Penalized Estimation of Sparse Directed Acyclic Graphs From Categorical Data Under Intervention

Fei Fu and Qing Zhou†
Department of Statistics, University of California, Los Angeles, CA 90095, USA

Abstract

We develop in this article a penalized likelihood method to estimate sparse causal Bayesian networks from categorical data under experimental intervention. The structure of a Bayesian network is represented by a directed acyclic graph (DAG). We model causal interactions in a discrete network by the multi-logit regression and achieve structure estimation of a DAG via maximizing a regularized likelihood. The adaptive group lasso penalty is employed to encourage sparsity by selecting grouped dummy variables encoding the level of a factor together. We develop a blockwise coordinate descent algorithm to solve the penalized likelihood problem subject to the acyclicity constraint of a DAG. We apply our method to three simulated networks and a real biological network, and demonstrate that our method shows very competitive performance compared to existing methods.

KEY WORDS: Adaptive group lasso; Coordinate descent; Discrete Bayesian network; Multi-logit regression; Structure learning.

1 Introduction

A Bayesian network is a probabilistic graphical model whose structure encodes the conditional independence relationship among a set of random variables. It is graphically represented by a directed acyclic graph (DAG). Recent years have seen its popularity in biological and medical sciences for inferring gene regulatory networks and cellular networks, partially attributed to

---

*This work was supported by NSF grants DMS-1055286 and DMS-1308376.
†To whom correspondence should be addressed (email: zhou@stat.ucla.edu).
the fact that it can be used for causal inference. Uncovering the structure of these biological networks from data is a key to understanding their functions. Most methods that have been proposed for structure learning of DAGs fall into two categories. The first category encompasses the so-called constraint-based methods that rely on a set of conditional independence tests. The PC algorithm proposed by Spirtes et al. (1993) is a well-known example. The second category includes scoring-based methods whose goal is to search for a DAG that maximizes certain scoring function. The scoring functions that have been employed include several Bayesian Dirichlet metrics (Buntine 1991; Cooper and Herskovits 1992; Heckerman et al. 1995), Bayesian information criterion (Chickering and Heckerman 1997), minimum description length (Bouckaert 1993; Suzuki 1993; Bouckaert 1994; Lam and Bacchus 1994), entropy (Herskovits and Cooper 1990), et cetera.

With the rising interest in sparse statistical modeling, scoring-based methods seem particularly attractive since various sparse regularization techniques are potentially applicable to them. Assuming a given natural ordering among the nodes, Shojaie and Michailidis (2010) decomposed DAG estimation into a sequence of $\ell_1$-penalized linear regression problems. Fu and Zhou (2013) recently developed an $\ell_1$-penalized likelihood approach to structure estimation of sparse DAGs from Gaussian data without assuming a given ordering. This method has been further generalized to the use of concave penalties by Aragam and Zhou (2014). There is also theoretical development on $\ell_0$-penalized estimation of sparse high-dimensional DAGs under a multivariate Gaussian model (van de Geer and Bühlmann 2013).

Despite the recent fast developments on sparse regularization methods for learning Gaussian DAGs, a generalization to categorical data is highly nontrivial. First, each node now represents a factor coded by a group of dummy variables. In order to select a group of dummy variables together, we need to use the group lasso penalty instead of penalizing individual coefficients. Second, the log-likelihood for categorical data is more complicated than that for Gaussian data, and consequently, development of an algorithm to maximize the penalized log-likelihood becomes much more challenging. In this paper, we propose a principled generalization of the penalized likelihood methodology in our previous work (Fu and Zhou 2013) to estimate sparse DAGs from categorical data without knowing the ordering among variables.
The remainder of this paper is organized as follows. Section 2 gives a brief introduction to causal discrete Bayesian networks and describes a multi-logit model that we propose to formulate the structure learning problem. Section 3 develops a blockwise coordinate descent algorithm used for parameter estimation, where each coordinate descent step is achieved by applying a quadratic approximation iteratively. Section 4 establishes asymptotic properties of our penalized estimator. Section 5 reports numerical results of our method on three types of simulated networks, and Section 6 presents results on a biological data set. The paper is concluded with discussion in Section 7. Several technical proofs are relegated to the Appendix.

2 Problem Formulation

2.1 Causal discrete Bayesian networks

The structure of a Bayesian network with $p$ random variables $X_1, \ldots, X_p$ is often visualized as a DAG $\mathcal{G} = (V, E)$. The set of nodes $V = \{1, \ldots, p\}$ represents the set of random variables $\{X_1, \ldots, X_p\}$, and the set of edges $E = \{(j, i) \in V \times V : j \to i \text{ is a directed edge in } \mathcal{G}\}$. Given the structure of $\mathcal{G}$, the joint probability distribution of $\{X_1, \ldots, X_p\}$ can be factorized as

$$P(X_1, \ldots, X_p) = \prod_{i=1}^{p} P(X_i | \Pi^\mathcal{G}_i),$$

(1)

where $\Pi^\mathcal{G}_i = \{X_j : j \in V \text{ and } (j, i) \in E\}$ is called the set of parents of $X_i$.

Given a joint distribution, there may exist multiple factorizations of the form in (1), leading to different DAGs. The DAGs encoding the same set of joint distributions form an equivalence class of Bayesian networks. Therefore, it may not be possible to distinguish equivalent DAGs from observational data alone. However, when used for causal inference, equivalent DAGs do not have the same causal interpretation and can be differentiated based on experimental data. We describe below how the joint distribution of a Bayesian network can be modified to incorporate experimental data. For a detailed account of causal inference using Bayesian networks, please refer to Pearl (2000) and references therein. Briefly, assuming $X_i, i \in \mathcal{M}$, is under experimental
intervention, the joint distribution in (1) becomes

\[
P(X_1, \ldots, X_p) = \prod_{i \notin M} P(X_i|\Pi_i^G) \prod_{i \in M} P(X_i|\bullet),
\]

where \(P(X_i|\bullet)\) denotes the distribution of \(X_i\) under intervention. Experimental data generated from \(G\) can therefore be considered as being generated from the DAG \(G'\) obtained by removing all directed edges in \(G\) pointing to the variables under intervention. In this paper, we focus on estimating causal discrete Bayesian networks from experimental data. But it should be noted that (2) also applies to observational data for which \(M\) is simply empty.

In a discrete Bayesian network, each variable \(X_i\) is considered a factor with \(r_i\) levels, indexed by \(\{1, \ldots, r_i\}\). The set of its parents \(\Pi_i^G\) has a total of \(q_i = \prod_{j \in \Pi_i^G} r_j\) possible joint states \(\{\pi_k : k = 1, \ldots, q_i\}\). Let \(\Theta_{ijk} = P(X_i = j | \Pi_i^G = \pi_k)\). A discrete Bayesian network \(G\) may be parameterized by \(\Theta = \{\Theta_{ijk} \geq 0 : \sum_j \Theta_{ijk} = 1\}\) via a product multinomial model given the graph structure. Denote by \(\text{supp}(\mathbf{v})\) the set of nonzero components of a vector \(\mathbf{v}\). The number of parameters in this product multinomial model

\[
|\text{supp}(\Theta)| = \sum_{i=1}^{p} r_i q_i = \sum_{i=1}^{p} r_i \prod_{X_j \in \Pi_i^G} r_j.
\]

If we assume that every variable has the same number of levels, i.e., \(r_i = r\) for all \(i\), then

\[
|\text{supp}(\Theta)| = \sum_{i=1}^{p} r^{1+|\Pi_i^G|},
\]

which grows exponentially as the size of the parent set \(|\Pi_i^G|\) increases. To reduce the number of free parameters, we propose a multi-logit model for discrete Bayesian networks under which development of a penalized likelihood method is straightforward. For the same DAG structure, the number of parameters can be much smaller compared to the product multinomial model.
2.2 A multi-logit model

