Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JASA

Learning Sparse Causal Gaussian Networks With Experimental Intervention: Regularization and Coordinate Descent

Fei FU and Qing ZHOU

Causal networks are graphically represented by directed acyclic graphs (DAGs). Learning causal networks from data is a challenging problem due to the size of the space of DAGs, the acyclicity constraint placed on the graphical structures, and the presence of equivalence classes. In this article, we develop an L_1-penalized likelihood approach to estimate the structure of causal Gaussian networks. A blockwise coordinate descent algorithm, which takes advantage of the acyclicity constraint, is proposed for seeking a local maximizer of the penalized likelihood. We establish that model selection consistency for causal Gaussian networks can be achieved with the adaptive lasso penalty and sufficient experimental interventions. Simulation and real data examples are used to demonstrate the effectiveness of our method. In particular, our method shows satisfactory performance for DAGs with 200 nodes, which have about 20,000 free parameters. Supplementary materials for this article are available online.

KEY WORDS: Adaptive lasso; Experimental data; L_1 regularization; Penalized likelihood; Structure learning.

1. INTRODUCTION

Conditional independence structures among random variables are often visualized as graphical models, where the nodes represent the variables and the edges encode the relationships among them. Depending on whether the edges are directional or not, graphical models can be classified as either directed or undirected. The Bayesian network (BN) is a special type of graphical models, whose structure is represented by a directed acyclic graph (DAG). It has become a popular probabilistic model in many research areas, including computational biology, medical sciences, image processing, speech recognition, etc.

Learning the structure of a BN from data is an important and challenging problem in statistics. The major difficulty lies in the fact that the number of DAGs grows superexponentially in the number of nodes (Robinson 1973). A substantial amount of research has been devoted to the structure learning problem of BNs and many methods have been proposed. These methods can be roughly classified into two primary approaches. The constraint-based approach relies on a set of conditional independence tests. A well-known example in this category is the Peter-Clark (PC) algorithm proposed by Spirtes, Glymour, and Scheines (1993). Recently, Kalisch and Bühlmann (2007) considered the problem of estimating BNs with the PC algorithm and proposed an efficient implementation suitable for sparse high-dimensional DAGs. The second approach to learning BNs is score based, which attempts to find a DAG that maximizes some scoring function through a certain search strategy (Cooper and Herskovits 1992; Lam and Bacchus 1994; Heckerman, Geiger, and Chickering 1995) or sample DAGs from a Bayesian posterior distribution (Madigan and York 1995; Friedman and Koller 2003; Ellis and Wong 2008; Zhou 2011). Many algorithms in this category work well for graphs that do not have a large number of nodes. However, due to the size of the space of DAGs, they become computationally impractical for large networks.

In recent years, a number of researchers proposed to estimate the structure of graphical models through L_1-regularized likelihood approaches (lasso-type penalties). The L_1 penalty becomes popular because of the parsimonious solution it leads to as well as its computational tractability. Much of the research has focused on estimating undirected graphs with the L_1 penalty. Yuan and Lin (2007) proposed to maximize an L_1-penalized log-likelihood based on the "max-det" problem considered by Vandenberghe, Boyd, and Wu (1998), while Banerjee, El Ghaoui, and d'Aspremont (2008) employed a blockwise coordinate descent (CD) algorithm to solve the optimization problem. Friedman, Hastie, and Tibshirani (2008) built on the method of Banerjee, El Ghaoui, and d'Aspremont (2008) a remarkably efficient algorithm called the graphical lasso. Another computationally attractive method was developed by Meinshausen and Bühlmann (2006), where an undirected graph is constructed by fitting a lasso regression for each node separately.

Compared to undirected graphs, BNs have an attractive property: they can be used to represent causal relationships among random variables. Although some authors discussed the possibility of causal inference from observational data (Spirtes, Glymour, and Scheines 1993; Pearl 2000), most researchers agree that causal relations can only be reliably inferred using experimental data. Experimental interventions reveal causality among a set of variables by breaking down various connections in the underlying causal network. As for undirected graphs, sparsity in the structure of a causal BN is desired, which often gives more interpretable results. A natural generalization is to use the L_1 penalty in structure learning of causal BNs with experimental data. However, there are a number of difficulties for this seemingly natural generalization. First, existing theories on L_1-regularized estimation and penalized likelihood may not be directly applicable to structure learning of DAGs with

Fei Fu (E-mail: *fufei05@hotmail.com*) is Ph.D. Student, and Qing Zhou (E-mail: *zhou@stat.ucla.edu*) is Associate Professor, Department of Statistics, University of California, Los Angeles, CA 90095. This work was supported by the National Science Foundation grant DMS-1055286 to Q.Z. The authors thank the editor, the associate editor, and two referees for helpful comments and suggestions, which significantly improved the article.

Journal of the American Statistical Association
March 2013, Vol. 108, No. 501, Theory and Methods
DOI: 10.1080/01621459.2012.754359

interventional data. Different interventions effectively change the structure of a DAG as shown in Section 2. Second, it is expected that the computation for estimating the structure of DAGs is much more challenging than that for undirected graphs because of the acyclicity constraint. Indeed, the recent work of Shojaie and Michailidis (2010) assumed a known ordering of the variables to simplify the computation for the structure learning problem of DAGs, which eliminates the need for estimating the directions of causality among random variables.

In this article, we develop an L_1-penalized likelihood approach to structure learning of causal Gaussian Bayesian networks (GBNs) using experimental data. We consider this problem in the general setting where the ordering of the variables is unknown. To the best of our knowledge, this is the first method that estimates the structure of DAGs based on L_1-penalized likelihood without assuming a known ordering. In Section 2, we formulate the problem of learning causal DAGs with experimental data. We develop a CD algorithm in Section 3 to search for a locally optimal solution to this optimization problem and establish in Section 4 theoretical properties of the corresponding estimator. In Section 5 we present results of a simulation study, and in Section 6, we apply our method to a real dataset. The article is concluded with discussion in Section 7. All proofs are provided in the Appendix or the online supplementary materials.

2. PROBLEM FORMULATION

2.1 Causal Bayesian Networks

The joint probability distribution of a set of random variables $X_1, \ldots, X_p$ in a BN can be factorized as

$$P(X_1, \ldots, X_p) = \prod_{i=1}^{p} P\left(X_i | \Pi_i^{\mathcal{G}}\right), \tag{1}$$

where $\Pi_i^{\mathcal{G}} \subseteq \{X_1, \ldots, X_p\} \backslash \{X_i\}$ is called the set of parents of X_i. If X_i does not have any parents, then $\Pi_i^{\mathcal{G}} = \varnothing$. We can construct a DAG $\mathcal{G} = (V, E)$ to represent the structure of a BN. Here, $V = \{1, \ldots, p\}$ denotes the set of nodes in the graph, where the ith node in V corresponds to X_i. For simplicity, we use X_i and i interchangeably throughout the article to represent the ith node. The set of edges $E = \{(i, j) : X_i \in \Pi_j^{\mathcal{G}}\}$ and an edge $(i, j) \in E$ is written as $i \to j$. The structure of $\mathcal{G}$ must be acyclic so that (1) is a well-defined joint distribution. For any DAG $\mathcal{G}$, there exists at least one ordering of the nodes, known as a topological sort of $\mathcal{G}$, such that $i \prec j$ if $i \in \Pi_j^{\mathcal{G}}$. A more convenient representation of the structure of a DAG is the adjacency matrix, a $p \times p$ matrix $\mathbf{A}$ whose (i, j)th entry is 1 if $i \to j$ and 0 otherwise. Estimating the structure of DAGs from data is equivalent to estimating their adjacency matrices.

For some joint distributions, there exist multiple factorizations of the form in (1). Those DAGs that encode the same set of joint distributions form an equivalence class. We cannot distinguish equivalent DAGs from observational data. However, equivalent DAGs do not have the same causal interpretation. In this article, we only consider using DAGs for causal inference, following Pearl's formulation of causal BNs (Pearl 2000). In this setting, experimental interventions can help us distinguish equivalent DAGs. For instance, consider the causal interpretations of two equivalent DAGs $\mathcal{G}_1$: $X_1 \to X_2$ and $\mathcal{G}_2$: $X_1 \leftarrow X_2$. Suppose that X_2 is fixed experimentally at x_2 (the fixed value itself might be drawn from some distribution independent of the DAG). If $\mathcal{G}_1$ is the true causal model, fixing X_2 eliminates any dependency of X_2 on X_1, in effect removing the directed edge from X_1 to X_2. Thus, data generated in this manner follow the joint distribution $P(X_1, X_2) = P(X_1|\varnothing)P(X_2|\bullet)$, where $P(X_1|\varnothing)$ is the marginal distribution of X_1 and $P(X_2|\bullet)$ is the distribution from which experimental data on X_2 are drawn. On the other hand, if the true causal model is $\mathcal{G}_2$, interventions on X_2 leave the dependency between X_1 and X_2 intact. Hence, experimental data can be used to infer causal relationships among random variables. As this example demonstrates, if X_i ($i \in \mathcal{M}$) are under intervention, then the joint distribution in (1) becomes

$$P(X_1, \ldots, X_p) = \prod_{i \notin \mathcal{M}} P\left(X_i | \Pi_i^{\mathcal{G}}\right) \prod_{i \in \mathcal{M}} P(X_i|\bullet), \tag{2}$$

where $P(X_i|\bullet)$ denotes the distribution of X_i under intervention. In other words, we can view experimental data from $\mathcal{G}$ as being generated from the DAG $\mathcal{G}'$ obtained by removing all directed edges pointing to the nodes under intervention in $\mathcal{G}$. When we make causal inference using the likelihood function (2), the term $\prod_{i \in \mathcal{M}} P(X_i|\bullet)$ can be ignored, since they depend only on external parameters that are not relevant to the estimation of DAGs.

2.2 L_1-Regularized Log-Likelihood

In a causal GBN $\mathcal{G}$, causal relationships among random variables are modeled as

$$X_j = \sum_{i \in \Pi_j^{\mathcal{G}}} \beta_{ij} X_i + \varepsilon_j, \quad j = 1, \ldots, p, \tag{3}$$

where β_{ij} is the coefficient representing the influence of X_i on X_j, and ε_j's are independent Gaussian noise variables with mean 0 and variance σ_j^2. We assume, throughout this article, that all X_j have mean 0. Then the joint distribution of $(X_1, \ldots, X_p)$ defined by (3) is $\mathcal{N}_p(\mathbf{0}, \boldsymbol{\Sigma})$, where the covariance matrix $\boldsymbol{\Sigma}$ depends on β_{ij} ($i, j = 1, \ldots, p$ and $i \neq j$) and σ_j^2 ($j = 1, \ldots, p$). The set of equations in (3) can be regarded as the mechanism for generating these random variables.

Consider an $n \times p$ data matrix $\mathbf{X}$ generated from $\mathcal{G}$. The data matrix $\mathbf{X}$ consists of p blocks with the jth block $\mathbf{X}^j$ having size $n_j \times p$, where $n = \sum_{j=1}^{p} n_j$. Each row in $\mathbf{X}^j$ is generated by imposing an intervention on the node X_j, while the values for all other nodes X_k ($k \neq j$) are observational. The experimental data on X_j generated by intervention are assumed to follow $\mathcal{N}(0, \widetilde{\sigma}_j^2)$ for $j = 1, \ldots, p$.

Let $\mathbf{B} = (\beta_{ij})_{p \times p}$ be the coefficient matrix, where $\beta_{ij} = 0$ if $i \notin \Pi_j^{\mathcal{G}}$. Let $\boldsymbol{\sigma}^2 = (\sigma_j^2)_{1 \times p}$ and $\widetilde{\boldsymbol{\sigma}}^2 = (\widetilde{\sigma}_j^2)_{1 \times p}$ be vectors of variances. Apparently, we can learn the structure of $\mathcal{G}$ by estimating the coefficient matrix $\mathbf{B}$. In the rest of the article, we will call $\mathcal{G}$ the graph induced by $\mathbf{B}$.

