Learning Causal Bayesian Network Structures From Experimental Data

Byron ELLIS and Wing Hung WONG

We propose a method for the computational inference of directed acyclic graphical structures given data from experimental interventions. Order-space Markov chain Monte Carlo, equi-energy sampling, importance weighting, and stream-based computation are combined to create a fast algorithm for learning causal Bayesian network structures.

KEY WORDS: Equi-energy sampling; Flow cytometry; Markov chain Monte Carlo.

1. INTRODUCTION

The Bayesian network (BN) is a class of multivariate statistical models applicable to many areas in science and technology (Beinlich, Suermondt, Chavez, and Cooper 1989; Pearl 1988; Peér 2005; Friedman 2004). In particular, the BN has become popular as an analytical framework in causal studies, where the causal relations are encoded by the structure (or topology) of the network. But despite significant recent progress in algorithm development, the computational inference of network structure remains very much an open challenge in computational statistics that has proven infeasible except in cases with a very small number of variables.

In this article we report computational algorithms for determining BN structures from experimental data. In Section 2 we review the theory of the BN and its use in causal inference. In Section 3 we discuss the order-space sampling approach recently introduced by Friedman and Koller (2003). We show that sampling in order space induces an intrinsic bias in the resulting network structures and present methods to correct this bias. In Section 4 we combine the order graph sampler with a new energy-domain Monte Carlo method to design an efficient algorithm for generating samples of the BN structure from its marginal posterior distribution conditional on experimental data. In particular, to enhance the sampler's ability to cross energy barriers, we developed a new "single-memory" variant of the equi-energy algorithm. We also present a stream-based computation that greatly reduces the complexity of evaluating the posterior score of a structure. In Section 5 we present numerical results to document our method's effectiveness. On random BNs of various sizes, our approach is able to predict causal edges with an area under the receiver operating curve (ROC) (AUC) approaching 95%, whereas the AUC for a direct graph sampler never exceeds 75%. In Section 6 we apply the new algorithm to the study of the signal transduction network in human T cells. The problem is to reconstruct a part of the network topology from polychromatic flow cytometry data generated after selected vertices of the network are experimentally perturbed. Besides demonstrating the utility of our approach, this example is of considerable interest to the emerging field of systems biology. Finally, in Section 7 we discuss directions for future research and possible extensions of the methodology.

Byron Ellis is Senior Fraud Statistician, AdBrite, Inc., San Francisco, CA 94103 (E-mail: *bellis@adbrite.com*). Wing Hung Wong is Professor of Statistics and Professor of Health Research and Policy, Department of Statistics, Stanford University, Stanford, CA 94305 (E-mail: *whwong@stanford.edu*). This research was supported by National Science Foundation grant DMS-0505732 (to W.H.W.). The authors thank Rob Tibshirani for pointing them toward multiparameter flow cytometry as a rich source of data and Trevor Hastie for suggesting the cross-validation approach for assessing the quality of graphs obtained.

2. REVIEW OF BAYESIAN NETWORKS IN CAUSAL INFERENCE

2.1 Graphical Models

In a BN model, the joint distribution of a set of variables $V = \{V_1, \ldots, V_n\}$ is specified by the decomposition

$$P(V) = \prod_{i=1}^{n} P(V_i | \Pi_i^{\mathcal{G}}), \tag{1}$$

where $\Pi_i^{\mathcal{G}}$, a subset of $\{V_1, \ldots, V_n\} \setminus V_i$, is called the parent set of V_i. By using directed edges to connect each variable to its children variables, we can construct a graph $\mathcal{G}$ to represent the structure (or topology) of the network. In (1), the superscript $\mathcal{G}$ makes it explicit that the parent set for V_i is dependent on the structure of the network. Clearly, for (1) to define a joint distribution, a variable cannot serve as a child of its own descendants; that is, $\mathcal{G}$ must be acyclic. Thus the structure graph $\mathcal{G}$ for a BN is a directed acyclic graph (DAG). There have been many excellent treatments of graphical models and conditional independence structures (e.g., Lauritzen and Spiegelhalter 1988; Lauritzen 1996; Pearl 2000b).

In this article we study only the discrete case in which each V_i is a categorical variable taking values in a finite set $\{v_1, \ldots, v_{r_i}\}$. There are $q_i = \prod_{V_j \in \Pi_i} r_j$ possible values for the joint state $\Pi_i^{\mathcal{G}}$ of the parents of V_i. Given the kth joint state of its parents, V_i takes its jth value with probability θ_{ijk}. Thus the BN with a fixed structure $\mathcal{G}$ is parameterized by the set of probabilities $\Theta = \{\theta_{ijk} \geq 0 : \sum_j \theta_{ijk} = 1\}$. Given N independent observations $\mathbb{X} = \{X_1, X_2, \ldots, X_n\}$ from (1), the sufficient statistics for Θ is the set of counts $\{N_{ijk}\}$, where N_{ijk} denotes the number of times that V_i is found in state j with its parent set in state k. For each combination of child i and its parent set state k, the counts $\{N_{ijk}, j = 1, \ldots, r_i\}$ follow a multinomial distribution with $N_{i\cdot k} = \sum_j N_{ijk}$ trials and probability vector $(\theta_{i1k}, \theta_{i2k}, \ldots, \theta_{ir_ik})$, and

$$P(\mathbb{X} | \Theta, \mathcal{G}) = \prod_{i=1}^{n} \prod_{k=1}^{q_i} \theta_{i1k}^{N_{i1k}} \cdots \theta_{ir_ik}^{N_{ir_ik}}. \tag{2}$$

Example 1: A Four-Vertex Bayesian Network. A four-vertex BN, shown in Figure 1, is specified by

$$\begin{aligned} &P(v_1, v_2, v_3, v_4) \\ &\quad = P(V_2 = v_2 | V_3 = v_3, V_4 = v_4) P(V_3 = v_3 | V_4 = v_4) \\ &\qquad \times P(V_4 = v_4 | V_1 = v_1) P(V_1 = v_1). \end{aligned} \tag{3}$$

Journal of the American Statistical Association
June 2008, Vol. 103, No. 482, Theory and Methods
DOI 10.1198/016214508000000193

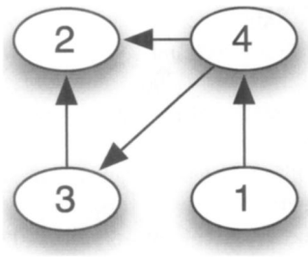

Figure 1. Structure of the BN specified by (3).