We encode the $r_i$ levels of $X_i$, $i = 1, \ldots, p$, by a group of $d_i = r_i - 1$ dummy variables. Suppose that we have a data set $X = (X_{hi})_{n \times p}$ generated from a causal discrete Bayesian network $G$, where $X_{hi}$ is the level of $X_i$ in the $h$th data point, $h = 1, \ldots, n$. Let $x_{h,i} \in \{0, 1\}^{d_i}$ be the group of dummy variables for $X_{hi}$ and $x_h = (1, x_{h,1}^T, \ldots, x_{h,p}^T)^T \in \{0, 1\}^d$, where $d = 1 + \sum_{i=1}^p d_i$. For a discrete Bayesian network $G$, we propose to model the conditional distribution $[X_j|\Pi_j]$ using the following multi-logit model

$$p_{j\ell}(x_h) \triangleq P(X_j = \ell|x_h) = \frac{\exp(\beta_{j0} + \sum_{i=1}^p x_{h,i}^T \beta_{j\ell i})}{\sum_{m=1}^{r_j} \exp(\beta_{jm0} + \sum_{i=1}^p x_{h,i}^T \beta_{jmi})} = \frac{\exp(x_h^T \beta_{j0} \cdot \ell)}{\sum_{m=1}^{r_j} \exp(x_h^T \beta_{jm} \cdot \ell)}, \; \ell = 1, \ldots, r_j,$$

(4)

where $\beta_{j0}$ is the intercept, $\beta_{j\ell_i} \in \mathbb{R}^{d_i}$ is the coefficient vector for $X_i$ to predict the $\ell$th level of $X_j$, and $\beta_{j\ell} = (\beta_{j0}, \beta_{j1}^T, \ldots, \beta_{j\ell_p}^T)^T \in \mathbb{R}^d$. Note that in the multi-logit model above, $\beta_{j\ell_i} = 0$ for all $\ell$ if $i \notin \Pi_j^G$. We choose to use a symmetric form of the multi-logit model here, as was done in [Zhu and Hastie (2004)] and [Friedman et al. (2010)]. To make this model identifiable, we impose the following constraints on the intercepts

$$\beta_{j10} = 0, \quad j = 1, \ldots, p.$$  

(5)

The nonidentifiability of other parameters can be resolved via regularization as demonstrated by [Friedman et al. (2010)]. The particular form of regularization we use leads to the following constraints

$$\sum_{m=1}^{r_j} \beta_{jmi} = 0, \quad \forall i, j = 1, \ldots, p.$$  

(6)

We define $\beta_{j,i} = (\beta_{j1i}^T, \ldots, \beta_{j\ell_p i})^T \in \mathbb{R}^{d_{r_j}}$ to be the vector of coefficients representing the influence of $X_i$ on $X_j$ and $\beta_{j,0} = (\beta_{j10}, \ldots, \beta_{jr_j0})^T \in \mathbb{R}^{r_j}$ to be the vector of intercepts for predicting $X_j$. Let $\beta = (\beta_{1,0}^T, \beta_{1,1}^T, \ldots, \beta_{1,p}^T, \beta_{p,0}^T, \beta_{p,1}^T, \ldots, \beta_{p,p}^T)^T$ be the vector of all parameters. Given the structure of $G$, the number of nonzero elements of $\beta$ is given by
\[ \sum_{j=1}^{p} \left[ (r_j - 1) + r_j \sum_{i \in \Pi_j^G} d_i \right] \] If we further assume that \( r_i = r \) for all \( i \),

\[ |\text{supp}(\beta)| = (r - 1) [r|E| + p], \tag{7} \]

which grows linearly in the total number of edges \( |E| \). This rate of growth is much slower compared to that of the product multinomial model (3).

In this paper, we assume the data set \( \mathcal{X} \) consists of \( p \) subsets, \( \mathcal{X}^j \) with dimension \( n_j \times p \), \( j = 1, \ldots, p \), where the level of \( X_j \) is experimentally fixed in each row of \( \mathcal{X}^j \). Let \( \mathcal{I}_j \) be the set of row indices of \( \mathcal{X}^j \) and \( \mathcal{O}_j = \{1, \ldots, n\} \setminus \mathcal{I}_j \) be the index set of rows in which \( X_j \) is observational. Denote the size of \( \mathcal{O}_j \) by \( n_{-j} = |\mathcal{O}_j| = n - n_j \). Under the multi-logit model (4), the log-likelihood function \( \ell(\beta) \) can be written according to the factorization (2) as

\[ \ell(\beta) \propto \sum_{j=1}^{p} \sum_{h \in \mathcal{O}_j} \log \left( P(X_{hj} | x_{h,i}, i \in \Pi_j^G) \right) \]

\[ = \sum_{j=1}^{p} \sum_{h \in \mathcal{O}_j} \left[ r_j \sum_{\ell=1}^{r_j} y_{h\ell} x_{h,\ell}^T \beta_{j\ell} - \log \left( \sum_{m=1}^{r_j} \exp(x_{h,\ell}^T \beta_{jm}) \right) \right], \tag{8} \]

where \( y_{h\ell} = I(X_{hj} = \ell) \) are indicator variables and \( \beta_{j\ell k} = 0 \) for \( k /\in \Pi_j^G \).

**Remark 1.** Although we have assumed the availability of experimental data, it is easy to see that the log-likelihood (8) applies to observational data as well: If there are no experimental data for \( X_j \), then \( \mathcal{O}_j = \{1, \ldots, n\} \) in (8).

### 2.3 Group lasso penalty

Estimating the structure of a discrete Bayesian network \( \mathcal{G} \) is equivalent to estimating the sparsity pattern of the parameter \( \beta \), due to the following equivalence

\[ \beta_{ji} = 0 \iff i /\in \Pi_j^G. \tag{9} \]

In order to learn a sparse DAG from data, we propose a penalized likelihood approach to estimate \( \beta \). It can be seen from (9) that, for discrete Bayesian networks, the set of parents of
$X_j$ is given by the set $\{i : \beta_{j,i} \neq 0\}$. The regular lasso penalty on $\beta$ is inappropriate for this purpose since it penalizes each component of $\beta$ separately. We instead wish to penalize $\beta_{j,i}$ as a whole to obtain a sparse DAG. Yuan and Lin (2006) proposed the group lasso penalty to select grouped variables (or factors) in linear regression. Subsequently, Kim et al. (2006) extended the group lasso to general loss functions and Meier et al. (2008) developed an alternative algorithm for group lasso penalized logistic regression. However, similar to the regular lasso penalty, the group lasso penalty suffers certain drawbacks such as inconsistent variable selection under certain circumstances. The adaptive group lasso penalty was developed to overcome the limitation of the group lasso (Wang and Leng 2008; Bach 2008; Wei and Huang 2010). Let $G_{\beta}$ denote the graph induced by $\beta$ so that $\Pi_{G_{\beta}} = \{i : \beta_{j,i} \neq 0\}$ for $j = 1, \ldots, p$. We propose the following adaptive group lasso estimator for learning the structure of causal discrete Bayesian networks:

$$
\hat{\beta}_\lambda = \arg \min_{\beta : G_{\beta} \text{ is acyclic}} \left[ R_\lambda(\beta) \triangleq -\ell(\beta) + \lambda \sum_{j=1}^{p} \sum_{i=1}^{p} w_{ji} \|\beta_{j,i}\|_2 \right],
$$

where $W = (w_{ji})_{p \times p}$ is a given weight matrix and $\lambda > 0$ is a tuning parameter. See Sections 3.4 and 3.5 for choosing the weights $w_{ji}$ and the parameter $\lambda$. Hereafter, we call $\beta_{j,i}$ a (component) group of $\beta$.