Let $\mathcal{I}_j$ denote the collection of the row indices of $\mathbf{X}^j$, and then $\mathcal{O}_j = \{1, \ldots, n\} \backslash \mathcal{I}_j$ gives the collection of data points in which X_j is not fixed experimentally, $j = 1, \ldots, p$. According to the factorization (2), the likelihood of the data matrix $\mathbf{X}$ can

be written as

$$f(\mathbf{X}) \propto \prod_{k=1}^{p} \prod_{h\in\mathcal{I}_k} \prod_{j\neq k} f(x_{hj}|\pi_{hj}) = \prod_{j=1}^{p} \prod_{h\in\mathcal{O}_j} f(x_{hj}|\pi_{hj}), \quad (4)$$

where x_{hj} is the value of X_j in the hth data point [the (h, j)th element of the data matrix $\mathbf{X}$], π_{hj} is the value of its parents, and $f(x_{hj}|\pi_{hj})$ is the conditional density of x_{hj} given π_{hj}. Note that, as mentioned in Section 2.1, the likelihood term $f(x_{hj}|\bullet)$ is ignored if the value x_{hj} is fixed experimentally. Let $n_{-j} = |\mathcal{O}_j| = n - n_j$. Using the relationship in (3), we can easily derive that the negative log-likelihood of $\mathbf{B}$ and $\boldsymbol{\sigma}^2$ is

$$\sum_{j=1}^{p} \left[\frac{n_{-j} \log\left(\sigma_j^2\right)}{2} + \frac{\left\|\mathbf{X}_{[\mathcal{O}_j,j]} - \mathbf{X}_{[\mathcal{O}_j,-j]}\mathbf{B}_{[-j,j]}\right\|^2}{2\sigma_j^2} \right], \quad (5)$$

where $\mathbf{M}_{[I_r,I_c]}$ denotes the submatrix of $\mathbf{M}$ with rows in I_r and columns in I_c.

For many applications, it is often the case that the underlying network structure is sparse. It is therefore important to find a sparse structure for the coefficient matrix $\mathbf{B}$. We propose here a penalized likelihood approach with the adaptive lasso penalty to learn the structure of $\mathbf{B}$. Specifically, given a weight matrix $\mathbf{W} = (w_{ij})_{p\times p}$, we seek the minimizer $(\widehat{\mathbf{B}}, \widehat{\boldsymbol{\sigma}}^2)$ of

$$\sum_{j=1}^{p} \left[\frac{n_{-j} \log\left(\sigma_j^2\right)}{2} + \frac{\left\|\mathbf{X}_{[\mathcal{O}_j,j]} - \mathbf{X}_{[\mathcal{O}_j,-j]}\mathbf{B}_{[-j,j]}\right\|^2}{2\sigma_j^2} + \lambda \sum_{i\neq j} w_{ij}|\beta_{ij}| \right], \quad \text{subject to} \quad \mathcal{G}_{\mathbf{B}} \text{ is acyclic}, \quad (6)$$

where $\mathcal{G}_{\mathbf{B}}$ denotes the graph induced by $\mathbf{B}$ and $\lambda > 0$ is the penalty parameter.

Remark 1. Due to the acyclicity constraint, one cannot transform (6) into an equivalent penalized least squares problem. Moreover, σ_j^2 cannot be ignored in our formulation, which makes the minimization problem considerably harder than a penalized least squares problem.

The adaptive lasso was proposed by Zou (2006) as an alternative to the lasso technique (Tibshirani 1996) for regression problems. The adaptive lasso enjoys the oracle properties considered by Fan and Li (2001). In particular, it is consistent for variable selection. In our setting, the weights are defined as $w_{ij} = |\hat{\beta}_{ij}^{(\dagger)}|^{-\gamma}$ for some $\gamma > 0$, where $\hat{\beta}_{ij}^{(\dagger)}$ is a consistent estimate of β_{ij}. Zou (2006) suggested using the ordinary least squares (OLS) estimates to define the weights in the regression setting. However, because of the existence of equivalent DAGs, the OLS estimates may not be consistent in our case. To obtain the initial consistent estimates $\hat{\beta}_{ij}^{(\dagger)}$'s, we let $\widetilde{w}_{ij} = \min(|\hat{\beta}_{ij}^{(\mathrm{OLS})}|^{-\gamma}, M^{\gamma})$, where M is a large positive constant (e.g., $M = 10^4$) and $\hat{\beta}_{ij}^{(\mathrm{OLS})}$'s are the OLS estimates obtained by regressing X_j on other nodes using data in $\mathcal{O}_j$. As will be shown in Section 4, there exists a $\sqrt{n}$-consistent local minimizer $\widehat{\mathbf{B}}$ of (6) with the weights $\widetilde{w}_{ij}$, which can be used as $\hat{\beta}_{ij}^{(\dagger)}$'s. Then, a consistent estimate of the graph structure can be obtained with weights $w_{ij} = |\hat{\beta}_{ij}^{(\dagger)}|^{-\gamma}$.

After minimizing with respect to $\boldsymbol{\sigma}^2$, problem (6) becomes

$$\min_{\mathbf{B}} V(\mathbf{B};\mathbf{W}) = \sum_{j=1}^{p} \left[\frac{n_{-j}}{2} \log\left(\left\|\mathbf{X}_{[\mathcal{O}_j,j]} - \mathbf{X}_{[\mathcal{O}_j,-j]}\mathbf{B}_{[-j,j]}\right\|^2\right) + \lambda \sum_{i\neq j} w_{ij}|\beta_{ij}| \right], \quad \text{subject to} \quad \mathcal{G}_{\mathbf{B}} \text{ is acyclic}, \quad (7)$$

which is the problem we aim to solve.

3. COORDINATE DESCENT ALGORITHM

Both the objective function V in (7) and the constraint set are nonconvex. Searching for the global minimizer of (7) may be impractical. Moreover, the theoretical results in Section 4 only establish consistency for a local minimizer (see Theorems 2 and 3). Therefore, we develop in this section a CD algorithm to find a local minimum to the constrained optimization problem (7). A local minimizer $\widehat{\mathbf{B}}$ is defined as follows: (i) any local change in the structure of $\mathcal{G}_{\widehat{\mathbf{B}}}$, that is, addition, removal, or reversal of a single edge, increases the value of V and (ii) given the structure of $\mathcal{G}_{\widehat{\mathbf{B}}}$, $\widehat{\mathbf{B}}$ is a local minimizer of V. CD methods have been successfully used to solve lasso-type problems (Fu 1998; Friedman et al. 2007; Wu and Lange 2008). They are attractive since minimizing the objective function one coordinate at a time is computationally simple and gradient free. As a result, these methods are easy to implement and are usually scalable.

3.1 Single-Parameter Update

Before detailing the CD algorithm, let us first consider minimizing V in (7) with respect to a single parameter β_{kj} $(k \neq j)$ without the acyclicity constraint. In particular, we seek the minimizer $\hat{\beta}_{kj}$ of

$$\begin{aligned} V_j &= \frac{n_{-j}}{2} \log\left(\left\|\mathbf{X}_{[\mathcal{O}_j,j]} - \mathbf{X}_{[\mathcal{O}_j,-j]}\mathbf{B}_{[-j,j]}\right\|^2\right) + \lambda \sum_{i\neq j} w_{ij}|\beta_{ij}| \\ &= \frac{n_{-j}}{2} \log\left[\sum_{h\in\mathcal{O}_j} \left(x_{hj} - \sum_{i\neq j,k} x_{hi}\beta_{ij} - x_{hk}\beta_{kj} \right)^2 \right] \\ &\quad + \lambda \sum_{i\neq j,k} w_{ij}|\beta_{ij}| + \lambda w_{kj}|\beta_{kj}|, \end{aligned} \quad (8)$$

assuming all β_{ij}'s $(i \neq j, k)$ are fixed. We can transform the weighted lasso penalty in (8) into an ordinary lasso penalty:

$$\begin{aligned} \min_{\tilde{\beta}_{kj}} \widetilde{V}_j = \frac{n_{-j}}{2} \log&\left[\sum_{h\in\mathcal{O}_j} \left(x_{hj} - \sum_{i\neq j,k} \tilde{x}_{hi}\tilde{\beta}_{ij} - \tilde{x}_{hk}\tilde{\beta}_{kj} \right)^2 \right] \\ &+ \lambda \sum_{i\neq j,k} |\tilde{\beta}_{ij}| + \lambda|\tilde{\beta}_{kj}|, \end{aligned} \quad (9)$$

by letting $\tilde{\beta}_{ij} = w_{ij}\beta_{ij}$ and $\tilde{x}_{hi} = x_{hi}/w_{ij}$ for $i \neq j$. We further define $y_{hj}^{(k)} = x_{hj} - \sum_{i\neq j,k} \tilde{x}_{hi}\tilde{\beta}_{ij}$, $\xi_{kj} = \sum_{h\in\mathcal{O}_j} \tilde{x}_{hk} y_{hj}^{(k)} / \sum_{h\in\mathcal{O}_j} \tilde{x}_{hk}^2$, $c_{kj} = \sum_{h\in\mathcal{O}_j} (y_{hj}^{(k)})^2 / \sum_{h\in\mathcal{O}_j} \tilde{x}_{hk}^2$, and $\eta = \lambda/n_{-j}$. Note that according to Cauchy–Schwarz inequality, $c_{kj} - \xi_{kj}^2$

≥ 0. Then equivalently, (9) can be simplified to the problem

$$\min_{\tilde{\beta}_{kj}} g(\tilde{\beta}_{kj}) = \frac{1}{2}\log\left[(\tilde{\beta}_{kj} - \xi_{kj})^2 + \left(c_{kj} - \xi_{kj}^2\right)\right] + \eta|\tilde{\beta}_{kj}|. \tag{10}$$

The form of g is reminiscent of the lasso problem with a single predictor. However, minimizing g with respect to $\tilde{\beta}_{kj}$ is not as easy as the corresponding lasso problem, since g is not a convex function and might have two local minima for some values of ξ_{kj}, c_{kj}, and η. It is therefore necessary to compare the two local minima under certain conditions. We summarize the solution to (10) in the following proposition and provide its proof in the online supplementary materials.

Proposition 1. Let $\Delta = 1 - 4(c_{kj} - \xi_{kj}^2)\eta^2$ and $\beta_1^* = \operatorname{sgn}(\xi_{kj})(|\xi_{kj}| - \frac{1-\sqrt{\Delta}}{2\eta})$. The solution to the optimization problem (10) is given by

$$\arg\min_{\tilde{\beta}_{kj}} g = \begin{cases} \beta_1^*, & \text{if } 0 < \eta < |\xi_{kj}|/c_{kj}, \\ \beta_1^*, & \text{if } |\xi_{kj}|/c_{kj} \leq \eta < \left(2\sqrt{c_{kj} - \xi_{kj}^2}\right)^{-1}, \\ & \quad \eta > (2|\xi_{kj}|)^{-1} \text{ and } g(\beta_1^*) < g(0), \\ 0, & \text{otherwise.} \end{cases}$$

Remark 2. The form of β_1^* suggests that $\arg\min_{\tilde{\beta}_{kj}} g$ is similar to a soft thresholded version (Donoho and Johnstone 1995) of ξ_{kj} in nature. One difference, however, is that $\arg\min_{\tilde{\beta}_{kj}} g$ can be zero even when $|\xi_{kj}| - (1 - \sqrt{\Delta})(2\eta)^{-1} > 0$ (see the proof of Proposition 1 in the online supplementary materials). Note that if $4(c_{kj} - \xi_{kj}^2)\eta^2 = o(1)$, by Taylor expansion $\sqrt{\Delta} \approx 1 - 2(c_{kj} - \xi_{kj}^2)\eta^2$. Then $\beta_1^* \approx \operatorname{sgn}(\xi_{kj})(|\xi_{kj}| - (c_{kj} - \xi_{kj}^2)\eta) = \operatorname{sgn}(\xi_{kj})(|\xi_{kj}| - c_{kj}(1 - \zeta^2)\eta)$, where ζ is the correlation coefficient between $\tilde{x}_{hk}$ and $y_{hj}^{(k)}$ for $h \in \mathcal{O}_j$.