Suppose that eight independent observations from (3) are obtained, as shown in Table 1; then the sufficient statistics N_{ijk} can be obtained from simple counting in the table, for example, $N_{311} = \#\{V_3 = 1, V_4 = 1\} = 2$ and $N_{24(1,3)} = \#\{V_2 = 4, (V_3, V_4) = (1, 3)\} = 1$.

Because any discrete multivariate distribution can be represented by (1) and (2), the class of BN models is too general to be useful unless suitable assumptions on the network structure are made. A natural assumption is that the graph should be sparse. This is encoded by specifying a prior distribution for the graph that penalizes for the number of edges; that is, for some $0 < \gamma < 1$,

$$P(\mathcal{G}) \propto \gamma^{\sum_{i=1}^{n} |\Pi_i^{\mathcal{G}}|} = \prod_{i=1}^{n} \gamma^{|\Pi_i^{\mathcal{G}}|}. \tag{4}$$

We believe that some type of sparseness assumption is unavoidable for BN structure learning, even as we are mindful that such an assumption sometimes may lead to missing edges. The number of possible structures grows very quickly as a function of the number of edges in the structure; thus if the prior does not impose a penalty for increasing the number of edges, then the marginal prior density for the number of edges will overwhelmingly favor nonsparse graphs. It is known that likelihood and Bayesian methods do not perform well if the parameter space has very large cardinality, unless sieve or penalty functions are used to reduce the effective cardinality (Birge and Massart 1993; Wong and Shen 1995; Shen and Wasserman 2001). Because the space of DAGs is of very large cardinality, we must worry about this lack of sparsity. Furthermore, causal structures are of the most interest precisely when the structure is sparse. In situations such as the signaling network example, substantive knowledge supports the assumption that most of the interactions should involve a small number of genes and proteins; in fact, the network will offer little scientific insight if most proteins have a large number of parents. By using a sparseness prior, we give ourselves a much better chance of discovering the useful causal model if it exists.

Table 1. Eight observations from the graph defined by (3)

	V_1	V_2	V_3	V_4
1	2	1	1	1
2	1	1	1	1
3	2	4	1	3
4	1	1	3	1
5	1	2	2	2
6	1	3	3	3
7	1	1	3	1
8	3	2	3	3

The prior on the Θ parameters is usually chosen to be a product-Dirichlet distribution,

$$P(\Theta|\mathcal{G}) = \prod_{i=1}^{n} \prod_{k=1}^{q_i} \frac{\Gamma(\alpha_{i\cdot k})}{\Gamma(\alpha_{i1k}) \cdots \Gamma(\alpha_{ir_ik})} \theta_{i1k}^{\alpha_{i1k}-1} \cdots \theta_{ir_ik}^{\alpha_{ir_ik}-1}, \tag{5}$$

where $\alpha_{ijk} = \frac{\alpha}{r_i q_i}$. With this specification for the prior $P(\Theta, \mathcal{G}) = P(\Theta|\mathcal{G})P(\mathcal{G})$, we obtain the posterior distribution

$$P(\Theta, \mathcal{G}|\mathbb{X}) \propto \prod_{i=1}^{n} \gamma^{|\Pi_i^{\mathcal{G}}|} \prod_{k=1}^{q_i} \frac{\Gamma(\alpha_{i\cdot k})}{\Gamma(\alpha_{i1k}) \cdots \Gamma(\alpha_{ir_ik})} \times \theta_{i1k}^{N_{i1k}+\alpha_{i1k}-1} \cdots \theta_{ir_ik}^{N_{ir_ik}+\alpha_{ir_ik}-1}. \tag{6}$$

Integrating out Θ, we obtain, in closed form (Cooper and Herskovits 1992),

$$P(\mathcal{G}|\mathbb{X}) \propto \prod_{i=1}^{n} \left[\gamma^{|\Pi_i^{\mathcal{G}}|} \prod_{k=1}^{q_i} \frac{\Gamma(\alpha_{i\cdot k})}{\Gamma(\alpha_{i\cdot k} + N_{i\cdot k})} \prod_{j=1}^{r_i} \frac{\Gamma(N_{ijk} + \alpha_{ijk})}{\Gamma(\alpha_{ijk})} \right] = \prod_{i=1}^{n} P(V_i; \Pi_i^{\mathcal{G}}). \tag{7}$$

Note that both (6) and (7) also depend on hyperparameters γ and α. The marginal posterior distribution in (7) contains all of the information for the network structure $\mathcal{G}$ provided by the data. The main purpose of this article is to introduce computational methods for the inference of $\mathcal{G}$ starting from $P(\mathcal{G}|\mathbb{X})$. First, however, we must discuss some conceptual issues in the use of BNs in causal inference.

2.2 Causal Networks and Experimental Data

The use and interpretation of graphical models and BNs in causal inference has been studied in computer science (reviewed in Spirtes, Glymour, and Scheines 1993 and Pearl 2000b) and has rich connections to such well-established areas of statistics as path analysis and structural equations (Wright 1923; Holland 1988), potential outcomes, and randomization (Neyman 1990; Rubin 1978; Robins 1986). Our brief discussion here follows Pearl's formulation, which emphasizes modeling of the effects of intervention; refer to the foregoing references for deeper discussions on causal inference.