### 3 Algorithm

Parameter estimation for discrete Bayesian networks is computationally demanding because of the nonlinear nature of the multi-logit model (4). We develop in this section a blockwise coordinate descent algorithm to solve (10). Coordinate descent type of algorithms have been proved successful in various settings (Fu 1998; Friedman et al. 2007; Wu and Lange 2008) and their implementations are relatively straightforward.
3.1 Single coordinate descent step

We first consider minimizing $R_\lambda(\beta)$ \((10)\) with respect to $\beta_{..}$ given the current estimates of all the other parameters. We define

$$R_{\lambda,j}(\beta_{..}) = -\sum_{h \in O_j} \left[ \sum_{\ell=1}^{r_j} y_{hj\ell} x_h^T \beta_{..} - \log \left( \sum_{m=1}^{r_j} \exp(x_h^T \beta_{jm} \cdot) \right) \right] + \lambda \sum_{k=1}^{p} w_{jk} \| \beta_{jk} \|_2$$

$$\Delta = -\ell_j(\beta_{..}) + \lambda \sum_{k=1}^{p} w_{jk} \| \beta_{jk} \|_2,$$

\((11)\)

where $\beta_{..} = (\beta^T_{..}, \beta^T_{..1}, \ldots, \beta^T_{..p})^T$. For simpler notation, we suppress the dependence of $R_{\lambda,j}$ and $\ell_j$ on $\beta_{..}$ when considering the problem of minimizing $R_{\lambda,j}(\cdot)$ over $\beta_{..}$. Adding the penalty, the quadratic approximation is

$$Q_{\lambda,j}^{(t)}(\beta_{..}) = -\left\{ (\beta_{ji} - \beta_{ji}^{(t)})^T \nabla \ell_j(\beta_{ji}^{(t)}) + \frac{1}{2} (\beta_{ji} - \beta_{ji}^{(t)})^T H_{ji}^{(t)} (\beta_{ji} - \beta_{ji}^{(t)}) \right\} + \lambda w_{ji} \| \beta_{ji} \|_2,$$

\((12)\)

up to an additive term that does not depend on $\beta_{..}$. The gradient of the log-likelihood function $\ell_j(\cdot)$ is

$$\nabla \ell_j(\beta_{ji}^{(t)}) = \sum_{h \in O_j} \begin{pmatrix} (y_{hj1} - p_{j1}^{(t)}(x_h)) x_{hi} \\ \vdots \\ (y_{hjr_j} - p_{jr_j}^{(t)}(x_h)) x_{hi} \end{pmatrix},$$

\((13)\)

where $p_{j\ell}^{(t)}(x_h)$ defined in \([4] \), $\ell = 1, \ldots, r_j$, are evaluated at the current parameter values. To give a reasonable quadratic approximation, we use a negative definite matrix $H_{ji}^{(t)} = h_{ji}^{(t)} I_{d_ir_j}$ in \((12)\) to approximate the Hessian of $\ell_j(\cdot)$, where the scalar $h_{ji}^{(t)} < 0$ and $I_{d_ir_j}$ is the identity matrix of size $d_ir_j \times d_ir_j$. As suggested by Tseng and Yun (2009) and Meier et al. (2008), we
choose

\[ h_{ji}^{(t)} = -\max\{\text{diag}(-H_{ji}(\beta_j^{(t)})), b\}, \]

where \( H_{ji} \) is the Hessian of the log-likelihood function \( \ell_j(\cdot) \) and \( b \) is a small positive number used as a lower bound to help convergence. Note that it is not necessary to recompute \( h_{ji}^{(t)} \) every iteration (Meier et al. 2008).

It is not difficult to show the following proposition, which is a direct consequence of the Karush-Kuhn-Tucker (KKT) conditions for minimizing (12).

**Proposition 1.** Let \( H_{ji}^{(t)} = h_{ji}^{(t)} I_{d,r_j} \) for some scalar \( h_{ji}^{(t)} < 0 \) and \( d_{ji}^{(t)} = \nabla \ell_j(\beta_j^{(t)}) - h_{ji}^{(t)} \beta_j^{(t)} \).

The minimizer of \( Q_\lambda,\beta^{(t)}(\beta_j) \) in (12) is

\[
\beta_{j,i}^{(t+1)} = \begin{cases} 
0 & \text{if } \|d_{ji}^{(t)}\|_2 \leq \lambda w_{ji}, \\
-\frac{1}{h_{ji}^{(t)}} \left[ d_{ji}^{(t)} - \lambda w_{ji} d_{ji}^{(t)} \|d_{ji}^{(t)}\|_2 \right] & \text{otherwise.}
\end{cases}
\] (14)

**Remark 2.** It follows from Proposition 1 with \( w_{j0} = 0 \) that for the unpenalized intercepts

\[
\beta_{j,0}^{(t+1)} = -\frac{1}{h_{j0}^{(t)}} d_{j0}^{(t)}.
\] (15)

In addition, some of the parameters are always constrained to zero, e.g., \( \beta_{j,j} \) and \( \beta_{j10} \) for all \( j \). They should not be updated.

In order to minimize \( R_{\lambda,\beta}(\cdot) \) with respect to \( \beta_{j,i} \), we apply the quadratic approximation iteratively until some stopping criterion is met.

### 3.2 Blockwise coordinate descent

Due to acyclicity, we know a priori that \( \beta_{i,j} \) and \( \beta_{j,i} \) cannot simultaneously be nonzero for \( i \neq j \). This suggests performing the minimization in blocks, minimizing over \( \{\beta_{i,j}, \beta_{j,i}\} \) simultaneously. An immediate consequence of this is a substantial reduction in the number of free parameters.

In order to enforce acyclicity, we use a simple heuristic (Fu and Zhou 2013): For each block \( \{\beta_{i,j}, \beta_{j,i}\} \), we check to see if adding an edge from \( X_i \rightarrow X_j \) induces a cycle in the estimated
DAG. If so, we set $\beta_{j,i} = 0$ and minimize with respect to $\beta_{k,j}$. Alternatively, if the edge $X_j \rightarrow X_i$ induces a cycle, we set $\beta_{k,j} = 0$ and minimize with respect to $\beta_{j,i}$. If neither edge induces a cycle, we minimize over both parameters simultaneously.

We outline below (Algorithm 1) the complete blockwise coordinate descent (CD) algorithm for discrete Bayesian networks. In the algorithm, $\beta_{j,i} \leftarrow 0$ is used to indicate that given current estimates of other parameters, $\beta_{j,i}$ must be set to zero due to the acyclicity constraint. Minimization of $R_{\lambda,j}()$ with respect to $\beta_{j,i}$ is done with the method presented in Section 3.1.

Algorithm 1 CD algorithm for estimating discrete Bayesian networks

1: Initialize $\beta$ such that $\mathcal{G}_\beta$ is acyclic
2: for $i = 1, \ldots, p - 1$ do
3:   for $j = i + 1, \ldots, p$ do
4:     if $\beta_{j,i} \leftarrow 0$ then
5:       $\beta_{i,j} \leftarrow \arg\min_{\beta_{i,j}} R_{\lambda,i}()$, $\beta_{j,i} \leftarrow 0$
6:     elseif $\beta_{k,j} \leftarrow 0$ then
7:       $\beta_{k,j} \leftarrow 0$, $\beta_{j,i} \leftarrow \arg\min_{\beta_{j,i}} R_{\lambda,j}()$
8:     else
9:       $S_1 \leftarrow \min_{\beta_{i,j}} R_{\lambda,i}() + R_{\lambda,j}()|_{\beta_{i,j}=0}$
10:      $S_2 \leftarrow R_{\lambda,i}()|_{\beta_{i,j}=0} + \min_{\beta_{j,i}} R_{\lambda,j}()$
11:      if $S_1 \leq S_2$ then
12:        $\beta_{i,j} \leftarrow \arg\min_{\beta_{i,j}} R_{\lambda,i}()$, $\beta_{j,i} \leftarrow 0$
13:      else
14:        $\beta_{i,j} \leftarrow 0$, $\beta_{j,i} \leftarrow \arg\min_{\beta_{j,i}} R_{\lambda,j}()$
15:     end if
16:   end if
17: end for
18: end for
19: Update intercepts $\beta_{j,0}$ by (15) for $j = 1, \ldots, p$
20: Repeat step 2 to 19 until some stopping criterion is met

Although the objective function (10) is convex, our search space is nonconvex due to the acyclicity constraint. Rigorous theory on convergence of a CD algorithm over a nonconvex set is not available. Therefore, we run Algorithm 1 until $\|\beta^t - \beta^{t-1}\|_2 \leq \epsilon$ or $t \geq M$, where $\epsilon > 0$ is a small number and $M$ is the maximal number of iterations. If $\|\beta^M - \beta^{(M-1)}\|_2 > \epsilon$ at the last iteration, one may re-run the algorithm with a different initial value and/or with a different ordering to cycle through the coefficient blocks. For all the examples we have tested, the CD algorithm showed no problems in convergence.
3.3 Solution path