Remark 3. In Proposition 1, we could find a more explicit condition on η to determine when $g(\beta_1^*) < g(0)$, but the condition does not have a closed-form expression. Thus, it seems more effective to compare $g(\beta_1^*)$ and $g(0)$ directly.

3.2 Description of the CD Algorithm

The difficulty in minimizing V in (7) is due to the constraint that the graphical representation of BNs is acyclic. One immediate consequence of this constraint is that a pair of coefficients β_{ij} and β_{ji} cannot both be nonzero. We thus take advantage of this implication when designing the CD algorithm. Instead of minimizing V over a single parameter β_{ij} at each step, we perform minimization over β_{ij} and β_{ji} simultaneously. Hence, our method can be naturally described as a blockwise CD method. For a p-node problem, the $p(p-1)$ coefficients are partitioned into $p(p-1)/2$ blocks. Each block consists of a pair of coefficients β_{ij} and β_{ji}. The algorithm starts with an initial estimate of the coefficient matrix $\mathbf{B}$ (for instance, the zero matrix) and assumes a predefined order to cycle through the $p(p-1)/2$ blocks. At each step, V is minimized over a certain block of β_{ij} and β_{ji} while all other blocks are held constant. Given the current estimates of other blocks, β_{ij} (or β_{ji}) is constrained to zero if a nonzero value introduces cycles in the resulting graph. In this case, V is only minimized over β_{ji} (or β_{ij}). Otherwise, the algorithm compares $\min_{\beta_{ij},\beta_{ji}=0} V$ with $\min_{\beta_{ij}=0,\beta_{ji}} V$ to update β_{ij} and β_{ji}. We repeat cycling through the $p(p-1)/2$ blocks until some stopping criterion is satisfied.

The major steps in the CD algorithm are summarized as follows, where we use $\tilde{\beta}_{ij} \Leftarrow 0$ to mean that $\tilde{\beta}_{ij}$ must be set to zero due to the acyclicity constraint. In the following, different $\mathbf{X}_{[\mathcal{O}_j,\cdot]}$'s are treated as different entities so that operations on $\mathbf{X}_{[\mathcal{O}_j,\cdot]}$ will not affect $\mathbf{X}_{[\mathcal{O}_k,\cdot]}$ for $k \neq j$.

Algorithm 1. CD Algorithm for Estimating DAGs

1. Center and standardize the columns of $\mathbf{X}_{[\mathcal{O}_j,\cdot]}(j = 1, \ldots, p)$ to have mean zero and unit L_2 norm. Transform the weighted lasso problem (7) to an ordinary lasso problem by defining $\widetilde{\mathbf{X}}_{[\mathcal{O}_j,i]} = \mathbf{X}_{[\mathcal{O}_j,i]}/w_{ij}$, $i \neq j$, for $j = 1, \ldots, p$. Choose $\mathbf{B}^0$ such that $\mathcal{G}_{\mathbf{B}^0}$ is acyclic.
2. Cycle through the $p(p-1)/2$ blocks of coefficients. Specifically, do one of the following for the pair of coefficients $\tilde{\beta}_{ij}$ and $\tilde{\beta}_{ji}$ $(i < j)$, given the current estimates of other coefficients.
   (a) If $\tilde{\beta}_{ji} \Leftarrow 0$, minimize $\widetilde{V}_j$ in (9) with respect to $\tilde{\beta}_{ij}$ according to Proposition 1 and find $\tilde{\beta}_{ij}^* = \arg\min_{\tilde{\beta}_{ij}} \widetilde{V}_j$. Then set $(\tilde{\beta}_{ij}, \tilde{\beta}_{ji}) = (\tilde{\beta}_{ij}^*, 0)$.
   (b) If $\tilde{\beta}_{ij} \Leftarrow 0$, minimize $\widetilde{V}_i$ with respect to $\tilde{\beta}_{ji}$ according to Proposition 1 and find $\tilde{\beta}_{ji}^* = \arg\min_{\tilde{\beta}_{ji}} \widetilde{V}_i$. Then set $(\tilde{\beta}_{ij}, \tilde{\beta}_{ji}) = (0, \tilde{\beta}_{ji}^*)$.
   (c) If 2(a) and 2(b) do not apply, then compare the following two sums: $S_1 = \widetilde{V}_i|_{\tilde{\beta}_{ji}=0} + \widetilde{V}_j|_{\tilde{\beta}_{ij}=\tilde{\beta}_{ij}^*}$ and $S_2 = \widetilde{V}_i|_{\tilde{\beta}_{ji}=\tilde{\beta}_{ji}^*} + \widetilde{V}_j|_{\tilde{\beta}_{ij}=0}$. Set $(\tilde{\beta}_{ij}, \tilde{\beta}_{ji}) = (\tilde{\beta}_{ij}^*, 0)$ if $S_1 \leq S_2$. Otherwise, set $(\tilde{\beta}_{ij}, \tilde{\beta}_{ji}) = (0, \tilde{\beta}_{ji}^*)$.
3. Repeat Step 2 until the maximum absolute difference among all coefficients between successive cycles is below some threshold or until the maximum number of iterations is reached.
4. Output the estimates $\hat{\beta}_{ij} = \tilde{\beta}_{ij}/w_{ij}$ for $i, j = 1, \ldots, p$ and $i \neq j$.

To ensure the acyclicity constraint by checking whether $\tilde{\beta}_{ij} \Leftarrow 0$, we employ a breadth-first search algorithm based on algorithm 4 in Ellis (2006). A detailed description of this algorithm is given in the online supplementary materials.

3.3 Practical Considerations

Since it is difficult in practice to predetermine the optimal value of λ, we compute solutions using a decreasing sequence of values for λ, following the practice in Friedman, Hastie, and Tibshirani (2010). The solution for the current λ is used as the initial estimate for the next value of λ in the sequence. Since large values of λ force many β_{ij} to be zero and make the optimization much easier, the solution for large λ is likely to agree well with some sub-graph of the true model. Therefore, employing warm starts may boost the performance of the CD algorithm.

To speed up the CD algorithm, we use an active set method that is better suited for warm starts, as was done by Friedman, Hastie, and Tibshirani (2010). The algorithm first performs a complete cycling through all $p(p-1)/2$ blocks of coefficients

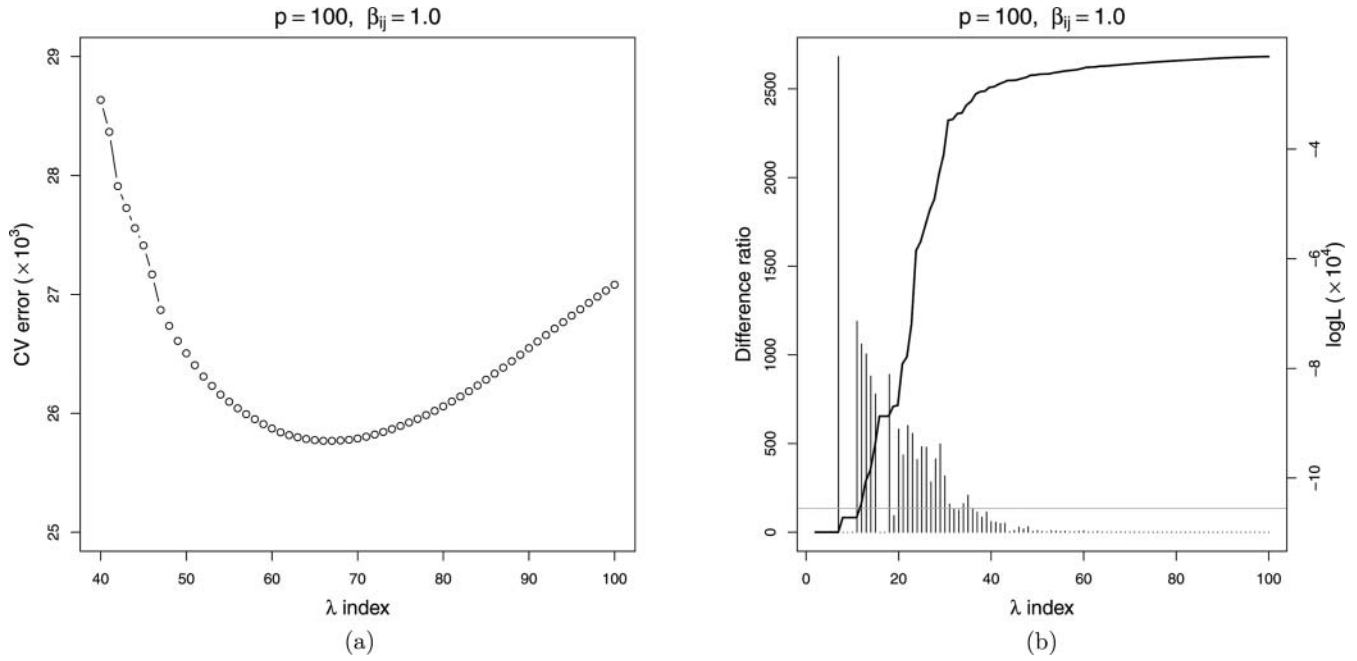

Figure 1. Plots of (a) CV error, (b) difference ratio ("histogram-like" vertical lines), and log-likelihood (solid line) for graphs estimated using a decreasing sequence of λ.

to identify the active set—the set of blocks with a nonzero coefficient. We then only iterate over the active set until the maximum coefficient difference falls below the threshold or the maximum number of iterations has been reached. The algorithm stops if another full cycle through all the blocks does not change the active set; otherwise the above process is repeated. Note that when the active set changes, the skeleton of the estimated DAG is updated. Furthermore, when the algorithm iterates over a given active set of blocks, the edges may still be reversed.

It should be noted that convergence of CD methods often requires the objective function to be strictly convex and differentiable. For nondifferentiable functions, CD may get stuck at nonoptimal points, although Tseng (2001) considered generalizations to nondifferentiable functions with certain separability and regularity properties. Because of the nonconvex nature of the objective function V in (7) and the constraint set, convergence of the CD algorithm deserves a rigorous investigation, which is beyond the scope of this study. We conjecture that the CD algorithm converges under certain conditions. In practice, we have never encountered any examples so far where the algorithm does not converge. For a demonstration of convergence of our algorithm, see Figure S2 in the online supplementary materials.

3.4 Choice of the Tuning Parameter

The graphical model learned by the CD algorithm depends on the choice of the penalty λ. Model selection is usually based on an estimated prediction error, and commonly used model selection methods include the Bayesian information criterion (BIC) and cross-validation (CV) among others. As established by Meinshausen and Bühlmann (2006) for L_1-penalized linear regression, a model selected based on minimizing the prediction error is often too complex compared to the true model. Figure 1(a) plots the five-fold CV error for a sequence of graphs learned given a decreasing sequence of λ from a simulated dataset with $p = 100$, $n = 500$, and $\beta_{ij} = 1.0$. The CV error is minimized at the 67th λ. The corresponding graph $\hat{\mathcal{G}}_{67}$ (obtained using λ_{67} as the tuning parameter on the whole dataset) has a total of 993 predicted edges with an 82.6% false discovery rate, while the true graph only has 200 directed edges. Similar results are obtained if we use BIC or other scoring metrics such as the Bayesian score of a graph.