Given two variables Y and Z, we say that Y precedes Z causally if experimental interventions that change the value of Y can affect the distribution of Z but not vice versa. Given a collection of variables, the causal parents of Z are the set of variables whose values, if fixed by experimental intervention, can shield the intervention effect of any other variable that precedes Z causally; that is, once the values of the causal parents are fixed by intervention, the distribution of Z will not be affected by intervention on any other variables in the collection. We encode the causal relations among a collection of variables by a graph in which the vertexes represent the variables and the directed edges represent causal parent–child relations. The

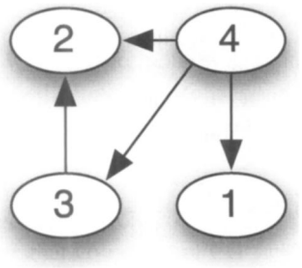

Figure 2. Network implied by an alternative decomposition.

graph must be a DAG because if a cycle exists, then two variables in the cycle will causally precede each other, which is not allowed for a causal relation.

The formulation of a Bayesian network as a graphical model for a joint distribution, as in Section 2.1, is not sufficient for causal inference. To illustrate this with a simple example, suppose that we want to interpret the directed edges in Figure 1 as causal relations, in which case we will immediately run into the difficulty that $p(v_1, v_2, v_3, v_4)$ also can be written as

$$\begin{aligned} &P(v_1, v_2, v_3, v_4) \\ &\quad = P(V_2 = v_2 | V_3 = v_3, V_4 = v_4) P(V_3 = v_3 | V_4 = v_4) \\ &\qquad \times P(V_1 = v_1 | V_4 = v_4) P(V_4 = v_4), \end{aligned} \tag{8}$$

which leads to the graph in Figure 2. Although both graphs (Figs. 1 and 2) represent exactly the same joint distribution, the direction of the edge between V_1 and V_4 is reversed. Clearly, only one of these can have a causal interpretation.

Thus the use of the BN model in causal inference depends on the assumption that the decomposition (1) that specifies the BN corresponds to an underlying causal structure; that is, the parents of V_i in the BN are the same as the causal parents of V_i. The conditional probabilities in this special decomposition admit an interpretation that is independent of the joint distribution. In fact, we consider these probabilities, referred to as causal probabilities, as the building blocks that allow specification of many related joint probability distributions under experimental intervention. For example, the causal probability $P(V_4 = a | V_1 = b)$ in the specification of $p(\cdot)$ in Example 1 is assumed to be the same as $P(V_4 = a | V_1$ set to $b)$; that is, the probability of observing $V_4 = a$ when V_1 is experimentally fixed to be b, with the implicit assumption that once the value of V_1 is fixed experimentally, the distribution of V_4 is also fixed regardless of the intervention on any other variables in the network. This is the unique characteristic of causal probabilities. A noncausal probability such as $P(V_1 = b | V_4 = a)$ can only be interpreted as a conditional probability derived from the joint distribution, that is, as

$$\frac{\sum_{v_2} \sum_{v_3} p(b, v_2, v_3, a)}{\sum_{v_1} \sum_{v_2} \sum_{v_3} p(v_1, v_2, v_3, a)},$$

where $p(v_1, v_2, v_3, v_4)$ is as given in Example 1. Note that this is not the same as the probability $P(V_1 = b | V_4$ set to $a)$, which in this case is just the casual probability $P(V_1 = b)$. (For a comprehensive treatment of intervention and causal modeling, see Pearl 2000a.)

In general, the inference of the causal relations and causal probabilities is a controversial topic. Although some authors (Spirtes et al. 1993; Pearl 2000a) have argued that causal inference from observational data is possible under a condition called "faithfulness," which requires that the true distribution processes only those independencies shared by all distributions whose densities can be factorized according to the DAG that represents the causal relations, their methodology has been challenged by others (Robins and Wasserman 1999). However, most authors agree that experimental interventions will partially mitigate the difficulties in causal inference. In this article we are concerned with inferring causal relations from data obtained through experimental interventions (i.e., experimental data). Typically, such interventions allow us to fix the value of some of the variables and then observe what happens to the other variables. Then the question is how to compute the posterior distribution for the structure $\mathcal{G}$ given such experimental data. Once the structure is known, inferring the causal probabilities is easy.

The answer to this key question has been given in many contexts in the study of intervention and causality (Robins 1986, 1987; Goldszmidt and Pearl 1992; Pearl 1993a,b; Spirtes et al. 1993). Cooper and Yoo (1999) gave the following formulation for the purpose of scoring a BN given data: consider case 3 in Table 1 of Example 1. If this data point is observational, then, according to (3), it contributes a factor (to the likelihood) equal to

$$\theta_{24(1,3)} \theta_{313} \theta_{432} \theta_{12\Phi}, \tag{9}$$

where Φ denotes the null state because V_1 has no parents and $(1, 3)$ denotes the joint state of the parent set of V_2. Thus the observation of this data point results in the increment (by 1) of the counts

$$N_{24(1,3)}, N_{313}, N_{432}, \text{ and } N_{12\Phi}$$

in the sufficient statistics $\{N_{ijk}\}$.

Next, consider the data point $(V_1, V_2, V_3, V_4) = (2, 4, 1^*, 3)$, where the notation 1^* means that V_3 is set at 1 by experimental intervention. Then, according to the causal interpretation of the probabilities in the decomposition in Example 1, the likelihood factor contributed by this data point is now given by (9) with θ_{313} omitted because there is no need to include the probability of observing $V_3 = 1$. Thus the only effect of the experimental intervention is that a corresponding data point will lead to increments of the sufficient statistics $\{N_{ijk}\}$ in the usual way, except when $V_i = j$ is set by the experimental intervention. Once the sufficient statistics are computed in this way, we can continue to use (7) for the marginal posterior of $\mathcal{G}$ given the experimental data.