We use Algorithm 1 to compute $\hat{\beta}_\lambda$ over a grid of $J$ values for the tuning parameter, $\lambda_1 > \ldots > \lambda_J > 0$, where at $\lambda_1$ every parameter other than the intercepts is penalized to zero. Let $1_{n-j}$ be the vector of ones of length $n-j$, $y_{j\ell} = (y_{hj\ell})_{h \in \mathcal{O}_j} \in \mathbb{R}^{n-j}$, and $\bar{y}_{j\ell} = \bar{y}_{j\ell}1_{n-j}$ where $\bar{y}_{j\ell}$ is the mean of $y_{j\ell}$. Define $X_{ji}$ to be an $n-j \times d_i$ matrix whose rows are composed of $x_{h,i}^T$ for $h \in \mathcal{O}_j$. It follows from the KKT conditions for (10) that

$$
\lambda_1 = \max_{1 \leq i,j \leq p} \frac{1}{w_{ji}} \left\| \nabla \ell_j(\beta_{j\cdot i})_{\beta_{j\cdot i}=0} \right\|_2
= \max_{1 \leq i,j \leq p} \frac{1}{w_{ji}} \left\| \begin{pmatrix}
X_{ji}^T(y_{j1} - \bar{y}_{j1}) \\
\vdots \\
X_{ji}^T(y_{jr_j} - \bar{y}_{jr_j})
\end{pmatrix} \right\|_2,
$$

(16)

where $\beta_{j\cdot0}$ are set to the maximum likelihood estimates (MLEs) of the intercepts assuming all $\beta_{j\cdot i}$, $i = 1, \ldots, p$, are zero.

The solution $\hat{\beta}_{\lambda_t}$ is used as a warm start for calculating $\hat{\beta}_{\lambda_{t+1}}$, $t = 1, \ldots, J-1$. To further speed up computation, we do not cycle through all blocks of parameters in each coordinate descent cycle. Instead, we only iterate over the current active set until the stopping criterion is reached. The algorithm stops if another full coordinate descent cycle does not alter the active set. Otherwise, the process is repeated.

3.4 Choosing adaptive weights

We choose the weights $w_{ji} = \|\hat{\beta}_{j\cdot i}\|_2^{-\gamma}$ for some $\gamma > 0$, where $\hat{\beta}_{j\cdot i}$ is a consistent estimate of $\beta_{j\cdot i}$. In the generalized linear model setting with a fixed $p$, one may use the MLEs to define the weights. However, because of the existence of equivalent DAGs, the MLEs may not be consistent in our case. In this work, we construct the initial consistent estimates $\tilde{\beta}_{j\cdot i}$ by minimizing $R_\lambda(\beta)$ in (10) with weights $w_{ji} = 1$ for all $i \neq j$, i.e., $\tilde{\beta}_{j\cdot i}$ is simply the group lasso estimate. As will be shown in Section 4 our penalized estimator $\hat{\beta}_\lambda$ defined by (10) with weights $w_{ji} = \|\hat{\beta}_{j\cdot i}\|_2^{-\gamma}$ has desired asymptotic properties.
3.5 Tuning parameter selection

Traditional model selection criteria such as cross-validation and BIC do not work well for the purpose of estimating DAGs from data, because the DAG with the smallest prediction error often has much more edges than the true DAG. In order to select a suitable tuning parameter $\lambda_t$, we use an empirical model selection criterion proposed by Fu and Zhou (2013). Let $\hat{G}_{\lambda_t}$ be the DAG induced by $\hat{\beta}_{\lambda_t}$ and $e_{\lambda_t}$ be the number of directed edges in $\hat{G}_{\lambda_t}$. We reestimate $\beta$ by the maximizer $\hat{\beta}^\dagger_{\lambda_t}$ of the log-likelihood $\ell(\beta)$ (8) given $G = \hat{G}_{\lambda_t}$. Regressing $X_j$ on $\Pi^G_j$ is done using the R package mlogit (Croissant 2011). We define the difference ratio between two estimated DAGs $\hat{G}_{\lambda_t}$ and $\hat{G}_{\lambda_{t+1}}$ by 

$$dr(t, t+1) = \frac{\Delta \ell(t, t+1)}{\Delta e(t, t+1)},$$

where $\Delta \ell(t, t+1) = \ell(\beta^T_{\lambda_{t+1}}) - \ell(\beta^T_{\lambda_t})$ and $\Delta e(t, t+1) = e_{\lambda_{t+1}} - e_{\lambda_t}$, provided that $\Delta e(t, t+1) \geq 1$. Otherwise, we set $dr(t, t+1) = dr(t-1, t)$. The selected tuning parameter is indexed by

$$T = \sup \{2 \leq t \leq J : dr(t-1, t) \geq \alpha \cdot \max\{dr(1, 2), \ldots, dr(J-1, J)\}\}.$$  

(17)

The typical value of the thresholding parameter $\alpha$ is 0.1. According to this criterion, an increase in model complexity, measured by the number of predicted edges, is accepted only if there is a substantial increase in the log-likelihood.

4 Asymptotic Properties

In this section we establish asymptotic theory for the adaptive group lasso estimator $\hat{\beta}_\lambda$ assuming that $p$ is fixed. By rearranging and relabeling individual components, we rewrite $\beta$ as $\phi = (\phi^T_{(1)}, \phi^T_{(2)})^T$, where $\phi_{(1)} = (\beta^T_{1,1}, \ldots, \beta^T_{1,p}, \ldots, \beta^T_{p,1}, \ldots, \beta^T_{p,p})^T$ is the parameter vector of interest and $\phi_{(2)} = (\beta^T_{1,0}, \ldots, \beta^T_{p,0})^T$ denotes the vector of intercepts. Hereafter, we denote by $\phi_j$ the $j^{th}$ group of $\phi$, such that $\phi_1 = \beta_{1,1}$, $\phi_2 = \beta_{1,2}$, $\ldots$, $\phi_{p,2} = \beta_{p,2}$, and so on. We concatenate the rows of the weight matrix $W$ into a vector $T = (\tau_j)_{1:p^2} = (w_{11}, \ldots, w_{1p}, \ldots, w_{p1}, \ldots, w_{pp})^T$. We say $\phi$ is acyclic if the graph $G_\phi$ induced by $\phi$ (or the corresponding $\beta$) is acyclic. Let $\Omega = \{\phi : \phi$ is acyclic and $\phi$ satisfies [5] and [6]\} denote the parameter space, $\phi^* \in \Omega$ be the true parameter, and $G_{\phi^*}$ be the true DAG. Let $A = \{j : \phi^*_j = 0, 1 \leq j \leq p^2\}$ and
Define \( \phi_{[k]} (k \in \{1, \ldots, p\}) \) to be the parameter vector obtained from \( \phi \) by setting \( \beta_{k,i} = 0 \) for \( i = 1, \ldots, p \). In other words, the DAG \( \mathcal{G}_{\phi_{[k]}} \) is obtained by deleting all edges pointing to the \( k \)th node in \( \mathcal{G}_\phi \). As demonstrated by [2], we can model interventional data in the \( k \)th block of the data matrix \( X^k \) as i.i.d. observations from a joint distribution factorized according to \( \mathcal{G}_{\phi_{[k]}} \). Denote the corresponding probability mass function by \( P(x|\phi_{[k]}) \), where \( x = (x_1, \ldots, x_p) \) and \( x_j \in \{1, \ldots, r_j\} \) for \( j = 1, \ldots, p \). The log-likelihood of the full data matrix \( X \) is

\[
L(\phi) = \sum_{k=1}^{p} L_k(\phi_{[k]}) = \sum_{k=1}^{p} \log P(X^k | \phi_{[k]}),
\]

where \( \log P(X^k|\phi_{[k]}) = \sum_{h \in X_k} \log(P(X_h|\phi_{[k]})) \) and \( X_h = (X_{h1}, \ldots, X_{hp}) \). Then the penalized log-likelihood function with the adaptive group lasso penalty is

\[
R(\phi) = L(\phi) - \lambda_n \sum_{j=1}^{p^2} \tau_j \| \phi_j \|_2 = \sum_{k=1}^{p} L_k(\phi_{[k]}) - \lambda_n \sum_{j=1}^{p^2} \tau_j \| \phi_j \|_2,
\]

where the component group \( \phi_j (j = 1, \ldots, p^2) \) represents the influence of one variable on another. A penalized likelihood estimator \( \hat{\phi} \) is obtained by maximizing \( R(\phi) \) in the parameter space \( \Omega \).