In this article, we employ an empirical model selection criterion that works well in practice. Note that as we decrease λ and thus increase model complexity, the log-likelihood of the estimated graph will increase. Denote by $\widehat{\mathbf{B}}_{\lambda_i}$ the solution to (7) with the ith penalty parameter λ_i. Given the estimated graph $\hat{\mathcal{G}}_{\lambda_i}$ induced by $\widehat{\mathbf{B}}_{\lambda_i}$, we estimate the unpenalized coefficient matrix, denoted by $\widetilde{\mathbf{B}}_i$, by regressing X_k on $\Pi_k^{\hat{\mathcal{G}}_{\lambda_i}}$, $k = 1, \ldots, p$. Given two estimated graphs $\hat{\mathcal{G}}_{\lambda_i}$ and $\hat{\mathcal{G}}_{\lambda_j}$ ($\lambda_i > \lambda_j$), let $\Delta L_{ij} = L(\widetilde{\mathbf{B}}_j) - L(\widetilde{\mathbf{B}}_i)$ and $\Delta e_{ij} = e_{\lambda_j} - e_{\lambda_i}$, where $L(\widetilde{\mathbf{B}}) = -V(\widetilde{\mathbf{B}}; \mathbf{0})$ denotes the log-likelihood function and e denotes the total number of edges in an estimated graph. We then define the difference ratio between the two estimated graphs as $dr_{(ij)} = \Delta L_{ij}/\Delta e_{ij}$. We reason that an increase in model complexity, which is represented by an increase in the total number of predicted edges, is desirable only if there is a substantial increase in the log-likelihood. Therefore, we compute successively the difference ratios between two adjacent graphs in the solution path, $\{dr_{(12)}, \ldots, dr_{(m-1,m)}\}$, where m is the number of λ in the sequence. The graph with the following index is selected:

$$K = \sup\{k : dr_{(k-1,k)} \geq \alpha \times \max(dr_{(12)}, \ldots, dr_{(m-1,m)}), \quad k = 2, \ldots, m\}, \tag{11}$$

where α is a thresholding parameter. Essentially, this is the graph from which further increase in model complexity will not lead to substantial increase in the likelihood. We find that $\alpha \in [0.05, 0.1]$ works well in our simulation. Figure 1(b) plots

the difference ratio as well as the log-likelihood for different graphs learned from the same dataset. The graph selected according to (11) with $\alpha = 0.05$ is $\hat{\mathcal{G}}_{36}$, which has 168 edges with a 77% true positive rate and an 8.3% false discovery rate, much less than 82.6%.

4. ASYMPTOTIC PROPERTIES

In this section, we develop asymptotic theories on the penalized likelihood estimator of DAGs. To simplify notations, we write $\mathbf{B}$ in a vector format as $\boldsymbol{\phi} = (\phi_j)_{1\times d} = ((\mathbf{B}_{[-1,1]})^T, \ldots, (\mathbf{B}_{[-p,p]})^T)$, where $d = p(p-1)$ is the length of $\boldsymbol{\phi}$. Similarly, we write the weight matrix $\mathbf{W}$ in a vector format as $\mathcal{T} = (\tau_j)_{1\times d}$. We say that $\boldsymbol{\phi}$ is acyclic if the graph $\mathcal{G}_{\boldsymbol{\phi}}$ induced by $\boldsymbol{\phi}$ (or the corresponding $\mathbf{B}$) is acyclic. Let $\boldsymbol{\theta} = (\boldsymbol{\phi}, \boldsymbol{\sigma}^2, \widetilde{\boldsymbol{\sigma}}^2)$ be the vector of parameters and $\boldsymbol{\Omega} = \{\boldsymbol{\theta} : \boldsymbol{\phi}$ is acyclic, $\boldsymbol{\sigma}^2 > 0, \widetilde{\boldsymbol{\sigma}}^2 > 0\}$ be the parameter space. Recall that $\boldsymbol{\sigma}^2 = (\sigma_j^2)_{1\times p}$ and $\widetilde{\boldsymbol{\sigma}}^2 = (\widetilde{\sigma}_j^2)_{1\times p}$ are vectors of variances defined in Section 2.2. Denote the true parameter value by $\boldsymbol{\theta}^* = (\boldsymbol{\phi}^*, (\boldsymbol{\sigma}^2)^*, (\widetilde{\boldsymbol{\sigma}}^2)^*) \in \boldsymbol{\Omega}$. Let $\mathcal{G}_{\boldsymbol{\phi}^*}$ denote the DAG induced by $\boldsymbol{\phi}^*$, that is, the true DAG.

Let $\boldsymbol{\theta}_k = (\boldsymbol{\phi}_k, \boldsymbol{\sigma}^2_{[-k]}, \widetilde{\sigma}^2_k)$, where $\boldsymbol{\phi}_k$ is obtained from $\boldsymbol{\phi}$ by replacing $\mathbf{B}_{[-k,k]}$ with $\mathbf{0}$, that is, by suppressing all edges pointing to the kth node from its parents. Here $\boldsymbol{\nu}_{[I]}$ denotes the subvector of a vector $\boldsymbol{\nu}$ with components in I. As mentioned in Section 2.1, $\mathbf{X}^k$, the kth block of the data matrix, can be regarded as independent and identically distributed (iid) observations from a distribution factorized according to the DAG $\mathcal{G}_{\boldsymbol{\phi}_k}$, and we denote the corresponding density by $f(\mathbf{x}|\boldsymbol{\theta}_k)$, where $\mathbf{x} = (x_1, \ldots, x_p)$. For Gaussian random variables, f is the density function of $\mathcal{N}_p(\mathbf{0}, \boldsymbol{\Sigma}(\boldsymbol{\theta}_k))$. Here we emphasize the dependence of the variance-covariance matrix $\boldsymbol{\Sigma}$ on $\boldsymbol{\theta}_k$. Recall that $\mathcal{I}_k$ denotes the collection of the row indices of $\mathbf{X}^k$. Then we define the penalized log-likelihood with the adaptive lasso penalty as

$$R(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) - \lambda_n \sum_{j=1}^{d} \tau_j |\phi_j| = \sum_{k=1}^{p} L_k(\boldsymbol{\theta}_k) - \lambda_n \sum_{j=1}^{d} \tau_j |\phi_j|, \tag{12}$$

where $L_k(\boldsymbol{\theta}_k) = \sum_{h\in\mathcal{I}_k} \log f(\mathbf{X}_{[h,\cdot]}|\boldsymbol{\theta}_k)$. Our goal is to seek a local maximizer of $R(\boldsymbol{\theta})$ in the parameter space $\boldsymbol{\Omega}$ to obtain an estimator $\hat{\boldsymbol{\theta}}$. Note that the log-likelihood function $L(\boldsymbol{\theta})$ is different from the one in (6) and (7), since here we also include in $L(\boldsymbol{\theta})$ terms depending on $\widetilde{\boldsymbol{\sigma}}^2$. It is easily seen that these two formulations of the likelihood function are equivalent for the purpose of estimating the coefficients and the structure of BNs.

Even with interventional data, the coefficient matrix of a DAG may not be identifiable because of interventional Markov equivalence among DAGs (Hauser and Bühlmann 2012). We introduce below the notion of natural parameters to establish identifiability of DAGs for the case where each variable has interventional data. Suppose that X_i is an ancestor of X_j in a DAG $\mathcal{G}$, that is, there exists at least one path from X_i to X_j (see Lauritzen 1996, chap. 2, for terminology used in graphical models). Let

$$\Gamma(i,j) = \{(i_0, \ldots, i_m) : i_k \to i_{k+1} \text{ for } 0 \le k \le m-1, \\ i_0 = i, i_m = j, m \ge 1\} \tag{13}$$

be the set of paths from X_i to X_j, and define the coefficient of influence of X_i on X_j by $\beta_{i\to j} = \sum_{\Gamma(i,j)} \prod_{k=0}^{m-1} \beta_{i_k i_{k+1}}$.

Denote the set of ancestors of X_j by $\text{an}(X_j)$.

Definition 1 (Natural parameters). We say that $\boldsymbol{\theta}$ is natural if the corresponding coefficient matrix $\mathbf{B}$ satisfies

$$\beta_{i\to j} \ne 0 \quad \text{for all } X_i \in \text{an}(X_j), \quad 1 \le j \le p. \tag{14}$$

Note that if the underlying DAG is a polytree, the corresponding parameter is always natural. For more general DAGs, a natural parameter implies that the causal effects along multiple paths connecting the same pair of nodes do not cancel, which is a reasonable assumption for many real-world problems. If the true parameter is natural, then with sufficient experimental data, the parameter $\boldsymbol{\theta}$ is identifiable as indicated by the following theorem. The proof of Theorem 1 is given in the Appendix.

Theorem 1. Suppose that $\mathbf{X}^k$ is an iid sample from the normal distribution $\mathcal{N}_p(\mathbf{0}, \boldsymbol{\Sigma}(\boldsymbol{\theta}_k^*))$ with density $f(\mathbf{x}|\boldsymbol{\theta}_k^*)$ for $k = 1, \ldots, p$. Assume that the true parameter $\boldsymbol{\theta}^*$ is natural. Then

$$f(\mathbf{x}|\boldsymbol{\theta}_k) = f(\mathbf{x}|\boldsymbol{\theta}_k^*) \quad \text{almost everywhere (a.e.)} \\ \text{for all } k = 1, \ldots, p \qquad \Longrightarrow \qquad \boldsymbol{\theta} = \boldsymbol{\theta}^*. \tag{15}$$

If we further assume that $n_k/n \to \alpha_k > 0$ as $n \to \infty$, then

$$P_{\boldsymbol{\theta}^*}(L(\boldsymbol{\theta}^*) > L(\boldsymbol{\theta})) \to 1 \tag{16}$$

for any $\boldsymbol{\theta} \ne \boldsymbol{\theta}^*$.

Now we state two theorems to establish the asymptotic properties of $\hat{\boldsymbol{\theta}}$. We follow arguments similar to those given by Fan, Feng, and Wu (2009) to prove Theorems 2 and 3. However, one cannot directly apply Fan et al.'s results here, because the parameters must satisfy the acyclicity constraint, the data we have are not iid observations due to interventions, and the identifiability of a DAG is not always guaranteed.

Let $\hat{\phi}_k^{(\text{OLS})}$ $(1 \le k \le d)$ be the estimate of ϕ_k when the corresponding β_{ij} $(i \ne j)$ is estimated by $\hat{\beta}_{ij}^{(\text{OLS})}$. Let $\mathcal{A} = \{j : \phi_j^* = 0\}$ and $\boldsymbol{\phi}_{\mathcal{A}} = (\phi_j)_{j\in\mathcal{A}}$. It is assumed that $\boldsymbol{\theta}^*$ is natural in the following two theorems. We relegate the proof of Theorem 2 to the Appendix. The proof of Theorem 3 is given in the online supplementary materials.

Theorem 2. Assume the adaptive lasso penalty with weights $\tau_j = \min(|\hat{\phi}_j^{(\text{OLS})}|^{-\gamma}, M^\gamma)$ for all j, where $\gamma, M > 0$. As $n \to \infty$, if $\lambda_n/\sqrt{n} \to 0$ and $n_k/n \to \alpha_k > 0$ for $k = 1, \ldots, p$, then there exists a local maximizer $\hat{\boldsymbol{\theta}}$ of $R(\boldsymbol{\theta})$ such that $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\| = O_p(n^{-1/2})$.

Theorem 3. Assume the adaptive lasso penalty with weights $\tau_j = |\tilde{\phi}_j|^{-\gamma}$ for some $\gamma > 0$ and all j, where $\tilde{\phi}_j$ is $\sqrt{n}$-consistent for ϕ_j^*. As $n \to \infty$, if $\lambda_n/\sqrt{n} \to 0$, $\lambda_n n^{(\gamma-1)/2} \to \infty$, and $n_k/n \to \alpha_k > 0$ for $k = 1, \ldots, p$, then there exists a local maximizer $\hat{\boldsymbol{\theta}}$ of $R(\boldsymbol{\theta})$ such that $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\| = O_p(n^{-1/2})$. Furthermore, with probability tending to one, the $\sqrt{n}$-consistent local maximizer $\hat{\boldsymbol{\theta}}$ must satisfy $\hat{\boldsymbol{\phi}}_{\mathcal{A}} = \mathbf{0}$.