3. SAMPLING BAYESIAN NETWORKS

3.1 Order-Space Sampling

Because directed acyclic graphs are equivalent to partial orderings of the vertices, there exists at least one total ordering of the vertices, denoted by $\sqsubset$, such that $V_i \prec V_j$ if $V_i \in \Pi_j$. Each of these total orderings is known as a "topological sort" or "linear extension" of the graph $\mathcal{G}$. Cooper and Herskovits (1992) exploited this fact to reduce the amount of computation

needed to find an optimal graph by assuming the knowledge of the ordering of the vertices. Suppose that the variables are indexed according to a known ordering; then the parents of V_j can only come from $V_1, \ldots, V_{j-1}$—that is, parents are chosen only from elements that precede V_j in the ordering, eliminating the need for cycle checking. Thus optimization of the parent sets $\Pi_i, i = 1, 2, \ldots, n$, can be performed independently rather than jointly. If the maximum in-degree (i.e., the maximum size of a parent set) is assumed to be bounded, then the search for the optimal structure compatible with a given ordering can be performed in time complexity polynomial in n.

Let us now consider computation of the posterior probability of a given order $\sqsubset$, defined as a sum over all graphs consistent with this order,

$$P(\sqsubset) \propto \sum_{\mathcal{G} \in \mathcal{G}_\sqsubset} P(\mathcal{G}) = \sum_{\mathcal{G} \in \mathcal{G}_\sqsubset} \prod_{i=1}^{n} P(V_i; \Pi_i^{\mathcal{G}}), \tag{10}$$

where $P(V_i; \Pi_i^{\mathcal{G}})$ is defined as in (7) and $\mathcal{G}_\sqsubset$ is defined as

$$\mathcal{G}_\sqsubset = \{\mathcal{G} : \text{all elements of } \Pi_i^{\mathcal{G}} \text{ precede } V_i \text{ in } \sqsubset\}. \tag{11}$$

Brute force summation is infeasible because even if the maximum in-degree is bounded, the number of DAGs consistent with an order is still $2^{O(n \log(n))}$. Friedman and Koller (2003) made the key observation that this summation also can be computed efficiently based on a similar argument as in the optimization case, which they attributed to Buntine (1991); specifically,

$$P(\sqsubset) \propto \prod_{i=1}^{n} \sum_{\Pi_i \in \Pi_i^\sqsubset} P(V_i; \Pi_i), \tag{12}$$

where $\Pi_i^\sqsubset$ is the collection of admissible parents sets for i in the order $\sqsubset$. This reduces the computational complexity from $O(\binom{n}{k}^{n/2})$ to $O(n\binom{n}{k})$, where $1 \le k \le n-1$ is the maximum in-degree (Dash and Cooper 2004).

With an efficient computation of $P(\sqsubset)$ in hand, Friedman and Koller (2003) then introduced a two-stage algorithm for sampling DAGs:

- Use the Metropolis–Hastings algorithm to sample the order $\sqsubset$ according to its probability (12). Note that this step is possible only because $P(\sqsubset)$ can be evaluated efficiently. Specifically, the chain is first initialized with a random permutation of the ordering. Moves are proposed by the swapping of one or more elements in the ordering. Other moves that maintain the detailed balance are possible; "deck cutting" moves and specific crossover moves from the Traveling Salesman problem domain are also possible, although we have found that they do not do much to improve performance.
- Given an order $\sqsubset$ generated as before, sample a network structure $\mathcal{G}$ from the space of DAGS compatible with this order according to the conditional probability distribution. This is easy, because the parent set of each V_i can be sampled independently from $\sqsubset$ by considering only the proper term of the product in (12) to obtain each Π_i.

3.2 Order-Graph Sampling

Whereas the Friedman–Koller (FK) algorithm is an important advance by its virtue of fast mixing, the BN structures generated by the FK algorithm does not follow the correct posterior distribution. The difficulty stems from the fact that the order $\sqsubset$ is not a function of the structure $\mathcal{G}$. A graph may be compatible with more than one order, so that the orders do not induce a partition in DAG space. The problem cannot be solved by defining the graph-to-order function by arbitrarily choosing one of the many orders (say, the one closest to a prespecified order) compatible with a given graph $\mathcal{G}$ as its image. If we do so, then although this order variable will have a well-defined posterior distribution, it will be different than that given by (10). In other words, the probability distribution in order space targeted by the FK sampling is not a true marginal posterior distribution of a function of the variable $\mathcal{G}$. This fact induces an intrinsic bias in the samples generated from the FK algorithm, so that they can no longer be expected to follow the correct posterior distribution.

Example 2. To illustrate this bias, consider the trivial example of a two-vertex problem, which has three possible graphs, $V_1 \to V_2$, $V_1 \perp V_2$, and $V_1 \leftarrow V_2$, with assigned probabilities $P(V_1 \to V_2) = \frac{1}{6}$, $P(V_1 \perp V_2) = \frac{2}{6}$, and $P(V_1 \leftarrow V_2) = \frac{3}{6}$. The orderings V_1V_2 and V_2V_1 have probabilities $\frac{6}{8}(\frac{1}{6} + \frac{2}{6}) = \frac{3}{8}$ and $\frac{5}{8}$. Calculating the probabilities of the graph through the orderings gives

$$\begin{aligned} P(V_1 \to V_2) &= P(V_1V_2)P(V_1 \to V_2 | V_1V_2) \\ &= \frac{3}{8}\frac{1}{3} = \frac{1}{8}, \\ P(V_1 \perp V_2) &= P(V_1V_2)P(V_1 \perp V_2 | V_1V_2) \\ &\quad + P(V_2V_1)P(V_1 \perp V_2 | V_2V_1) \\ &= \frac{3}{8}\frac{2}{3} + \frac{5}{8}\frac{2}{5} = \frac{1}{2}, \end{aligned}$$

and

$$\begin{aligned} P(V_2 \leftarrow V_1) &= P(V_2V_1)P(V_1 \leftarrow V_2 | V_2V_1) \\ &= \frac{5}{8}\frac{3}{5} = \frac{3}{8}. \end{aligned}$$

As we can see, the true probabilities are such that $P(\mathcal{G}_1) < P(\mathcal{G}_2) < P(\mathcal{G}_3)$, whereas the probabilities calculated using the orderings instead satisfy $P(\mathcal{G}_1) < P(\mathcal{G}_2) > P(\mathcal{G}_3)$.