Though interventional data help to distinguish equivalent DAGs, the following notion of natural parameters is used to completely establish identifiability of DAGs for the case where each variable has interventional data. We say that \( X_i \) is an ancestor of \( X_j \) in a DAG \( \mathcal{G} \) if there exists at least one directed path from \( X_i \) to \( X_j \). Denote the set of ancestors of \( X_j \) by \( \text{an}(X_j) \).

**Definition 1** (Natural parameters). We say that \( \phi \in \Omega \) is natural if \( X_i \in \text{an}(X_j) \) in \( \mathcal{G}_\phi \) implies that \( X_j \) is dependent of \( X_i \) according to the joint distribution \( P(x|\phi_{[i]}) \) for all \( i, j = 1, \ldots, p \).

For a causal DAG, a natural parameter implies that the effects along multiple causal paths connecting the same pair of nodes do not cancel, which is a reasonable assumption for many real-world problems. To establish asymptotic properties of our penalized likelihood estimator for fixed \( p \), we make the following assumptions:

(A1) The true parameter \( \phi^* \) is natural and a fixed interior point of \( \Omega \).
(A2) The conditional distribution $[X_j|\Pi^G_j]$, $\phi \in \Omega$, is specified by the multi-logit model (4) with constraints (5) and (6) for $j = 1, \ldots, p$.

Recall that the $k$th block of our data, $X^k$, can be regarded as an i.i.d. sample of size $n_k$ from the distribution $P(x|\phi^*_k)$ for $k = 1, \ldots, p$. Proofs of the following theorems are relegated to the Appendix.

**Theorem 1.** Assume (A1) and (A2). If $P(x|\phi_k) = P(x|\phi^*_k)$ for all possible $x$ and all $k = 1, \ldots, p$, then $\phi = \phi^*$. Furthermore, if $n_k/n \to \alpha_k > 0$, then for any $\phi \neq \phi^*$

$$P(L(\phi^*) > L(\phi)) \to 1 \quad \text{as } n \to \infty.$$  \hfill (20)

**Theorem 2.** Assume (A1), (A2), and the weights $\tau_j = O_p(1)$ in (19) for all $j$. 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 $\hat{\phi}$ of $R(\phi)$ such that $\|\hat{\phi} - \phi^*\|_2 = O_p(n^{-1/2})$ as $n \to \infty$.

**Theorem 3.** Assume (A1) and (A2). Let the weights $\tau_j = \|\tilde{\phi}_j\|_2^{-\gamma}$ in (19) for all $j$, where $\gamma > 0$ and $\tilde{\phi}_j$ is $\sqrt{n}$-consistent for $\phi^*_j$. 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 $\hat{\phi}$ of $R(\phi)$ such that $\|\hat{\phi} - \phi^*\|_2 = O_p(n^{-1/2})$ and $\hat{\phi}_A = (\hat{\phi}_j)_{j \in A} = 0$ with probability tending to one as $n \to \infty$.

**Remark 3.** Theorem 1 confirms that the causal DAG model is identifiable with interventional data assuming a natural parameter. Theorem 2 implies that there is a $\sqrt{n}$-consistent local maximizer of $R(\phi)$ with unweighted group lasso penalty ($\tau_j = 1$, $\forall j$), which in turn can be used to construct adaptive weights to achieve model selection consistency (Theorem 3).

**Remark 4.** It is interesting to generalize the above asymptotic properties for fixed $p$ to the case where $p = p_n$ grows with the sample size $n$, say, by developing nonasymptotic bounds on the $\ell_2$ estimation error $\|\hat{\phi} - \phi^*\|_2$. However, in order to estimate the causal network consistently, sufficient intervention data are needed for each node, i.e., $n_k$ must approach infinity, and thus $p/n \to 0$ as $n \to \infty$. This limits us to the low-dimensional setting with $p < n$. Suppose we have a large network with $p \gg n$. One may first apply some regularization method on
observational data to screen out independent nodes and to partition the network into small subgraphs that are disconnected to each other. Then for each small subgraph, we can afford to generate enough interventional data for each node and apply the method in this paper to infer the causal structure. Our asymptotic theory provides useful guidance for the analysis in the second step.

5 Simulation Study

We evaluated the CD algorithm on simulated data sets. Three types of networks were simulated: a Markov chain, a scale-free network and a small-world network (Figure 1). All three networks have $p = 50$ nodes. Given a network structure, 10 data sets, each of size $n = 500$, were simulated. The size of the $j^{th}$ block of a data set, $X_j^j$ $(1 \leq j \leq p)$, is $n_j = 10$. The value of $X_j$ in each row of $X_j^j$ was randomly fixed to one of its levels regardless of its parents, while all other variables were generated according to the multi-logit model (4). Each variable was assumed to be binary, i.e., $r_j = 2$ for all $j$. In this case, each group of parameters $\beta_{j,i} = (\beta_{j1i}, \beta_{j2i})^T \in \mathbb{R}^2$. If $\Pi_j = \emptyset$, $X_j$ was sampled from its two levels with equal probability. Otherwise, the parameters $\beta_{j,0}$ and $\beta_{j,i}, i \in \Pi_j$, were chosen such that

$$p_{j\ell}(x_h) = \frac{\exp(2 \sum_{i \in \Pi_j} y_{hi\ell})}{\exp(2 \sum_{i \in \Pi_j} y_{hi1}) + \exp(2 \sum_{i \in \Pi_j} y_{hi2})}$$

for $\ell = 1, 2$, where $y_{hi\ell} = I(X_{hi} = \ell)$.

The scale-free network and the small-world network (Figure 1B and Figure 1C) were generated using the R package igraph (Csárdi and Nepusz 2006). The skeleton of the scale-free network in Figure 1B was generated using the Barabási-Albert model (Barabási and Albert 1999). This network has 49 directed edges. The small-world network in Figure 1C was generated using the Watts-Strogatz model (Watts and Strogatz 1998). The graph initially generated by the model was undirected. To convert it to a DAG, edge directions were chosen according to a randomly generated topological sort. See Appendix for the definition of a topological sort. This small-world network has 100 directed edges.
Figure 1: Three simulated DAGs: (A) Markov chain; (B) Scale-free network; (C) Small-world network.

We used the CD algorithm to estimate DAGs from each data set over a grid of tuning parameters, starting from $\lambda_1$ defined in (16). Model selection was done using criterion (17) with $\alpha = 0.1$. For speed consideration, the group lasso weights were all set to 1, as the initial estimates with $w_{ji} = 1$ and the adaptive estimates (with weights specified by the initial estimates) gave comparable DAGs. The accuracy of DAG estimation is measured by true positive rate (TPR) and false discovery rate (FDR), defined as

\[
\begin{align*}
\text{TPR} & = \frac{E}{T}, \\
\text{FDR} & = \frac{R + FP}{P},
\end{align*}
\]

where P, E, R, and FP denote, respectively, the number of predicted edges, the number of expected edges, the number of reversed edges, and the number of false positive edges (excluding the reversed ones) in an estimated DAG, and T denotes the number of edges in the true graph. Results reported in this section are averages over 10 data sets for each of the three simulated networks. Table II summarizes the performance of our CD algorithm. Apparently, both the Markov chain and the scale-free network can be estimated very accurately from data. For the small-world network, which has a denser topology, DAGs estimated by the CD algorithm have lower TPRs on average. This is primarily because about one third of the predicted edges have wrong directions. However, the estimated networks have few false positive edges, suggesting
Table 1: Average results for DAGs learned by the CD algorithm from simulated data

| Network         | T   | P   | E    | R    | M   | FP  | TPR     | FDR     |
|-----------------|-----|-----|------|------|-----|-----|---------|---------|
| Markov chain    | 49  | 53.8| 44.0 | 5.0  | 0.0 | 4.8 | 0.898 (0.035) | 0.181 (0.052) |
| Scale-free      | 49  | 52.0| 44.3 | 4.7  | 0.0 | 3.0 | 0.904 (0.048) | 0.146 (0.078) |
| Small-world     | 100 | 89.7| 54.1 | 30.6 | 15.3| 5.0 | 0.541 (0.058) | 0.397 (0.035) |

Note: M reports the number of missed edges. The numbers in parentheses are the standard deviations across 10 data sets.

