 evenly spaced values between $(0, \lambda_{0.9})$. However, only the 6 largest values in λ_{other} are shown in Figure 2 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.

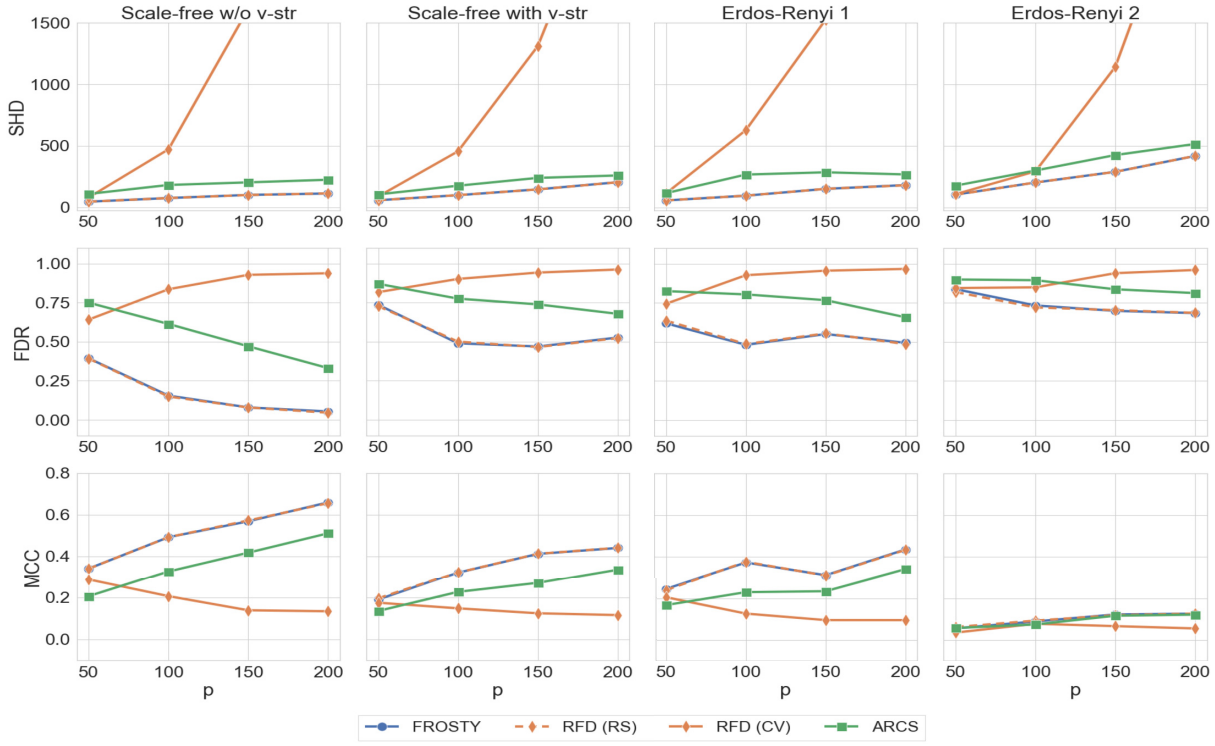
When the regularization parameter λ is chosen by RobSel, graphical lasso conservatively estimates Θ resulting in low FDR with relatively high MCC on its CPDAG. However, notice that FDR surges and MCC drops rapidly when we select λ smaller than the smallest one chosen by RobSel, i.e. $\lambda_{0.9}$. This empirically motivates one to choose RobSel over cross-validation, which always selects smaller λ than RobSel (Cisneros et al., 2020) and has a tendency toward overfitting (Hastie et al., 2009).

4.2 Synthetic Graphs

Scale-Free Graph Figure 3 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 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



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



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

Figure 3: Performance comparison for various graph structures.

Table 1: Runtime analysis.