Remark 4. To achieve consistency in model selection with the adaptive lasso penalty, we need some consistent estimate of the vector $\boldsymbol{\phi}$ to construct the weights. Theorem 2 suggests that we first use $\tau_j = \min(|\hat{\phi}_j^{(\text{OLS})}|^{-\gamma}, M^\gamma)$ as weights to obtain

an initial consistent estimate $\tilde{\boldsymbol{\phi}}$. Then with weights constructed from $\tilde{\boldsymbol{\phi}}$, Theorem 3 guarantees model selection consistency. In a similar spirit, Shojaie and Michailidis (2010) proposed a two-stage lasso penalty where the initial estimate is constructed by a regular lasso. As seen from Theorem 2, our initial estimate is in fact obtained by a weighted lasso with weights bounded from above.

5. SIMULATION STUDY

5.1 Performance of the CD Algorithm

To test the performance of the CD algorithm, we conducted a simulation study. We randomly generated graphs with p nodes ($p = 20, 50, 100, 200$) and $2p$ edges. To further control the sparsity of the graphs, we set the maximum number of parents for any given node to be 4. For each value of p, we simulated 10 different random graphs, and for each graph, three datasets were generated according to Equation (3) with $\beta_{ij} = 0.2, 0.5$, and 1.0, respectively. The variance σ_j^2 of the Gaussian noise variable ε_j ($j = 1, \ldots, p$) was set to 1 in all our simulation. The sample size of each dataset is $n = 5p$. As described in Section 2, a data matrix is divided into p blocks such that the sample size of each block is $n_j = 5$, $j = 1, \ldots, p$. The jth block $\mathbf{X}^j$ contains experimental data on the node X_j, which were drawn from the standard normal distribution $\mathcal{N}(0, 1)$. For each dataset, we applied the CD algorithm to compute the solution path using a geometric sequence of λ's, starting from the largest value $\lambda_{\max}$ for which $\widehat{\mathbf{B}}_{\lambda_{\max}} = \mathbf{0}$ and decreasing to the smallest value $\lambda_{\min}$. The sequence typically contained 50 or 100 different values of λ's with the ratio $\lambda_{\min}/\lambda_{\max}$ set to some small value such as 0.001. Graphical models were then selected according to (11) with $\alpha = 0.1$. We used $\gamma = 0.15$ for all datasets, except for the two cases with $p \geq 100$ and $\beta_{ij} = 1.0$, where γ was set to 0.5.

Table 1 summarizes the average performance of the CD algorithm over 10 datasets for each combination of p and β_{ij}. For instance, when $p = 100$ and $\beta_{ij} = 0.5$, the estimated graphical model on average contains 220.9 directed edges, of which 156.5 edges are present in the true graph, 28.3 edges have directions reversed, and the rest 36.1 edges are not included in the true graph. On average, there are also 15.2 true edges missing in the estimated model. Results in Table 1 suggest that our method can estimate the structure of a DAG with reasonable accuracy even when the sample size is limited. All TPRs (defined in Table 1) are above 0.70 except for cases with $\beta_{ij} = 0.2$, where signal-to-noise ratios are too small. The accuracy of estimation can be greatly improved if a large sample is available (see Table S1 in the online supplementary materials). Note that when $p = 200$, the number of parameters to be estimated is around 20,000, which is much larger than the sample size $n = 5p = 1000$. Even in this high-dimensional setting, our CD algorithm was still able to estimate DAGs quite accurately.

Since the CD algorithm computes a set of solutions along the solution path of problem (7), another way to evaluate the performance is to investigate the relationship between TPR and false positive rate [FPR $= (\mathrm{R} + \mathrm{FP})/(p(p-1) - \mathrm{T})$] as the penalty parameter λ varies, which is known as the receiver operating characteristic (ROC) analysis. However, since the sequence of λ's we used was data dependent, we examined the TPR–FPR relationships as the number of predicted edges increases. Figure 2 presents the results of the ROC analysis. Again, these ROC curves suggest satisfactory performance of the CD algorithm except when the signal-to-noise ratio is small ($\beta_{ij} = 0.2$). In particular, we note that for large networks ($p = 100, 200$), as we increase the number of predicted edges and the complexity of estimated graphs by adjusting the penalty λ, we will increase the TPR without affecting the FPR that much until the TPR reaches a plateau, a level at which the estimated DAGs are structurally similar to the true DAG. This is consistent with our sensitivity analysis on the tuning parameter α in (11). The analysis shows that when α is greater than 0.1, the FDR falls into an acceptable and stable level (see Figure S3). Further decrease in α to below 0.05 would result in a drastic increase in false positive edges.

Table 1. The average number of predicted (P), expected (E), reversed (R), missed (M), and false positive (FP) edges and the average true positive rate (TPR[a]) and false discovery rate (FDR[b]) for DAGs learned by the CD algorithm

		CD algorithm							KO method	
p	β_{ij}	P	E	R	M	FP	TPR	FDR	TPR	FDR
20	0.2	59.6	17.3	10.9	11.8	31.4	0.433(0.069)	0.694(0.080)	0.375	0.213
	0.5	48.6	29.2	6.1	4.7	13.3	0.730(0.152)	0.399(0.083)	0.908	0.086
	1.0	65.5	34.0	2.8	3.2	28.7	0.850(0.092)	0.429(0.138)	0.723	0.065
50	0.2	158.9	54.0	32.8	13.2	72.1	0.540(0.048)	0.652(0.061)	0.732	0.128
	0.5	114.9	74.5	17.4	8.1	23.0	0.745(0.100)	0.351(0.085)	0.992	0.045
	1.0	132.7	70.5	5.0	24.5	57.2	0.705(0.113)	0.453(0.090)	0.763	0.050
100	0.2	246.0	137.9	53.2	8.9	54.9	0.690(0.027)	0.431(0.075)	0.952	0.088
	0.5	220.9	156.5	28.3	15.2	36.1	0.783(0.058)	0.290(0.071)	0.993	0.032
	1.0	167.8	149.1	11.4	39.5	7.3	0.746(0.087)	0.109(0.074)	0.508	0.011
200	0.2	421.2	325.3	72.2	2.5	23.7	0.813(0.054)	0.226(0.061)	1.000	0.051
	0.5	430.2	341.8	41.9	16.3	46.5	0.855(0.053)	0.203(0.071)	1.000	0.016
	1.0	328.1	298.5	18.3	83.2	11.3	0.746(0.100)	0.090(0.049)	0.549	0.004

NOTE:
1. The numbers in parentheses are the standard deviations across 10 datasets.
2. As a comparison, the last two columns list the average TPR and FDR for DAGs estimated by the approach of Shojaie and Michailidis (2010) assuming that the ordering of the variables is known. (KO: known ordering).
[a]TPR = E/T, where T = $2p$ is the total number of true edges; [b]FDR = (R + FP)/P.

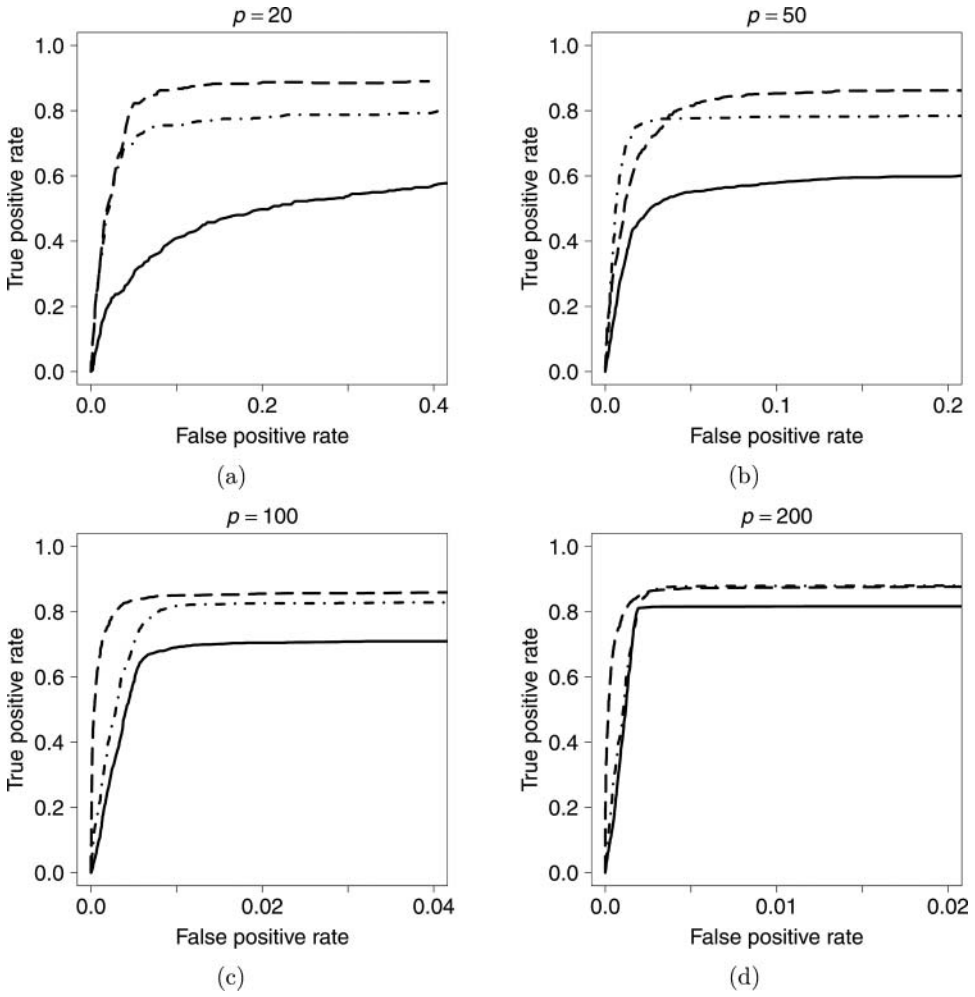

Figure 2. ROC curves for $\beta_{ij} = 0.2$ (solid lines), $\beta_{ij} = 0.5$ (dot-dashed lines), and $\beta_{ij} = 1.0$ (long-dashed lines).

As expected, the performance of the CD algorithm decreases when the graph is less sparse. We varied the number of edges from p to $4p$ in our simulation and found a decrease in the TPR with an increase in the FDR when the underlying DAG became denser. However, even for the most dense cases, the result is still reasonably good, as reported in Table S2 in the online supplementary materials. Note that the parameter α was fixed to 0.1 for all the simulation results. It seems that this choice may control the FDR at an acceptable level unless the sample size is too small or the signal-to-noise ratio is too low (small β_{ij}). To obtain a rough measure of the amount of information that interventional data can provide to resolve directionality of DAGs, we also applied the CD algorithm to simulated observational data with the same sample sizes as their interventional counterparts. The results are summarized in Tables S3 in the online supplementary materials. We found that interventional data helped to increase the TPR and simultaneously reduce the FDR, and the boost in the TPR ranges from 2% up to about 50%.

5.2 Comparison With Other Methods

To benchmark the performance of the CD algorithm, we compared our method to a PC-algorithm-based approach. The PC algorithm is a classical constraint-based method that can estimate DAGs with hundreds of nodes. We did not compare with Monte Carlo approaches, as even the most recent developments, such as the order-graph sampler (Ellis and Wong 2008), have not shown convincing performance on graphs with more than 50 nodes.