Kalisch and Bühlmann (2007) considered the problem of estimating Bayesian networks with the PC algorithm (Spirtes et al. 1993) and developed an efficient implementation suitable for estimating sparse high-dimensional DAGs. We chose to use the PC algorithm to benchmark our CD algorithm due to the availability of an efficient implementation and its nice asymptotic properties. 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, called the PC-based method hereafter, 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 intervention data and produce a DAG. In this comparison, however, we simply count 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 result reported for the PC-based method is an upper bound performance one can obtain. To ease comparison, from the solution path generated by the CD algorithm we selected the DAG with the closest match to the number of edges of that learned by the PC algorithm. From the results in Table 2 we conclude that the performance of our CD algorithm is comparable to the upper limit of a PC-based procedure. Note that, for the Markov chain and the scale-free network, directionality of most predicted
Table 2: Performance comparison between the PC-based method and the CD algorithm on simulated data sets

| Network          | PC-based method |          |          |          |          | CD algorithm |          |          |          |
|------------------|-----------------|----------|----------|----------|----------|--------------|----------|----------|----------|
|                  | P   | U   | TPR     | FDR     | P   | TPR     | FDR     |          |          |
| Markov chain      | 49.2 | 46.9 | 0.957 (0.136) | 0.046 (0.139) | 49.2 | 0.898 (0.035) | 0.106 (0.034) |
| Scale-free        | 41.2 | 39.8 | 0.824 (0.032) | 0.019 (0.029) | 46.7 | 0.869 (0.087) | 0.089 (0.042) |
| Small-world       | 64.9 | 15.1 | 0.405 (0.035) | 0.375 (0.058) | 65.1 | 0.415 (0.037) | 0.362 (0.057) |

Note: The column “U” lists the average number of undirected edges for the CPDAGs learned by the PC algorithm. Reported are the averages over 10 data sets with standard deviations given in the parentheses.

edges cannot be determined by the PC algorithm alone (see column “U” in Table 2). This is because the two networks are very sparse and the sizes of their corresponding equivalence classes are large. Consequently, our counting strategy strongly favors the PC-based method for these two cases.

6 A Real Data Example

We consider in this section applying the CD algorithm to a real data set that has been extensively studied. The data set was generated from a flow cytometry experiment conducted by Sachs et al. (2005), who studied a well-known signaling network in human primary CD4+ T-cells of the immune system. This chosen network was perturbed by various stimulatory and inhibitory interventions. Each interventional condition was applied to an individual component of the network. Simultaneous measurements were taken on $p = 11$ proteins and phospholipids of this network from individual cells under each condition. Since three interventions were targeted at proteins that were not measured, samples collected under these conditions were observational. Among the 11 measured components, five proteins and phospholipids were perturbed. The data set contains measurements for $n = 5400$ cells. Each variable has three levels (high, medium and low), and consequently, the size of a component group of $\beta$ is 6 for this data set.

Figure 2A shows the known causal interactions among the 11 measured components of this
signaling network. These causal relationships are well established, and no consensus has been reached on interactions beyond those present in the network. This network structure is often used as the benchmark to assess the accuracy of an estimated network. Therefore, we call it the consensus model. Two estimated networks with 21 and 26 edges, respectively, are shown in Figure 2B and 2C. Both DAGs are qualitatively close to the consensus model. The detailed performance measures are reported in Table 3. As a comparison, we include the DAG estimated by the order-graph sampler (Ellis and Wong 2008) and that by the multi-domain sampler (Zhou 2011). The CD algorithm predicted more expected edges and fewer false positive edges than the order-graph sampler did. Though the DAG estimated by the multi-domain sampler is the best one in terms of TPR, our CD algorithm seems to estimate the skeleton of the network more accurately (E + R) with fewer false positives. Markov chain Monte Carlo (MCMC) methods for DAG estimation often have good performance when the number of nodes $p$ is small, but they do not scale well. Thus, it is comforting to see that our method, which can handle larger networks, shows very competitive performance compared to MCMC methods on this relatively small network.

7 Discussion

We have developed a penalized likelihood framework for estimating sparse discrete Bayesian networks, under a multi-logit model for causal interactions. In order to avoid penalizing sepa-
Table 3: Comparison between the CD algorithm and two other methods on the flow cytometry data set

| Method                        | P  | E  | R  | M  | FP |
|-------------------------------|----|----|----|----|----|
| CD algorithm (21 edges)       | 21 | 10 | 7  | 3  | 4  |
| Order-graph sampler           | 20 | 8  | 4  | 8  | 8  |
| CD algorithm (26 edges)       | 26 | 10 | 9  | 1  | 7  |
| Multi-domain sampler          | 25.9 | 15.55 | 2.05 | 2.4 | 8.3 |

Note: The order-graph sampler result comes from the mean graph (Figure 11) in Ellis and Wong (2008), while the multi-domain sampler result is the average over 20 independent runs (Table 3 in Zhou (2011)).
penalized loss function. Other future directions include studying the consistency of our penalized estimator when the number of nodes $p = p_n$ grows with the sample size $n$ and investigating the use of group concave penalties.

Appendix: Proofs

We first review the definition of a topological sort of a DAG.

**Definition 2 (Topological sorts).** A topological sort of a DAG $G$ is a linear ordering $\sqsubseteq$ of its nodes such that $i \prec j$ in $\sqsubseteq$ if $i \in \Pi^G_j$.

Due to its directed acyclic nature, every DAG has at least one topological sort. If an ordering $\sqsubseteq$ is a topological sort of a DAG $G$, we say that $G$ and $\sqsubseteq$ are compatible. Let $S(G)$ denote the set of all topological sorts of a DAG $G$.

**Lemma A.1.** For any $\phi^0 \in \Omega$, there is a $\delta(\phi^0) > 0$ such that if $\phi \in \Omega$ and $\|\phi - \phi^0\|_2 < \delta(\phi^0)$ then $S(G_\phi) \cap S(G_{\phi^0}) \neq \emptyset$.

**Proof.** If $G_{\phi^0}$ is an empty graph which is compatible with any ordering, the statement holds trivially. Otherwise, let $\delta(\phi^0) = \frac{1}{2} \min_{j: \phi^0_j \neq 0} \|\phi^0_j\|_2 > 0$, where $\phi^0_j$ is the $j$th component group of $\phi^0$ (see Section 4). If $\|\phi - \phi^0\|_2 < \delta(\phi^0)$ and $\phi^0_i \neq 0$ for some $i \in \{1, \ldots, p^2\}$, then $\phi_i \neq 0$ as well, since otherwise $\|\phi - \phi^0\|_2 \geq \|\phi^0_i\|_2 > \delta(\phi^0)$. This implies that every edge in $G_{\phi^0}$ is also an edge in $G_\phi$ and thus $G_{\phi^0}$ and $G_\phi$ have at least one common topological sort.

This lemma shows that every DAG in a sufficiently small neighborhood of $\phi^0$, denoted by $\text{nb}(\phi^0) \subset \Omega$, has a topological sort that is also compatible with $G_{\phi^0}$.

**Proof of Theorem 1**

**Proof.** We prove the first claim by contradiction. Suppose $\phi \neq \phi^*$ and $P(x|\phi_{[k]}) = P(x|\phi^*_{[k]})$ for $k = 1, \ldots, p$. There are two cases to consider depending on the topological sorts of $G_\phi$ and those of $G_{\phi^*}$.