In general, the expectation under the FK sampler is given by the following result.

Lemma 1. Let $f(\cdot)$ be a function on the space of DAGs, and let $\mathcal{G}'$ be a random graph generated by the FK algorithm. Then

$$E[f(\mathcal{G}')] = \frac{\sum_{\mathcal{G}} |\sqsubset_\mathcal{G}| f(\mathcal{G}) P(\mathcal{G})}{\sum_{\mathcal{G}} |\sqsubset_\mathcal{G}| P(\mathcal{G})}, \tag{13}$$

where $\sqsubset_\mathcal{G}$ is the set of orders compatible with $\mathcal{G}$.

Proof.

$$E[f(\mathcal{G}')] = \sum_{\sqsubset} \frac{P(\sqsubset)}{\sum_{\sqsubset} P(\sqsubset)} \sum_{\mathcal{G} \in \mathcal{G}^\sqsubset} f(\mathcal{G}) \frac{P(\mathcal{G})}{\sum_{\mathcal{G} \in \mathcal{G}^\sqsubset} P(\mathcal{G})}$$

$$= \frac{\sum_{\sqsubset} \sum_{\mathcal{G} \in \mathcal{G}^{\sqsubset}} f(\mathcal{G}) P(\mathcal{G})}{\sum_{\sqsubset} P(\sqsubset)}$$

$$= \frac{\sum_{\mathcal{G}} |\sqsubset_{\mathcal{G}}| f(\mathcal{G}) P(\mathcal{G})}{\sum_{\mathcal{G}} |\sqsubset_{\mathcal{G}}| P(\mathcal{G})}.$$

Lemma 1 shows that $\mathcal{G}'$ generated by the FK sampler is distributed as

$$P(\mathcal{G}') \propto |\sqsubset_{\mathcal{G}}| P(\mathcal{G}), \tag{14}$$

which generally is biased due to the extra factor $|\sqsubset_{\mathcal{G}}|$. In small problems like our four vertex example, we know how many orders are consistent with each graph, allowing us to correct the bias introduced by the overlap. But in large problems, calculation of the overlap is infeasible, because the linear extension counting problem is #P hard (Brightwell and Winkler 1991).

To overcome this problem, we propose the following solution. First, we sample from each unique sampled order, $\sqsubset$, with replacement until we have sampled k unique graphs such that

$$\sum_{i=1}^{k} P(\mathcal{G}_i | \sqsubset) \geq (1 - \varepsilon). \tag{15}$$

Let $\mathcal{U}$ be the union of the graph samples from all orders. We treat $\mathcal{U}$ as an importance-weighted sample, in which the weight from a graph $\mathcal{G}_i$ in $\mathcal{U}$ is $P(\mathcal{G}_i)/\sum_{\mathcal{G} \in \mathcal{U}} P(\mathcal{G})$ and $P(\mathcal{G})$ is given by (7). By setting ε to be small, we can ensure that a graph with a high posterior score will have a high likelihood of being included in $\mathcal{U}$ as long as it is compatible with one of the sampled orders. Once it is included in $\mathcal{U}$, the graph will be weighted correctly in the final sample.

To assess the waiting time until (15) is satisfied, let g_i, $i = 1, \ldots, I$, be the distinct graphs in $\sqsubset$ ordered such that $p_i = P(g_i | \sqsubset)$ satisfy $p_1 \geq p_2 \geq \cdots \geq p_I$. Draw graphs from $\sqsubset$, with replacement, according to $\underset{\sim}{p}$. Let

$$I_i(n) = \begin{cases} 1 & \text{if } g_i \text{ has not been seen in } n \text{ trials} \\ 0 & \text{otherwise,} \end{cases}$$

which has expectation $E[I_i(n)] = (1 - p_i)^n$. Let U_n be the sum of probabilities for graphs not seen in n trials,

$$U_n = \sum_{i=1}^{I} I_i(n) p_i,$$

with expectation $E[U_n] = \sum_{i=1}^{I} p_i (1 - p_i)^n$. Now assume that $p_{i+1} \leq \alpha p_i$ for some $0 < \alpha < 1$ and choose k sufficiently large to satisfy (15), namely

$$k \geq \frac{\log(\varepsilon(1 - \alpha)/p_1)}{\log(\alpha)}.$$

Then,

$$E[U_n] \leq \sum_{i=1}^{k} p_i (1 - p_i)^n + \varepsilon \leq k(1 - p_k)^n + \varepsilon,$$

with n chosen so that $k(1 - p_k)^n \leq \varepsilon$. It follows that if

$$n \geq \frac{\log(\varepsilon/k)}{\log(1 - p_k)},$$

then we have

$$E[U_n] \leq 2\varepsilon. \tag{16}$$

This analysis shows that U_n can be bounded by γ if n is of order $\log(\frac{\log(1/\gamma)}{\gamma})$.

Lemma 2. If $p_{i+1} \leq \alpha p_i, \ \forall\, i$ and

$$n \geq \frac{\log(\gamma^2/k)}{\log(1 - p_k)},$$

where

$$k = \frac{\log(\gamma^2(1 - \alpha)/p_1)}{\log(\alpha)},$$

then $P(U_n > \gamma) \leq 2\gamma$.

Proof. Applying the Markov inequality, we have

$$P(U_n > \gamma) \leq \frac{E[U_n]}{\gamma},$$

where $E[U_n] \leq 2\varepsilon$ from (16). The result follows by setting $\varepsilon = \gamma^2$.