The PC algorithm is designed to estimate from observational data a completed partially directed acyclic graph (CPDAG), which contains both directed and undirected edges. We therefore took a two-step approach to produce results favorable for the PC algorithm. We first used the PC algorithm to estimate a CPDAG from data. Then one may try to estimate the direction of an undirected edge using interventions and produce a DAG. In this comparison, however, we simply counted an undirected edge between two nodes in a CPDAG as an expected edge, provided that there is a corresponding directed edge between the two nodes in the true DAG. Thus, the reported result is the best (or an upper bound) one can obtain by a two-step PC-algorithm-based method (PC-based method). The performance of this PC-based method applied to our simulated datasets is shown in Table 2. Unlike graphs selected by criterion (11), a graph learned by the PC-based method generally has fewer edges than the true model. So to make a fair comparison, we selected from the solution path constructed by the CD algorithm the graph that matches the total number of edges of the graph learned by the PC-based method. The corresponding results are also presented in Table 2. It can be easily seen that the CD algorithm outperforms the

Table 2. Performance comparison between the two-step PC-based method and the CD algorithm

		PC-based method			CD algorithm		
p	β_{ij}	P	TPR	FDR	P	TPR	FDR
20	0.2	7.6	0.103(0.042)	0.443(0.262)	8.3	0.115(0.044)	0.442(0.184)
	0.5	18.4	0.313(0.049)	0.311(0.134)	19.8	0.383(0.095)	0.227(0.148)
	1.0	15.7	0.290(0.061)	0.254(0.172)	15.7	0.318(0.103)	0.183(0.088)
50	0.2	53.3	0.221(0.037)	0.585(0.071)	52.8	0.299(0.032)	0.430(0.072)
	0.5	70.3	0.409(0.081)	0.422(0.082)	72.5	0.557(0.117)	0.233(0.109)
	1.0	54.7	0.313(0.042)	0.427(0.054)	50.5	0.355(0.094)	0.296(0.080)
100	0.2	173.8	0.399(0.050)	0.542(0.051)	173.5	0.610(0.034)	0.297(0.026)
	0.5	153.1	0.456(0.053)	0.405(0.048)	154.6	0.596(0.077)	0.231(0.066)
	1.0	107.8	0.328(0.069)	0.396(0.096)	107.2	0.513(0.042)	0.041(0.051)
200	0.2	429.1	0.506(0.030)	0.528(0.028)	431.8	0.815(0.053)	0.245(0.051)
	0.5	351.4	0.493(0.075)	0.438(0.085)	357.4	0.734(0.093)	0.181(0.068)
	1.0	235.3	0.335(0.056)	0.433(0.069)	234.8	0.561(0.059)	0.043(0.046)

NOTE: The numbers in parentheses are the standard deviations across 10 datasets.

PC-based method in all the cases of our simulation. Graphs estimated using our method have both higher TPRs and lower FDRs. This result shows the advantage of using experimental data in an integrated penalized likelihood method. In addition, we compared the performance of the CD algorithm and the PC-based method on observational data (see Table S4 in the online supplementary materials). We found that our method still outperforms the PC-based method for most cases.

We also compared the running time for both methods. Table 3 summarizes the CPU time for one run of each algorithm averaged over 10 datasets. Each run of the CD algorithm uses a sequence of 50 λ's with $\lambda_{\min}/\lambda_{\max} = 0.001$. The CD algorithm is implemented in R with the majority of its core computation executed in C programs. The PC algorithm we used was implemented by Kalisch et al. (2012) in the R package *pcalg*. The running time for the PC algorithm depends on the argument *u2pd*, which we assume to be *rand* (see online manuals for further details). According to Table 3, the average CPU time for the PC algorithm is shorter than the CD algorithm. However, considering that the CD algorithm estimates 50 (or more generally a sequence of) graphical models in each run, it is on average at least as fast as the PC algorithm for estimating a single graph.

Recently, Shojaie and Michailidis (2010) developed an approach to estimate DAGs assuming a known ordering of the variables, which we will refer to as the KO method. Knowing the ordering greatly simplifies the structure learning problem. Following their formulation, we can simply estimate the coefficient matrix **B** (and thus the structure of directed graphs) by regressing each variable on all preceding variables in a given ordering. Hence, the problem of estimating directed graphs becomes $p - 1$ separate lasso problems, which can be solved efficiently using either the least angle regression (LARS) algorithm (Efron et al. 2004) or the pathwise CD algorithm (Friedman et al. 2007). To obtain an estimate of a directed graph, Shojaie and Michailidis (2010) proposed to use $\lambda_i(\delta) = 2\tilde{n}^{-1/2} Z^*_{\delta/[2p(i-1)]}$ as the penalty for the ith individual lasso problem, where $\tilde{n}$ is the sample size, Z^*_q is the $(1-q)$th quantile of the standard normal distribution, and δ is a parameter controlling the probability of falsely joining two ancestral sets in a graph (see Shojaie and Michailidis 2010). We applied their method to our simulated datasets. Though this criterion worked well with a large sample size (see Table S1), it led to over-sparse solutions when applied to our datasets with a limited sample size ($n = 5p$). We thus scaled down the tuning parameters $\lambda_i(\delta)$ proportionately and the results are summarized in Table 1 (KO method). The δ level was chosen to be 0.1 as suggested by Shojaie and Michailidis (2010). As anticipated, most of the results obtained by assuming a known ordering are clearly better than the results of the CD algorithm. However, almost all the TPRs from the CD algorithm are above 75% of those from the KO method. Furthermore, the CD algorithm seemed to outperform the KO method when $p = 200$ and $\beta_{ij} = 1.0$. This comparison demonstrates the gain in prediction accuracy when the assumed ordering of the variables is correct. The gain mostly comes from a lower FDR as no reversed edges will be produced. However, such an assumption is often risky in practical

Table 3. Comparison of average CPU time (in seconds) between the PC algorithm and the CD algorithm

	PC algorithm				CD algorithm			
	$p = 20$	$p = 50$	$p = 100$	$p = 200$	$p = 20$	$p = 50$	$p = 100$	$p = 200$
$\beta_{ij} = 0.2$	0.04	0.28	4.18	28.74	0.09	1.32	17.54	255.09
$\beta_{ij} = 0.5$	0.09	1.23	9.67	76.94	0.15	4.54	112.69	1938.23
$\beta_{ij} = 1.0$	0.09	0.97	5.10	33.73	0.32	10.05	193.17	4595.95
Mean	0.07	0.83	6.32	46.47	0.19	5.30	107.80	2263.09

applications. Fortunately, the promising result of the CD algorithm for large networks with a reasonably strong signal ($p \geq 100$ and $\beta_{ij} \geq 0.5$) suggests that estimating a DAG without knowing the ordering is reliable with sufficient data.

6. REAL DATA EXAMPLE

In this section, we analyze a flow cytometry dataset generated by Sachs et al. (2005). This dataset contains simultaneous measurement on $p = 11$ protein and phospholipid components of the signaling network in human immune system cells. The original dataset contains continuous data collected from $n = 7466$ cells and consists of a mixture of observational and experimental samples on the 11 components. The dataset analyzed by Sachs et al. (2005) is a discretized version of the continuous dataset. A number of researchers studied the flow cytometry dataset, among whom Friedman, Hastie, and Tibshirani (2008) and Shojaie and Michailidis (2010) analyzed the continuous version.

Figure 3(a) shows the known causal interactions among the 11 components of the signaling network. These causal relationships are well established, and no consensus has been reached on interactions beyond those present in the network. Thus, this network structure is often used as the benchmark to assess the accuracy of an estimated network structure, and we therefore call it the consensus model. Friedman, Hastie, and Tibshirani (2008) applied the graphical lasso to this dataset and estimated a number of graphical models using different values of the L_1 penalty. Their models are all undirected and they observed moderate agreement between one of their estimates and the consensus model. Shojaie and Michailidis (2010) also analyzed the same dataset using their penalized likelihood method by assuming the ordering of the variables is known a priori. Their estimated DAG using the adaptive lasso penalty is shown in Figure 3(b). This graph has 27 directed edges in total, among which 14 are expected and 13 are false positives. We obtained a sequence of estimated DAGs by applying the CD algorithm to the continuous flow cytometry data. One of them is shown in Figure 3(c). Our model also has a total of 27 directed edges, of which 8 are expected, 6 are reversed, and 13 are false positives. It seems that the performance of the CD algorithm, if ignoring the directionality, is very comparable to the method assuming a known ordering.

To test the robustness of our method, we applied the CD algorithm to the discrete version of the flow cytometry dataset,

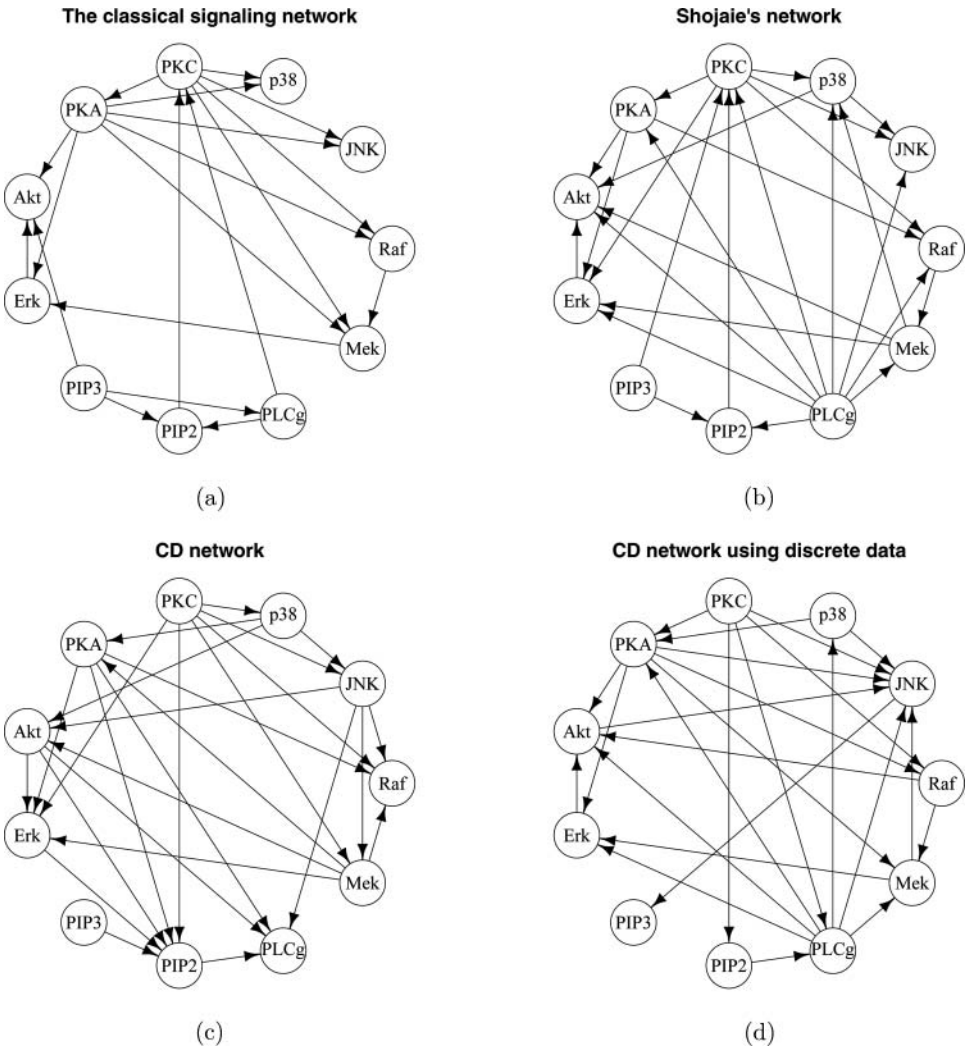

Figure 3. (a) The classical signaling network of human immune system cells, (b) Shojaie's network estimated from the continuous flow cytometry dataset. The CD networks estimated from (c) the continuous flow cytometry dataset and (d) the discrete flow cytometry dataset.