**Case 1:** $S(G_\phi) \cap S(G_{\phi^*}) \neq \emptyset$. Let $\sqsubseteq \in S(G_\phi) \cap S(G_{\phi^*})$, i.e., an ordering compatible with both $G_\phi$ and $G_{\phi^*}$. Assume without loss of generality that in this ordering $i \prec j$
if \( i < j \). Apparently, \( \sqsubset \) is also compatible with \( G_{\phi_{[k]}} \) and \( G_{\phi^*_k} \) for \( k = 1, \ldots, p \). Then we can write \( P(x|\phi_{[k]}) = \Pi_{i=1}^p P(x_i|x_1, \ldots, x_{i-1}, \phi_{[k]}) = \Pi_{i=1}^p P(x_i|\Pi_i G_{\phi_{[k]}}(\phi_{[k]})) \) and \( P(x|\phi^*_k) = \Pi_{i=1}^p P(x_i|\Pi_i \Pi^*_i G_{\phi^*_{[k]}}(\phi^*_{[k]})) \). Since \( P(x|\phi_{[k]}) = P(x|\phi^*_k) \) for all possible \( x \), it follows that \( \Pi_i G_{\phi_{[k]}} = \Pi_i G_{\phi^*_k} \) for all \( i \) and thus \( G_{\phi_{[k]}} = G_{\phi^*_k} \) for all \( k \). However, since \( \phi \neq \phi^* \), there exists some \( k \) such that \( \phi_{[k]} \neq \phi^*_k \). Therefore, there exists a \( k \) such that the common probability mass function \( P(x|\phi_{[k]}) = P(x|\phi^*_k) \), factorized according to a common structure \( G_{\phi_{[k]}} = G_{\phi^*_k} \), can be parameterized by two different parameters \( \phi_{[k]} \) and \( \phi^*_k \). This is impossible since the multi-logit model (4), subject to constraints (5) and (6), for \( X_j|\Pi_j G_{\phi} \) is identifiable.

**Case 2:** \( S(G_{\phi}) \cap S(G_{\phi^*}) = \emptyset \), that is, none of the orderings of \( G_{\phi^*} \) is compatible with \( G_{\phi} \). In this case, there must exist a pair of indices \( (i, j) \) such that in \( G_{\phi^*} \), \( X_i \in an(X_j) \), but in \( G_{\phi} \), \( X_j \) is a non-descendant of \( X_i \). Then \( X_j \) is independent of \( X_i \) in \( P(x|\phi_{[i]}) \), since in \( G_{\phi_{[i]}} \), \( X_i \) has no parents and \( X_j \) is a non-descendant of \( X_i \). However, in \( G_{\phi^*_{[i]}} \), we still have \( X_i \in an(X_j) \). Since \( \phi^* \) is natural, \( X_i \) and \( X_j \) are dependent in \( P(x|\phi^*_{[i]}) \). Therefore, there exists \( 1 \leq i \leq p \) such that \( P(x|\phi_{[i]}) \neq P(x|\phi^*_{[i]}) \), which contradicts our assumption.

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

To prove the second claim \([20]\), note that by the law of large numbers,

\[
\frac{1}{n} \left( L(\phi) - L(\phi^*) \right) = \sum_{k=1}^p \frac{\eta_k}{n} \sum_{h \in I_k} \log \frac{P(X_h|\phi_{[k]})}{P(X_h|\phi^*_{[k]})} \xrightarrow{p} -p \sum_{k=1}^p \alpha_k \mathbb{E}_{\phi_{[k]}} \log \frac{P(Y|\phi_{[k]})}{P(Y|\phi^*_{[k]})},
\]

where \( Y \) is a random vector with probability mass function \( P(x|\phi^*_{[k]}) \). Then the desired result follows immediately using Jensen’s inequality and the first claim proved above.

**Proof of Theorem 2**

Let \( Y \) be a random vector with probability mass function \( P(x|\phi) \) and

\[
I(\phi) = \mathbb{E}_\phi \left\{ \left[ \frac{\partial}{\partial \phi} \log P(Y|\phi) \right] \left[ \frac{\partial}{\partial \phi} \log P(Y|\phi) \right]^T \right\}
\]

22
be the Fisher information matrix.

Consider \( \phi \in \text{nb}(\phi^*) \). By Lemma A.1, \( G_\phi \) and \( G_{\phi^*} \) have a common compatible ordering. If we restrict to the lower dimensional space \( \Omega_{[k]} = \{ \phi_{[k]} : \phi \in \Omega \} \), the same argument applies to an arbitrarily small neighborhood of \( \phi^*_{[k]} \) in this space, that is, \( G_{\phi_{[k]}} \) and \( G_{\phi^*_{[k]}} \) share a compatible ordering. Then it follows from the arguments used in Case 1 in the proof of Theorem 1 that, for \( \phi_{[k]} \in \text{nb}(\phi^*_{[k]}) \backslash \{ \phi^*_{[k]} \} \), \( P(x | \phi_{[k]}) \neq P(x | \phi^*_{[k]}) \) for some non-negligible set of \( x \) and

\[
E_{\phi^*_{[k]}} \left[ \log P(Y | \phi^*_{[k]}) \right] > E_{\phi^*_{[k]}} \left[ \log P(Y | \phi_{[k]}) \right].
\]

This implies that \( I(\phi^*_{[k]}) \) is positive definite for all \( k \).

Let \( u \in \{ u : \phi^* + n^{-1/2} u \in \Omega \} \) and \( u_j \) be its \( j^{\text{th}} \) component group defined in the same way as \( \phi_j \). Further, let \( u_{[k]} \) be the vector defined similarly as \( \phi_{[k]} \) by setting \( u_j = 0 \) if the \( j^{\text{th}} \) group corresponds to an edge pointing to \( X_k \). Note that \( \sum_{k=1}^{p} \| u_{[k]} \|^2 \geq \| u \|^2 \). Let \( \delta^k_{\min} > 0 \) be the minimal eigenvalue of \( I(\phi^*_{[k]}) \) and \( \rho = \min_k (\alpha_k \delta^k_{\min}/2) \). Then

\[
\sum_{k=1}^{p} \frac{\alpha_k}{2} u_{[k]}^T I(\phi^*_{[k]}) u_{[k]} \geq \sum_{k=1}^{p} \frac{\alpha_k}{2} \delta^k_{\min} \| u_{[k]} \|^2 \geq \rho \sum_{k=1}^{p} \| u_{[k]} \|^2 \geq \rho \| u \|^2. \tag{A.2}
\]

Recall that \( B = \{ j : \phi^* j \neq 0, 1 \leq j \leq p^2 \} \). We have

\[
R(\phi^* + n^{-1/2} u) - R(\phi^*) \leq L(\phi^* + n^{-1/2} u) - L(\phi^*) - \lambda_n \sum_{j \in B} \tau_j (\| \phi^* j + n^{-1/2} u_j \|^2 - \| \phi^* j \|^2) \\
\leq \sum_{k=1}^{p} \left[ L_k(\phi^*_{[k]} + n^{-1/2} u_{[k]}) - L_k(\phi^*_{[k]}) \right] + \lambda_n n^{-1/2} \sum_{j \in B} \tau_j \| u_j \|^2, \tag{A.3}
\]

where the last line follows from the triangle inequality. Taylor expansion of \( L_k \) around \( \phi^*_{[k]} \) in
We omit the proof for \( \parallel \).\( \parallel \) Proof of Theorem 3 

prove the second claim, let us first, by permuting the indices, rewrite the parameter \( \phi \) any number of zero groups of \( \phi \) maximizer \( \hat{\phi} \) when \( n \) is sufficiently large. This implies that with probability at least 1\( - \varepsilon \), there exists a sufficiently large \( C \) such that 

\[
P \left( \sup_{\parallel u \parallel_2 = C} R(\phi^* + n^{-1/2} u) < R(\phi^*) \right) \geq 1 - \varepsilon, \tag{A.5}\]

when \( n \) is sufficiently large. This implies that with probability at least \( 1 - \varepsilon \), there exists a local maximizer \( \hat{\phi} \) of \( R(\phi) \) in the ball \( \{ \phi^* + n^{-1/2} u \in \Omega \} \). Hence, for any \( \varepsilon > 0 \), there exists a sufficiently large \( C \) such that 

\[
\parallel \hat{\phi} - \phi^* \parallel_2 = O_p(n^{-1/2}).
\]