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)

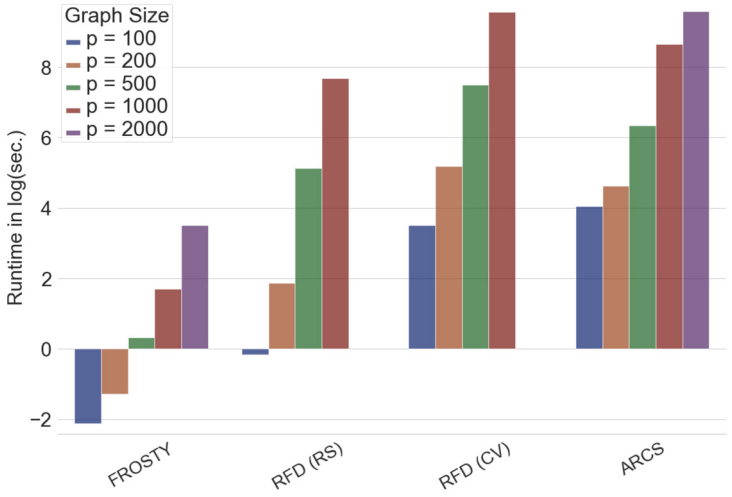


Figure 4: Log-scale runtime comparison between methods with varying graph sizes. The results are averaged over 20 runs except for the cases of $p = 1000$ and 2000, each of which is run only once.

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 1 and Figure 4 show the superb scalability of FROSTY.

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 3, 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.

4.3 Real Networks

We also investigate a various real networks. The flow cytometry data (Sachs et al., 2005) 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

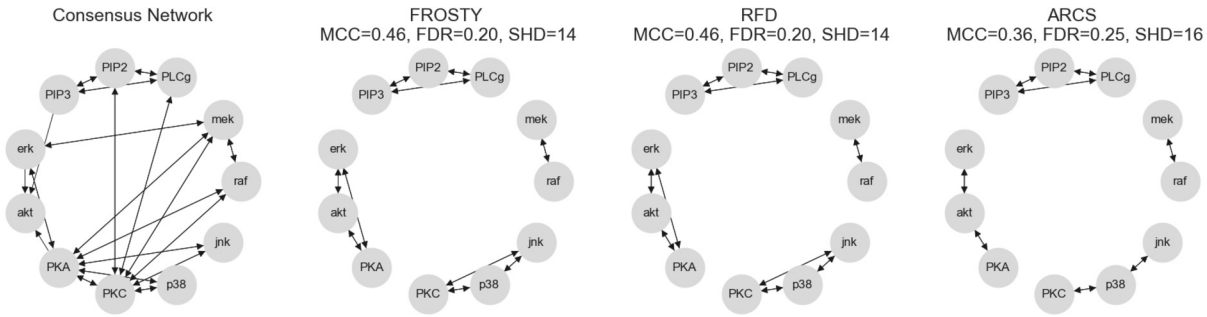


Figure 5: Consensus network (as in CPDAG) of flow cytometry data and estimated CPDAGs.

and Pathfinder networks from *bnlearn* repository. Hailfinder is a weather forecast system based on meteorological data with expert judgment (Abramson et al., 1996), and Pathfinder is an expert system that assists in diagnosing lymph-node pathology (Heckerman and Nathwani, 1992). 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: $n \in \{0.5p, 2p, 10p\}$.

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 5. 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 6. 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 7. Furthermore, Figure 7 also shows that FROSTY estimated true edges more consistently compared to ARCS. We can observe the estimated edges of ARCS for the true edges are lighter than those of FROSTY.

5 Conclusion

FROSTY estimates an undirected graph and converts it to a directed acyclic graph. The key contribution comes from the undirected graph estimation with computationally efficient parameter tuning that tightly controls false edges. It is interesting that FROSTY outperforms RFD whilst being essentially a subset class of RFD seeded with cross-validated input undirected graph indicates that many algorithms that take an undirected graph as an input can be significantly improved by adapting FROSTY's undirected graph estimation step. Furthermore, FROSTY is extremely scalable. For a large graph size of 2000 vertices, it estimates a Bayesian network less than a minute while it takes several hours for the other methods. FROSTY is also simple. There is practically one tuning parameter α, as in $(1-\alpha)$-confidence level for the Cholesky factor L, which can be chosen informatively without the need for computationally expensive and arbitrary grid-search with cross-validation.