Table 4. Comparison among the CD algorithm, the order-graph sampler, and the multidomain sampler applied to the discrete flow cytometry dataset

Method	P	E	M	R+FP
CD algorithm (26 edges)	26	11	5	15
CD algorithm (20 edges)	20	9	9	11
Order-graph sampler	20	8	8	12
Multidomain sampler	25.9	15.55	2.4	10.35

NOTE: The order-graph sampler result comes from the mean graph (figure 11) in Ellis and Wong (2008), while the multidomain sampler result is the average over 20 independent runs (see Zhou 2011, table 3).

which has $n = 5400$ cells. The discretization transformed the data into three levels, high, medium, and low, which are coded as 2, 1, and 0, respectively. As a result, the magnitude of the original measurement is partially preserved in the discrete data. An estimated DAG with 26 edges is shown in Figure 3(d). To our surprise, this graph is qualitatively better than the one estimated using the continuous dataset. In this graph, there are 11 expected edges and 15 false predictions (R+FP; see Table 4). We also applied the CD algorithm to 100 bootstrap samples generated from the discrete dataset to assess the sensitivity of our method to data perturbation. For each bootstrap sample, we selected a model with 26 edges and found that on average it shared 23.3 edges with the model shown in Figure 3(d), which confirms that our method is quite robust to data perturbation. Moreover, though our method was designed for Gaussian data, we were still able to obtain a reasonable network structure from the discretized dataset, which does not satisfy the Gaussian assumption.

Compared to the estimate obtained by Ellis and Wong (2008) using their order-graph sampler, our result with 20 predicted edges is slightly better in terms of the number of expected edges (E) and false predictions (R+FP; see Table 4). The multidomain sampler, recently developed by Zhou (2011) for Bayesian inference, yields better result than the CD algorithm. However, the CD algorithm is much faster than these Monte Carlo sampling approaches. For large networks with hundreds of nodes, the CD algorithm can still be used to obtain reasonably good estimates of DAGs, while Monte Carlo methods may not be applicable due to their long running time.

7. DISCUSSION

We have developed a method to estimate the structure of causal Gaussian networks using a penalized likelihood approach with the adaptive lasso penalty. Without knowing the ordering of the variables, we rely on experimental data to retrieve information about the directionality of the edges in a graph. The acyclicity constraint on the structure of BNs presents a challenge to the maximization of the penalized log-likelihood function. A blockwise CD algorithm has been developed for this optimization problem. The algorithm runs reasonably fast and can be applied to large networks. A simulation study has been conducted to demonstrate the performance of our method for BNs of various sizes, and a real data example is shown as well. Throughout this article, variables are assumed to be Gaussian, although our approach may be applied to datasets from other distributions as demonstrated by the result on the discrete flow cytometry data. However, a more principled generalization to other data types is expected to have a better performance.

We have established asymptotic properties for the penalized maximum likelihood estimator of the coefficient matrix of a GBN, assuming that the number of variables p is fixed. Asymptotic theory for the estimator if p is allowed to grow as a function of the sample size remains to be established in the future. This type of asymptotic problems has been studied in various settings of undirected graph and precision matrix estimation (e.g., Meinshausen and Bühlmann 2006; Lam and Fan 2009), where $p(n) = O(n^c)$ for some $c > 0$ or is of an even higher order. Following our current setup, however, we may need to restrict our attention to the case where $0 < c < 1$ so that every variable will have sufficient interventional data as $n \to \infty$. The satisfactory results in our simulation for $p \geq 100$ and $n = 5p$ seem to suggest that our CD algorithm is effective even for $p > \sqrt{n}$. It will be interesting to study the theoretical properties of this penalized likelihood approach when not all variables have experimental data, for which the concept of interventional Markov equivalence (Hauser and Bühlmann 2012) will be relevant.

APPENDIX: PROOFS

Proof of Theorem 1. We prove the first claim (15) by contradiction. Suppose $\boldsymbol{\theta} \neq \boldsymbol{\theta}^*$ and $f(\mathbf{x}|\boldsymbol{\theta}_k) = f(\mathbf{x}|\boldsymbol{\theta}_k^*)$ a.e. for $k = 1, \ldots, p$. Let $\mathcal{S}(\mathcal{G})$ denote the set of topological sorts of a DAG $\mathcal{G}$. Recall that we denote by $\mathcal{G}_{\boldsymbol{\phi}}$ and $\mathcal{G}_{\boldsymbol{\phi}^*}$ the DAGs induced by $\boldsymbol{\phi}$ and $\boldsymbol{\phi}^*$, respectively. There are two possibilities between the topological sorts of $\mathcal{G}_{\boldsymbol{\phi}}$ and $\mathcal{G}_{\boldsymbol{\phi}^*}$ if $\boldsymbol{\theta} \neq \boldsymbol{\theta}^*$.

Case 1: $\mathcal{S}(\mathcal{G}_{\boldsymbol{\phi}}) \cap \mathcal{S}(\mathcal{G}_{\boldsymbol{\phi}^*}) \neq \varnothing$. Let $\sqsubset \in \mathcal{S}(\mathcal{G}_{\boldsymbol{\phi}}) \cap \mathcal{S}(\mathcal{G}_{\boldsymbol{\phi}^*})$, that is, an ordering compatible with both $\mathcal{G}_{\boldsymbol{\phi}}$ and $\mathcal{G}_{\boldsymbol{\phi}^*}$. Assume without loss of generality that in this ordering $i \prec j$ if $i < j$. Apparently, $\sqsubset$ is also compatible with $\mathcal{G}_{\boldsymbol{\phi}_k}$ and $\mathcal{G}_{\boldsymbol{\phi}_k^*}$ for $k = 1, \ldots, p$. Then we can write $f(\mathbf{x}|\boldsymbol{\theta}_k) = \prod_{i=1}^p f(x_i|x_1, \ldots, x_{i-1}, \boldsymbol{\theta}_k) = \prod_{i=1}^p f(x_i|\Pi_i^{\mathcal{G}_{\boldsymbol{\phi}_k}}, \boldsymbol{\theta}_k)$ and similarly $f(\mathbf{x}|\boldsymbol{\theta}_k^*) = \prod_{i=1}^p f(x_i|\Pi_i^{\mathcal{G}_{\boldsymbol{\phi}_k^*}}, \boldsymbol{\theta}_k^*)$. Since $f(\mathbf{x}|\boldsymbol{\theta}_k) = f(\mathbf{x}|\boldsymbol{\theta}_k^*)$, it follows that $\Pi_i^{\mathcal{G}_{\boldsymbol{\phi}_k}} = \Pi_i^{\mathcal{G}_{\boldsymbol{\phi}_k^*}}$ for all i and thus $\mathcal{G}_{\boldsymbol{\phi}_k} = \mathcal{G}_{\boldsymbol{\phi}_k^*}$ for all k. However, since $\boldsymbol{\theta} \neq \boldsymbol{\theta}^*$, there exists some k such that $\boldsymbol{\theta}_k \neq \boldsymbol{\theta}_k^*$. Therefore, there exists a k such that the common multivariate normal density $f(\mathbf{x}|\boldsymbol{\theta}_k) = f(\mathbf{x}|\boldsymbol{\theta}_k^*)$, factorized according to a common structure $\mathcal{G}_{\boldsymbol{\phi}_k} = \mathcal{G}_{\boldsymbol{\phi}_k^*}$, can be parameterized by two different parameters $\boldsymbol{\theta}_k$ and $\boldsymbol{\theta}_k^*$. This is apparently impossible.

Case 2: $\mathcal{S}(\mathcal{G}_{\boldsymbol{\phi}}) \cap \mathcal{S}(\mathcal{G}_{\boldsymbol{\phi}^*}) = \varnothing$, that is, none of the orderings of $\mathcal{G}_{\boldsymbol{\phi}^*}$ is compatible with $\mathcal{G}_{\boldsymbol{\phi}}$. In this case, there must exist a pair of indices (i, j) such that in $\mathcal{G}_{\boldsymbol{\phi}^*}$, $X_i \in \text{an}(X_j)$, but in $\mathcal{G}_{\boldsymbol{\phi}}$, X_j is a nondescendant of X_i. Then X_j is independent of X_i in $f(\mathbf{x}|\boldsymbol{\theta}_i)$, since in $\mathcal{G}_{\boldsymbol{\phi}_i}$, X_i has no parents and X_j is a nondescendant of X_i. So $\text{cov}(X_i, X_j) = 0$ in $f(\mathbf{x}|\boldsymbol{\theta}_i)$. However, in $\mathcal{G}_{\boldsymbol{\phi}_i^*}$, we still have $X_i \in \text{an}(X_j)$. It is easy to show that $\text{cov}(X_i, X_j) = \beta_{i \to j}^* \text{var}(X_i) \neq 0$ in $f(\mathbf{x}|\boldsymbol{\theta}_i^*)$ since $\boldsymbol{\theta}^*$ is natural. Therefore, there exists $1 \leq i \leq p$ such that $f(\mathbf{x}|\boldsymbol{\theta}_i) \neq f(\mathbf{x}|\boldsymbol{\theta}_i^*)$, which contradicts our assumption.

So in both Case 1 and Case 2, we have a contradiction. Thus, the first claim holds.

For the second claim (16), first note that by the law of large numbers,

$$\frac{1}{n}(L(\boldsymbol{\theta}) - L(\boldsymbol{\theta}^*)) = \sum_{k=1}^p \frac{n_k}{n} \frac{1}{n_k} \sum_{h \in \mathcal{I}_k} \log \frac{f(\mathbf{X}_{[h,\cdot]}|\boldsymbol{\theta}_k)}{f(\mathbf{X}_{[h,\cdot]}|\boldsymbol{\theta}_k^*)} \longrightarrow_p \sum_{k=1}^p \alpha_k \mathbf{E}_{\boldsymbol{\theta}_k^*} \left[\log \frac{f(\mathbf{Y}|\boldsymbol{\theta}_k)}{f(\mathbf{Y}|\boldsymbol{\theta}_k^*)} \right], \quad \text{(A.1)}$$

where $\mathbf{Y}$ is a random vector with probability density $f(\mathbf{x}|\boldsymbol{\theta}_k^*)$. Then the desired result follows immediately using Jensen's inequality and (15). □

Proof of Theorem 2. Define $a_n = 1/\sqrt{n}$ and $\mathcal{B} = \{j : \phi_j^* \neq 0\}$. Let

$$\mathbf{I}(\boldsymbol{\theta}_k) = \mathbf{E}_{\boldsymbol{\theta}_k}\left\{\left[\frac{\partial}{\partial \boldsymbol{\theta}_k}\log f(\mathbf{x}|\boldsymbol{\theta}_k)\right]\left[\frac{\partial}{\partial \boldsymbol{\theta}_k}\log f(\mathbf{x}|\boldsymbol{\theta}_k)\right]^T\right\}$$

be the Fisher information matrix.

Consider $\boldsymbol{\theta} = (\boldsymbol{\phi}, \boldsymbol{\sigma}^2, \widetilde{\boldsymbol{\sigma}}^2) \in \mathrm{nb}(\boldsymbol{\theta}^*)$, where $\mathrm{nb}(\boldsymbol{\theta}^*)$ is an arbitrarily small neighborhood of $\boldsymbol{\theta}^*$. The components of $\boldsymbol{\phi}$ must satisfy $\phi_i\phi_i^* > 0$ if $\phi_i^* \neq 0$ $(i = 1, \ldots, d)$, since otherwise $\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\| \geq \min_{j:\phi_j^*\neq 0}|\phi_j^*|$. In particular, this implies that if $\boldsymbol{\theta} \in \mathrm{nb}(\boldsymbol{\theta}^*)$, $i \to j$ in $\mathcal{G}_{\boldsymbol{\phi}}$ for all $i \to j$ in $\mathcal{G}_{\boldsymbol{\phi}^*}$ and thus $\mathcal{G}_{\boldsymbol{\phi}}$ and $\mathcal{G}_{\boldsymbol{\phi}^*}$ have a compatible ordering. If we restrict to the lower-dimensional space $\boldsymbol{\Omega}_k = \{\boldsymbol{\theta}_k : \boldsymbol{\theta} \in \boldsymbol{\Omega}\}$, the same arguments apply to an arbitrarily small neighborhood of $\boldsymbol{\theta}_k^*$ in this space, that is, $\mathcal{G}_{\boldsymbol{\phi}_k}$ and $\mathcal{G}_{\boldsymbol{\phi}_k^*}$ have a compatible ordering as well. Then it follows from the arguments used in Case 1 in the proof of Theorem 1 that $f(\mathbf{x}|\boldsymbol{\theta}_k) \neq f(\mathbf{x}|\boldsymbol{\theta}_k^*)$ for $\boldsymbol{\theta}_k \in \mathrm{nb}(\boldsymbol{\theta}_k^*)\backslash\{\boldsymbol{\theta}_k^*\}$. Since f is a Gaussian density, it follows that $\mathbf{I}(\boldsymbol{\theta}_k^*)$ is positive definite for all k.