Proof of Theorem 3 

We omit the proof for \( \parallel \hat{\phi} - \phi^* \parallel_2 = O_p(n^{-1/2}) \), since it is similar to that of Theorem 2. To prove the second claim, let us first, by permuting the indices, rewrite the parameter \( \phi \) as 

\[
\phi = (\phi_a^T, \phi_b^T)^T = (\phi_a^T, \phi_B^T, \phi_{(2)}^T)^T, \quad \text{where} \quad \phi_a = \phi_A \text{ and } \phi_b = (\phi_B^T, \phi_{(2)}^T)^T. \quad \text{Let } \quad r = |A| \text{ be the number of zero groups of } \phi^*. \quad \text{Now we only need to show that with probability tending to } 1, \text{ for any } \phi_b \text{ satisfying } \parallel \phi_b - \phi_b^* \parallel_2 = O_p(n^{-1/2}) \text{ and any constant } C > 0,
\]

\[
\arg \max_{\phi_a : \parallel \phi_a \parallel_2 \leq C/\sqrt{n}} R((\phi_a^T, \phi_b^T)^T) = 0. \tag{A.6}
\]

To establish (A.6), we again study the behavior of \( R(\phi) \) around the point \( \phi^o = (0^T, \phi_b^T)^T \)
by expanding $L(\phi)$. Let $u = (u_0^T, u_b^T)^T$ such that $\|u\|_2 \leq C$, $u_b = 0$, and $\phi^o + n^{-1/2}u \in \Omega$. Then we have the following result similar to that in the proof of Theorem 2:

$$
R(\phi^o + n^{-1/2}u) - R(\phi^o) = \sum_{k=1}^p \left[ \sqrt{\alpha_k} \frac{\nabla L_k(\phi^o[k])}{\sqrt{n_k}} u_{[k]} \{1 + o_p(1)\} - \frac{\alpha_k}{2} u_{[k]}^T I(\phi^o[k]) u_{[k]} \{1 + o_p(1)\} \right] - \frac{\lambda_n}{\sqrt{n}} n^{\gamma/2} \sum_{j=1}^r \|\sqrt{n}\tilde{\phi}_j\|_2^{-\gamma} \|u_j\|_2, \text{ as } n \to \infty. \quad (A.7)
$$

Note that both the first and the second terms on the right-hand side of Equation (A.7) are on the order of $O_p(1)$ for any fixed constant $C$. Since $\tilde{\phi}_j$ is $\sqrt{n}$-consistent, $\|\sqrt{n}\tilde{\phi}_j\|_2 = O_p(1)$ for $j = 1, \ldots, r$ and the third term is on the order of $\lambda_n n^{(\gamma-1)/2} \to \infty$. Therefore, (A.6) holds, and the proof is complete.

References

Aragam, B. and Zhou, Q. (2014), “Concave penalized estimation of sparse Bayesian networks,” *ArXiv Preprint*, arXiv:1401.0852.

Bach, F. R. (2008), “Consistency of the Group Lasso and Multiple Kernel Learning,” *Journal of Machine Learning Research*, 9, 1179–1225.

Barabási, A.-L. and Albert, R. (1999), “Emergence of Scaling in Random Networks,” *Science*, 286, 509–512.

Bouckaert, R. R. (1993), “Probabilistic Network Construction Using the Minimum Description Length Principle,” in *Symbolic and Quantitative Approaches to Reasoning and Uncertainty: European Conference ECSQARU ’93*, Springer, vol. 747 of *Lecture Notes in Computer Science*, pp. 41–48.

— (1994), “Probabilistic Network Construction Using the Minimum Description Length Principle,” Tech. Rep. RUU-CS-94-27, Department of Computer Science, Utrecht University.
Buntine, W. (1991), “Theory Refinement on Bayesian Networks,” in *Proceedings of the Seventh Annual Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, pp. 52–60.

Chickering, D. M. and Heckerman, D. (1997), “Efficient Approximations for the Marginal Likelihood of Bayesian Networks with Hidden Variables,” *Machine Learning*, 29, 181–212.

Cooper, G. F. and Herskovits, E. (1992), “A Bayesian Method for the Induction of Probabilistic Networks from Data,” *Machine Learning*, 9, 309–347.

Croissant, Y. (2011), “Estimation of Multinomial Logit Models in R: The mlogit Packages,” Available at: http://cran.r-project.org/web/packages/mlogit/vignettes/mlogit.pdf.

Csárdi, G. and Nepusz, T. (2006), “The igraph Software Package for Complex Network Research,” *Proceedings of the International Conference on Complex Systems*.

Ellis, B. and Wong, W. H. (2008), “Learning Causal Bayesian Network Structures from Experimental Data,” *Journal of the American Statistical Association*, 103, 778–789.

Friedman, J., Hastie, T., Höfling, H., and Tibshirani, R. (2007), “Pathwise Coordinate Optimization,” *The Annals of Applied Statistics*, 1, 302–332.

Friedman, J., Hastie, T., and Tibshirani, R. (2010), “Regularization Paths for Generalized Linear Models via Coordinate Descent,” *Journal of Statistical Software*, 33, 1–22.

Fu, F. and Zhou, Q. (2013), “Learning Sparse Causal Gaussian Networks With Experimental Intervention: Regularization and Coordinate Descent,” *Journal of the American Statistical Association*, 108, 288–300.

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

Heckerman, D., Geiger, D., and Chickering, D. M. (1995), “Learning Bayesian Networks: The Combination of Knowledge and Statistical Data,” *Machine Learning*, 20, 197–243.
Herskovits, E. and Cooper, G. (1990), “Kutató: An Entropy-Driven System for Construction of Probabilistic Expert Systems from Databases,” in *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, pp. 54–62.

Kalisch, M. and Bühlmann, P. (2007), “Estimating High-Dimensional Directed Acyclic Graphs with the PC-Algorithm,” *Journal of Machine Learning Research*, 8, 613–636.

Kim, Y., Kim, J., and Kim, Y. (2006), “Blockwise Sparse Regression,” *Statistica Sinica*, 16, 375–390.

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

Meier, L., van de Geer, S., and Bühlmann, P. (2008), “The Group Lasso for Logistic Regression,” *Journal of the Royal Statistical Society. Series B.*, 70, 53–71.

Pearl, J. (2000), *Causality: Models, Reasoning, and Inference*, Cambridge University Press.

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.

Shojaie, A. and Michailidis, G. (2010), “Penalized Likelihood Methods for Estimation of Sparse High-Dimensional Directed Acyclic Graphs,” *Biometrika*, 97, 519–538.

Spirtes, P., Glymour, C., and Scheines, R. (1993), *Causation, Prediction, and Search*, Springer-Verlag.

Suzuki, J. (1993), “A Construction of Bayesian Networks from Databases Based on an MDL Principle,” in *Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence*, pp. 266–273.

Tseng, P. and Yun, S. (2009), “A Coordinate Gradient Descent Method for Nonsmooth Separable Minimization,” *Mathematical Programming B*, 117, 387–423.

van de Geer, S. and Bühlmann, P. (2013), “ℓ0-penalized maximum likelihood for sparse directed acyclic graphs,” *The Annals of Statistics*, 41, 536–567.
Wang, H. and Leng, C. (2008), “A Note on Adaptive Group Lasso,” *Computational Statistics and Data Analysis*, 52, 5277–5286.

Watts, D. J. and Strogatz, S. H. (1998), “Collective Dynamics of ‘Small-World’ Networks,” *Nature*, 393, 440–442.

Wei, F. and Huang, J. (2010), “Consistent Group Selection in High-Dimensional Linear Regression,” *Bernoulli*, 16, 1369–1384.

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

Yuan, M. and Lin, Y. (2006), “Model Selection and Estimation in Regression with Grouped Variables,” *Journal of the Royal Statistical Society. Series B*, 68, 49–67.

Zhou, Q. (2011), “Multi-Domain Sampling with Applications to Structural Inference of Bayesian Networks,” *Journal of the American Statistical Association*, 106, 1317–1330.

Zhu, J. and Hastie, T. (2004), “Classification of Gene Microarrays by Penalized Logistic Regression,” *Biostatistics*, 5, 427–443.