For example, in a series of 10-vertex simulations, we found an average k of 232 unique graphs required to achieve $(1 - \varepsilon) = .95$, requiring an average of 737 samples per ordering and an average estimated α value of .81. Because sampling graphs given an order is a fast computation, this step consumes negligible time compared with the time needed to sample the necessary orders. Finally, we note that before sampling, we can compute the graph with the highest probability within an order. This often can reduce the number of samples required for (15). To see the difference between the order MCMC sampler (FK) and our bias-corrected order graph (OG) sampler, we compared the posterior probability estimate obtained by the algorithms with the exact posterior probability for all graphs in the four-vertex problem. The results, illustrated in Figure 3, show that FK has considerable biases and that OG is effective in correcting these biases.

4. IMPLEMENTING THE ORDER GRAPH SAMPLER

4.1 Single-Queue Equi-Energy Sampling

Whereas the order MCMC approach greatly improves mixing compared with MCMC in the space of DAGs, it remains still susceptible to local trapping. We first observed this problem in real-world data sets such as the samples from the 11-vertex problem given in Figure 4, which shows the energies of 10 different chains run for 100,000 iterations. We also can see this problem in simulated data, such as the 32 chains of a simulated 10-vertex data set shown in Figure 5.

Thus, even when sampling is performed over order space, there is a need to use advanced MCMC methods that are less susceptible to local trapping. Here we chose to use equi-energy sampling (Kou, Zhou, and Wong 2006) for our software implementation. In brief, like other population-based MCMC approaches, the equi-energy sampler relies on a temperature ladder $1 = T_1 < \cdots < T_I$ that is used to define I Boltzmann distributions. In addition, the equi-energy sampler uses a ladder

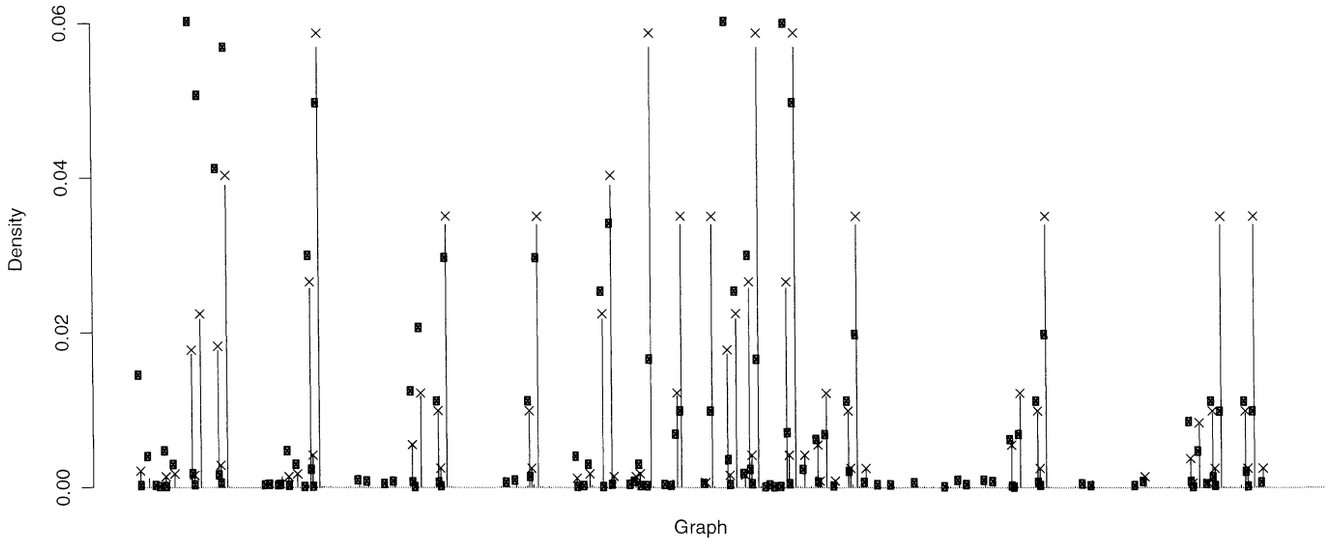

Figure 3. FK posterior probability estimates (■) and OG posterior probability estimates (×), compared with the true posterior probabilities (—) for the four-vertex problem. $\varepsilon = .95$.

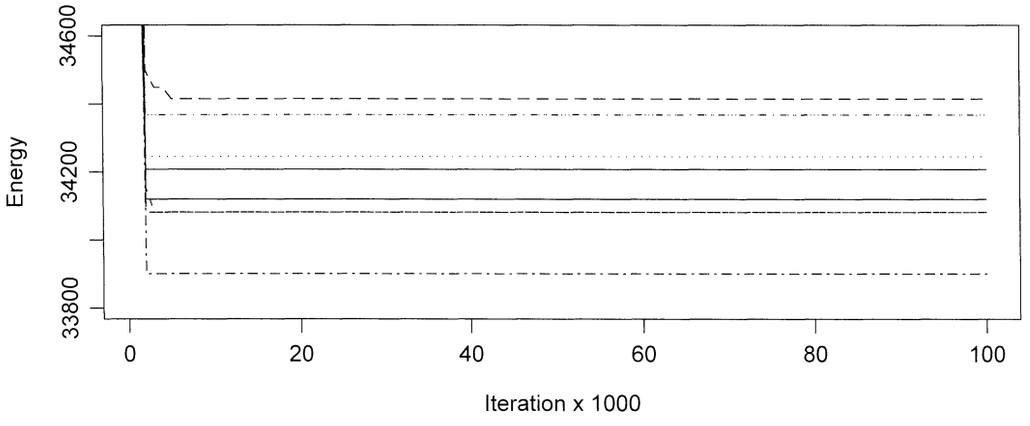

Figure 4. Energies for samples from 10 runs of the order MCMC sampler in an 11-vertex problem. As we can see, there are several local minima for the sampler often quite far from the global minimum energy, which is discovered by one run of the sampler. Although not visible due to scale, each chain is still moving in a small region around the trapping point with an acceptance rate of around 30%.