Let $\mathbf{u} \in \{\mathbf{u} : \boldsymbol{\theta}^* + a_n\mathbf{u} \in \boldsymbol{\Omega}\}$ and denote its components by u_j. Let $\mathbf{u}_k$ be the vector defined in the same way as $\boldsymbol{\theta}_k$, $k = 1, \ldots, p$. Note that $\sum_{k=1}^p \|\mathbf{u}_k\|^2 \geq \|\mathbf{u}\|^2$. Let $\delta_{\min}^k > 0$ be the minimal eigenvalue of $\mathbf{I}(\boldsymbol{\theta}_k^*)$ and $\rho = \min_k(\alpha_k\delta_{\min}^k/2)$. Then

$$\sum_{k=1}^p \frac{\alpha_k}{2}\mathbf{u}_k^T\mathbf{I}(\boldsymbol{\theta}_k^*)\mathbf{u}_k \geq \sum_{k=1}^p \frac{\alpha_k}{2}\delta_{\min}^k\|\mathbf{u}_k\|^2 \geq \rho\sum_{k=1}^p\|\mathbf{u}_k\|^2 \geq \rho\|\mathbf{u}\|^2. \tag{A.2}$$

Now we study the behavior of $R(\boldsymbol{\theta})$ in a small neighborhood of the true value $\boldsymbol{\theta}^*$ by expanding $L(\boldsymbol{\theta})$ around $\boldsymbol{\theta}^*$. We have, as $n \to \infty$,

$$\begin{aligned}
&R(\boldsymbol{\theta}^* + a_n\mathbf{u}) - R(\boldsymbol{\theta}^*)\\
&\quad\leq L(\boldsymbol{\theta}^* + a_n\mathbf{u}) - L(\boldsymbol{\theta}^*) - \lambda_n\sum_{j\in\mathcal{B}}\tau_j(|\phi_j^* + a_nu_j| - |\phi_j^*|)\\
&\quad= \sum_{k=1}^p[L_k(\boldsymbol{\theta}_k^* + a_n\mathbf{u}_k) - L_k(\boldsymbol{\theta}_k^*)] - \lambda_na_n\sum_{j\in\mathcal{B}}\tau_ju_j\mathrm{sgn}(\phi_j^*)\\
&\quad= \sum_{k=1}^p\left[a_nL_k'(\boldsymbol{\theta}_k^*)^T\mathbf{u}_k - \frac{1}{2}n_ka_n^2\mathbf{u}_k^T\mathbf{I}(\boldsymbol{\theta}_k^*)\mathbf{u}_k\{1 + o_p(1)\}\right]\\
&\qquad- \lambda_na_n\sum_{j\in\mathcal{B}}\tau_ju_j\mathrm{sgn}(\phi_j^*)\\
&\quad= \sum_{k=1}^p\left[\sqrt{\alpha_k}\frac{L_k'(\boldsymbol{\theta}_k^*)^T}{\sqrt{n_k}}\mathbf{u}_k\{1 + o_p(1)\} - \frac{\alpha_k}{2}\mathbf{u}_k^T\mathbf{I}(\boldsymbol{\theta}_k^*)\mathbf{u}_k\{1 + o_p(1)\}\right]\\
&\qquad- \frac{\lambda_n}{\sqrt{n}}\sum_{j\in\mathcal{B}}\tau_ju_j\mathrm{sgn}(\phi_j^*)\\
&\quad\leq \sum_{k=1}^p\left[\sqrt{\alpha_k}\frac{L_k'(\boldsymbol{\theta}_k^*)^T}{\sqrt{n_k}}\mathbf{u}_k\{1 + o_p(1)\}\right] - \rho\|\mathbf{u}\|^2\{1 + o_p(1)\}\\
&\qquad- \frac{\lambda_n}{\sqrt{n}}\sum_{j\in\mathcal{B}}\tau_ju_j\mathrm{sgn}(\phi_j^*).
\end{aligned}\tag{A.3}$$

The last inequality is due to (A.2). From the central limit theorem, $n_k^{-1/2}\|L_k'(\boldsymbol{\theta}_k^*)\| = O_p(1)$ for all k. By assumption, $\tau_j = O_p(1)$ for $j = 1, \ldots, d$ and $\lambda_n/\sqrt{n} = o_p(1)$. Therefore, for a sufficiently large C, the second term in the last line of (A.3) dominates the first and the third terms uniformly in $\{\mathbf{u} : \|\mathbf{u}\| = C, \boldsymbol{\theta}^* + a_n\mathbf{u} \in \boldsymbol{\Omega}\}$. Hence, for any given $\varepsilon > 0$, there exists a sufficiently large C such that

$$P\left(\sup_{\|\mathbf{u}\|=C}R(\boldsymbol{\theta}^* + a_n\mathbf{u}) < R(\boldsymbol{\theta}^*)\right) \geq 1 - \varepsilon, \tag{A.4}$$

which implies that with probability at least $1 - \varepsilon$, there exists a local maximizer $\hat{\boldsymbol{\theta}}$ of $R(\boldsymbol{\theta})$ in the ball $\{\boldsymbol{\theta}^* + a_n\mathbf{u} \in \boldsymbol{\Omega} : \|\mathbf{u}\| < C\}$. Thus, there exists a local maximizer $\hat{\boldsymbol{\theta}}$ of $R(\boldsymbol{\theta})$ such that $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\| = O_p(n^{-1/2})$.

SUPPLEMENTARY MATERIALS

Some technical proofs, the algorithm for checking the acyclicity constraint, and additional results can be found in the supplementary materials.

[Received October 2011. Revised November 2012.]

REFERENCES

Banerjee, O., El Ghaoui, L., and d'Aspremont, A. (2008), "Model Selection Through Sparse Maximum Likelihood Estimation for Multivariate Gaussian or Binary Data," *Journal of Machine Learning Research*, 9, 485–516. [288]

Cooper, G. F., and Herskovits, E. (1992), "A Bayesian Method for the Induction of Probabilistic Networks From Data," *Machine Learning*, 9, 309–347. [288]

Donoho, D. L., and Johnstone, I. M. (1995), "Adapting to Unknown Smoothness via Wavelet Shrinkage," *Journal of the American Statistical Association*, 90, 1200–1224. [291]

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004), "Least Angle Regression," *The Annals of Statistics*, 32, 407–499. [296]

Ellis, B. (2006), "Inference on Bayesian Network Structures," unpublished Ph.D. dissertation, Harvard University. [291]

Ellis, B., and Wong, W. H. (2008), "Learning Causal Bayesian Network Structures From Experimental Data," *Journal of the American Statistical Association*, 103, 778–789. [288,295,298]

Fan, J., Feng, Y., and Wu, Y. (2009), "Network Exploration via the Adaptive Lasso and SCAD Penalties," *The Annals of Applied Statistics*, 3, 521–541. [293]

Fan, J., and Li, R. (2001), "Variable Selection via Nonconcave Penalized Likelihood and Its Oracle Properties," *Journal of the American Statistical Association*, 96, 1348–1360. [290]

Friedman, J., Hastie, T., Höfling, H., and Tibshirani, R. (2007), "Pathwise Coordinate Optimization," *The Annals of Applied Statistics*, 1, 302–332. [290,296]

Friedman, J., Hastie, T., and Tibshirani, R. (2008), "Sparse Inverse Covariance Estimation With the Graphical Lasso," *Biostatistics*, 9, 432–441. [288,297]

——— (2010), "Regularization Paths for Generalized Linear Models via Coordinate Descent," *Journal of Statistical Software*, 33, 1–22. [291]

Friedman, N., and Koller, D. (2003), "Being Bayesian About Network Structure: A Bayesian Approach to Structure Discovery in Bayesian Networks," *Machine Learning*, 50, 95–125. [288]

Fu, W. (1998), "Penalized Regressions: The Bridge versus the Lasso," *Journal of Computational and Graphical Statistics*, 7, 397–416. [290]

Hauser, A., and Bühlmann, P. (2012), "Characterization and Greedy Learning of Interventional Markov Equivalence Classes of Directed Acyclic Graphs," *Journal of Machine Learning Research*, 13, 2409–2464. [293,298]

Heckerman, D., Geiger, D., and Chickering, D. M. (1995), "Learning Bayesian Networks: The Combination of Knowledge and Statistical Data," *Machine Learning*, 20, 197–243. [288]

Kalisch, M., and Bühlmann, P. (2007), "Estimating High-Dimensional Directed Acyclic Graphs With the PC-Algorithm," *Journal of Machine Learning Research*, 8, 613–636. [288]

Kalisch, M., Mächler, M., Colombo, D., Maathuis, M. H., and Bühlmann, P. (2012), "Causal Inference Using Graphical Models With the R Package pcalg," *Journal of Statistical Software*, 47, 1–26. [296]

Lam, C., and Fan, J. (2009), "Sparsistency and Rates of Convergence in Large Covariance Matrix Estimation," *The Annals of Statistics*, 37, 4254–4278. [298]

Lam, W., and Bacchus, F. (1994), "Learning Bayesian Belief Networks: An Approach Based on the MDL Principle," *Computational Intelligence*, 10, 269–293. [288]

Lauritzen, S. L. (1996), *Graphical Models*, Oxford: Oxford University Press. [293]

Madigan, D., and York, J. (1995), "Bayesian Graphical Models for Discrete Data," *International Statistical Review*, 63, 215–232. [288]

Meinshausen, N., and Bühlmann, P. (2006), "High-Dimensional Graphs and Variable Selection With the Lasso," *The Annals of Statistics*, 34, 1436–1462. [288,292,298]

Pearl, J. (2000), *Causality: Models, Reasoning, and Inference*, New York: Cambridge University Press. [288,289]

Robinson, R. W. (1973), "Counting Labeled Acyclic Digraphs," in *New Directions in the Theory of Graphs*, ed. Haray F., New York: Academic Press, pp. 239–273. [288]

Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., and Nolan, G. P. (2005), "Causal Protein-Signaling Networks Derived From Multiparameter Single-Cell Data," *Science*, 308, 523–529. [297]

Shojaie, A., and Michailidis, G. (2010), "Penalized Likelihood Methods for Estimation of Sparse High-Dimensional Directed Acyclic Graphs," *Biometrika*, 97, 519–538. [289,294,296,297]

Spirtes, P., Glymour, C., and Scheines, R. (1993), *Causation, Prediction, and Search*, New York: Springer. [288]

Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society*, Series B, 58, 267–288. [290]

Tseng, P. (2001), "Convergence of a Block Coordinate Descent Method for Nondifferentiable Minimization," *Journal of Optimization Theory and Applications*, 109, 475–494. [292]

Vandenberghe, L., Boyd, S., and Wu, S.-P. (1998), "Determinant Maximization With Linear Matrix Inequality Constraints," *SIAM Journal on Matrix Analysis and Applications*, 19, 499–533. [288]

Wu, T., and Lange, K. (2008), "Coordinate Descent Procedures for Lasso Penalized Regression," *The Annals of Applied Statistics*, 2, 224–244. [290]

Yuan, M., and Lin, Y. (2007), "Model Selection and Estimation in the Gaussian Graphical Model," *Biometrika*, 94, 19–35. [288]

Zhou, Q. (2011), "Multi-Domain Sampling With Applications to Structural Inference of Bayesian Networks," *Journal of the American Statistical Association*, 106, 1317–1330. [288,298]

Zou, H. (2006), "The Adaptive Lasso and Its Oracle Properties," *Journal of the American Statistical Association*, 101, 1418–1429. [290]