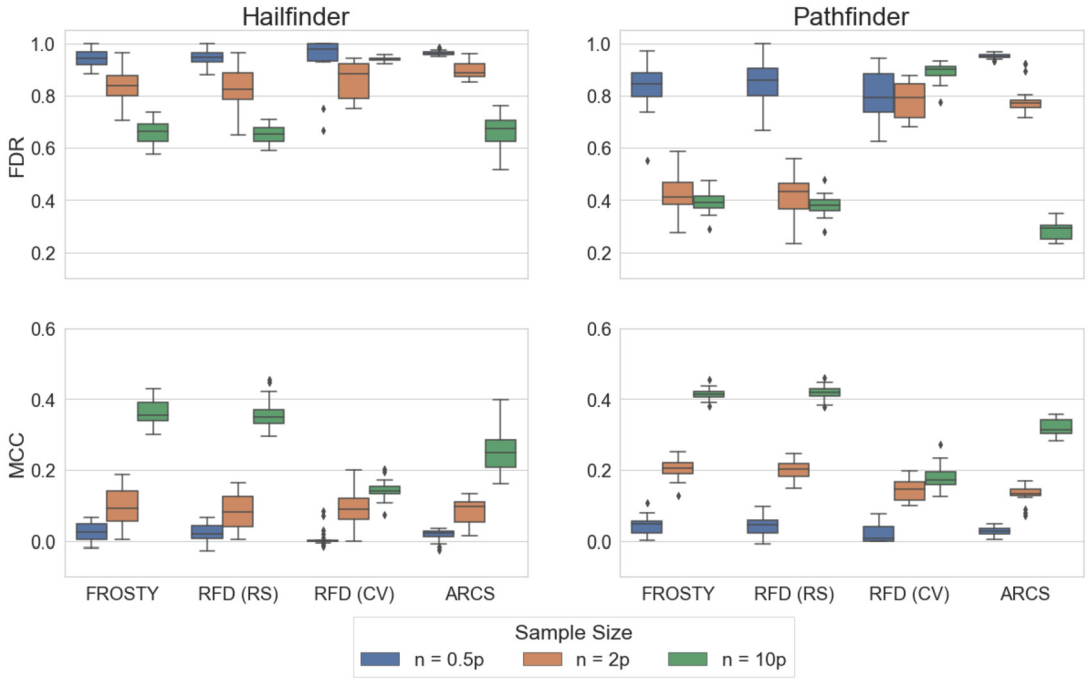


Figure 6: Performance comparison for Pathfinder and Hailfinder networks with varying sample sizes.

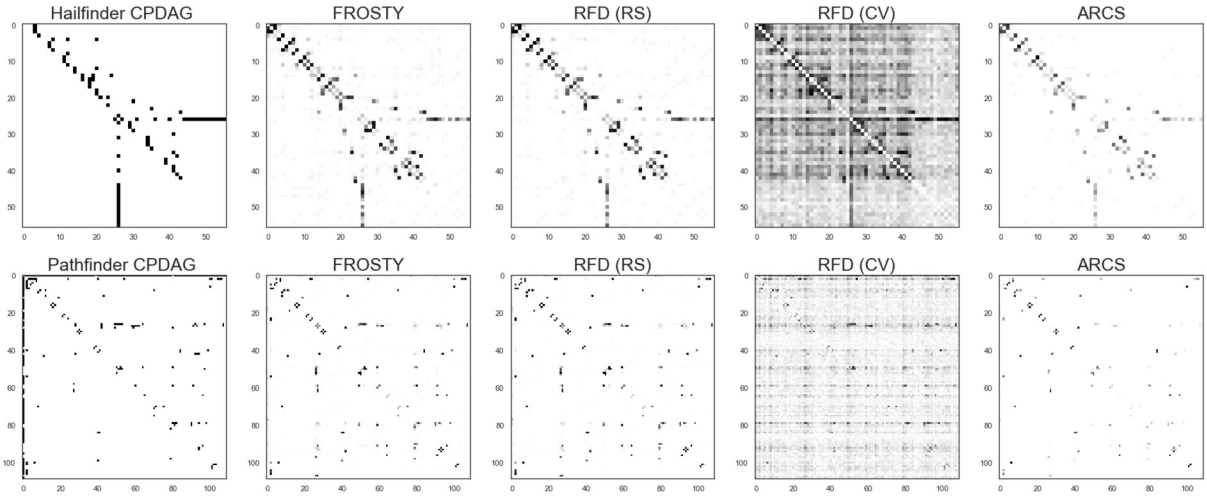


Figure 7: True and estimated CPDAGs for Hailfinder and Pathfinder networks. The edge inclusion percentage is represented by the grayscale. The results are based on sample size of $n = 10p$.

Supplementary Material

Supplementary material includes proofs of the theorems and the Python code for simulation.