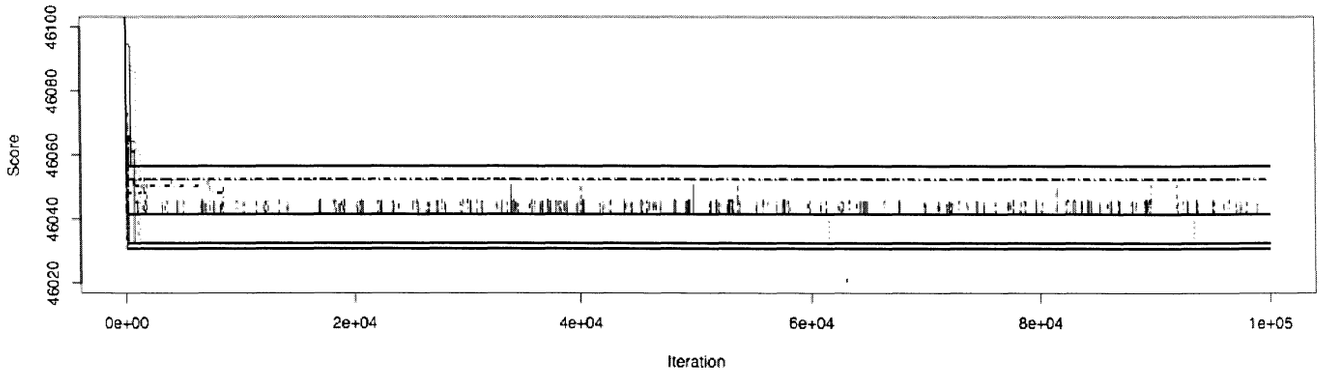

Figure 5. Energies for another set of order MCMC chains run on a simulated 10-vertex data set. In this case, 32 chains are run for 100,000 iterations. A trapping phenomenon similar to that shown in Figure 4 is evident. In this data set, the trapping locations apparently are closer in order space, allowing for some crossing between local minima; however, once trapped in a local minima, most of the chains tend to stay there.

of truncation energies $H_{min} = H_1 < \cdots < H_I < \infty$ such that tempered distributions are defined as

$$\pi_i(x) = \exp\left(\frac{-\max[H(x), H_i]}{T_i}\right). \tag{17}$$

Each chain is then started in a staggered fashion such that the ith chain starts $B + N$ iterations after the $i + 1$ chain, where $B + N$ are the numbers of burn-in and collected iterations of the chain of interest $\pi_1(x)$. The chains themselves run using the normal Metropolis–Hastings moves, although better results are usually obtained from scaling the step size of the higher-temperature chains, because they are more likely to accept a larger move. This allows the high-temperature chains to move more quickly through the space to find samples for later communication to lower-temperature chains. In the case of the OG sampler, the step size adjustment takes the form of a generalized swap move. In the ith chain, each position has a $p_i \propto \beta^{1/T_i}$ probability of being selected for a swap move. With positions $s_1, \ldots, s_k$ selected for swap, we then perform a "cylindrical shift" operation such that $\sqsubset_{s_i} \leftarrow \sqsubset_{s_{i+1}}$, $\sqsubset_{s_{i+1}} \leftarrow \sqsubset_{s_{i+2}}$ and $\sqsubset_{s_k} \leftarrow \sqsubset_{s_1}$. Typically, we aim to select on average $\frac{n}{4}$ and 2 positions, respectively for the highest-temperature and lowest-temperature chains. In our experience, this simple rule gives satisfactory results for most problems.

Kou et al. (2006) based their equi-energy move for the kth chain on an energy binning constructed from the $(k+1)$th chain (i.e., the next higher-temperature chain). We have found that better performance can be achieved by using a single energy binning constructed from all of the chains with temperature above k. Each energy bin now contains a mixture of samples from all chains. By tracking the number of samples, M_i, from each chain, we can calculate the modified transition kernel as

$$Q\left(y; x_i^{(t)}\right) = \sum_{j=i+1}^{I} w_j \pi_j(y), \tag{18}$$

where $w_j = M_j / \sum_{k=i+1}^{I} M_k$. This allows more immediate communication of low-energy (high-probability) samples from high-temperature chains to the chain of interest with less sensitivity to the selection of energy/temperature ladders. In this article we use this "single-queue" variant of the equi-energy sampler exclusively.

4.2 Efficient Order Scoring

The key to any order sampler, be it order MCMC sampler (FK) or the population-based order-graph equi-energy MCMC sampler (OGEE), is the ability to quickly assess the posterior probability of an ordering. The standard method for scoring DAGs realizes performance through the use of a family score cache that stores the local score for a particular V_i, Π_i configuration and its probability. For optimization and sampling in the space of DAGs, typically the number of changed configurations per move is limited to a single parent set; thus this scheme is sufficient for fast calculation. Order scoring, on the other hand, involves a large number of family checks, making the cache-based approach a poor choice for implementation.

To achieve fast order scoring, we have developed a stream-based approach similar to the Google MapReduce architecture (Dean and Ghemawat 2004). We precomputed an initial parent set database with all possible parent sets from size 0 to size k, where $k \ll n$. Each entry in the database consists of a key, specifying the parent set Π and a vector of n values containing $\log P(V_i; \Pi)$ or $-\infty$ if $i \in \Pi$. To calculate the probability of an order $\sqsubset$, we began by initializing an intermediate probability vector $(s_1, \ldots, s_n)$ for each $V_1, \ldots, V_n$. We then calculated the position vector $(p_1, \ldots, p_n)$ such that p_i is the index of V_i in $\sqsubset$. We considered each record in the database in an arbitrary order, possibly in parallel, first calculating the rightmost position of the elements in Π, $p_{max} = \max_{k \in \Pi} p_k$. We then updated the intermediate scoring vector as

$$s_i = \log\left(\exp(s_i) + P(V_i; \Pi)\right), \qquad i = \sqsubset_{p_{max}}, \ldots, \sqsubset_n, \tag{19}$$

for each entry and, finally, calculated $\log P(\sqsubset) = \sum_{i=1}^{n} s_i$. An example of a single step in this process is given in Figure 6.