References

Abramson B, Brown J, Edwards W, Murphy A, Winkler RL (1996). Hailfinder: A Bayesian system for forecasting severe weather. *International Journal of Forecasting*, 12(1): 57–71. https://doi.org/10.1016/0169-2070(95)00664-8

Amestoy PR, Davis TA, Duff IS (1996). An approximate minimum degree ordering algorithm. *SIAM Journal on Matrix Analysis and Applications*, 17(4): 886–905. https://doi.org/10.1137/S0895479894278952

Barabási AL, Albert R (1999). Emergence of scaling in random networks. *Science*, 286(5439): 509–512. https://doi.org/10.1126/science.286.5439.509

Cisneros-Velarde P, Petersen A, Oh SY (2020). Distributionally robust formulation and model selection for the graphical lasso. In: *International Conference on Artificial Intelligence and Statistics*, 756–765, PMLR.

Friedman J, Hastie T, Tibshirani R (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3): 432–441. https://doi.org/10.1093/biostatistics/kxm045

Friedman N (2004). Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659): 799–805. https://doi.org/10.1126/science.1094068

Goodfellow I, Bengio Y, Courville A, Bengio Y (2016). *Deep learning*, volume 1. MIT press, Cambridge.

Hastie T, Tibshirani R, Friedman JH, Friedman JH (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Heckerman DE, Nathwani BN (1992). An evaluation of the diagnostic accuracy of pathfinder. *Computers and Biomedical Research*, 25(1): 56–74. https://doi.org/10.1016/0010-4809(92)90035-9

Huang X, Acero A, Hon HW, Reddy R (2001). *Spoken language processing: A guide to theory, algorithm, and system development*. Prentice hall PTR.

Luo J, Savakis AE, Singhal A (2005). A Bayesian network-based framework for semantic image understanding. *Pattern recognition*, 38(6): 919–934. https://doi.org/10.1016/j.patcog.2004.11.001

Pearl J (2014). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier.

Pourahmadi M (1999). Joint mean-covariance models with applications to longitudinal data: Unconstrained parameterisation. *Biometrika*, 86(3): 677–690. https://doi.org/10.1093/biomet/86.3.677

Raskutti G, Uhler C (2018). Learning directed acyclic graph models based on sparsest permutations. *Stat*, 7(1): e183. https://doi.org/10.1002/sta4.183

Robinson RW (1977). Counting unlabeled acyclic digraphs. In: *Combinatorial Mathematics V: Proceedings of the Fifth Australian Conference, Held at the Royal Melbourne Institute of Technology, August 24–26, 1976*, 28–43. Springer.

Rose DJ (1970). Symmetric elimination on sparse positive definite systems and the potential flow network problem, Ph.D. thesis, Harvard University.

Rose DJ (1972). A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations. In: *Graph theory and computing*, 183–217. Elsevier.

Rue H, Held L (2010). Discrete spatial variation. In: *Handbook of spatial statistics*, 171–200.

Sachs K, Perez O, Pe'er D, Lauffenburger DA, Nolan GP (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721): 523–529. https://doi.org/10.1126/science.1105809

Sioutis M, Condotta JF (2014). 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*, 178–191. Springer.

Spirtes P, Glymour CN, Scheines R, Heckerman D (2000). *Causation, prediction, and search*. MIT press.

Squires C, Amaniampong J, Uhler C (2020). Efficient permutation discovery in causal dags. arXiv preprint: https://arxiv.org/abs/2011.03610.

Tran C, Cisneros-Velarde P, Oh SY, Petersen A (2022). Family-wise error rate control in Gaussian graphical model selection via distributionally robust optimization. *Stat*, 11(1): e477, Wiley Online Library.

Van de Geer S, Bühlmann P, et al. (2013). ℓ_0-penalized maximum likelihood for sparse directed acyclic graphs. *Annals of Statistics*, 41(2): 536–567.

Verma TS, Pearl J (2022). Equivalence and synthesis of causal models. In: *Probabilistic and Causal Inference: The Works of Judea Pearl*. 221–236.

Wainwright MJ, Jordan MI (2008). *Graphical models, exponential families, and variational inference*. Now Publishers Inc.

Yannakakis M (1981). Computing the minimum fill-in is np-complete. *SIAM Journal on Algebraic Discrete Methods*, 2(1): 77–79. https://doi.org/10.1137/0602010

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