This approach achieved its performance by using a large set of precomputed scores that are likely to be needed and eliminating the large number of random access lookups required by traditional algorithms through the use of sequential lookups with a minimal number of lookup failures (i.e., parent set records that have no meaning for the current ordering). In addition, the structure of the database does not depend on any intrinsic ordering of the parent sets, such that records may be removed or added in essentially $O(1)$ time, allowing for the adaptive construction of the database to accommodate, for example, parent set sizes larger than k in place of smaller, but low-scoring, parent sets over time.

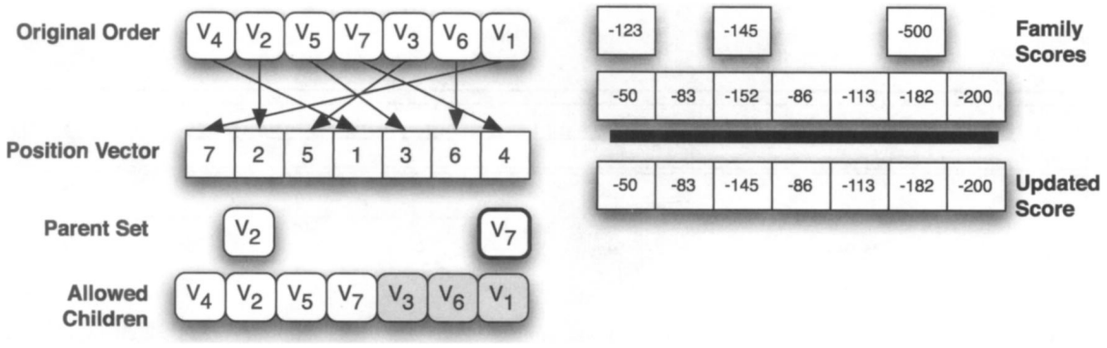

Figure 6. A single step in the order scoring process.

5. A SIMULATION STUDY TO ASSESS PERFORMANCE

To validate our algorithm's performance and correctness, we conducted a number of simulation experiments pitting our OGEE sampler against various DAG-based samplers in terms of their ability to recover true edges in randomly generated DAGs.

5.1 Simulation Setting

We generated the random DAGs using the Bayesian Network Toolkit (Murphy 2001), the algorithms of which follow the approach of Ide and Cozman (2002). In brief, the algorithm first chooses a random ordering and then randomly selects between 0 and $\min(i-1, K)$ parents for each vertex i from the vertices that precede it in the ordering. Each vertex has three states with local probability distributions taken from the product-Dirichlet. In each simulation, E $(0 \leq E \leq n)$ denotes the number of nodes to be experimentally fixed (intervention nodes). The amount of experimental data for each intervention node is set to N/n, where N is the number of observations and n is the number of vertices such that in a simulation with $N = 1{,}000$ observations and $n = 10$ nodes, $E = 2$ specifies that 200 of the total 1,000 observations will be experimental data.

5.2 Criterion for Comparison of Algorithms

In all of our tests, we chose to construct a threshold classifier as our method of algorithmic comparison, similar to the comparison used in the original order MCMC analysis (Friedman and Koller 2003). In this case, we used our graph sample to rank pairwise feature probabilities and select a cutoff value of α edges. For each possible α, we could determine the number of true positive edges as well as the number of false-positive edges, allowing for the construction of ROC and calculation of the AUC for each reconstruction. From the classifier literature, an AUC $> .9$ is considered an excellent classifier, an AUC of .8 is considered good classifier, and so on (Fawcett 2003). By chance, we would expect an AUC of .5 on average.

5.3 Results From Observational Data With Correctly Specified In-Degree

For our first simulation study, we restricted our attention to purely observational data and graphs whose maximum in-degree matches the maximum in-degree specified by our algorithms—in this case a maximum of four parents. To obtain a sense of how the size of the graphs and the number of observations affect different algorithms, we generated graphs with 8, 10, 12, and 14 vertices with observations ranging from 5,000 to 100. We replicated each experiment five times with different randomly sampled graphs and data. In all cases, we used a prior $P(\mathcal{G}) = \gamma^{-|\mathbb{E}|}$ with $\gamma = 10$.

For comparison, we used a direct DAG MCMC sampler that uses equi-energy sampling to improve mixing performance (DAG–EE), an importance reweighting sampler that uses simulated annealing to obtain numerous optimal graphs that are then renormalized to form a posterior sample (DAG–IS), and, finally, our own OGEE sampler. Because we are interested in sampling performance, for the moment we ignored optimizers such as the classic K2 algorithm.

Table 2. The mean AUC for the ROC curves for the DAG–EE, DAG–IS, and DAG–EE

		8	10	12	14
100	OGEE	.73	.69	.73	.71
	DAG–IS	.69	.64	.62	.67
	DAG–EE	.48	.51	.47	.50
500		.82	.81	.90	.87
		.70	.72	.71	.72
		.46	.55	.50	.52
1,000		.87	.84	.92	.89
		.77	.72	.75	.73
		.49	.54	.52	.51
2,500		.88	.86	.93	.89
		.77	.77	.74	.72
		.48	.58	.50	.50
5,000		.91	.91	.97	.94
		.73	.76	.76	.74
		.49	.58	.50	.53

In all three cases, we chose a number of iterations such that the amount of "wall time" (i.e., the physical time as opposed to CPU time or some other measure) for each algorithm was roughly matched running on identical hardware and identical operating systems (a 16-processor Apple XServe cluster in this case). We ran the DAG EE sampler (the fastest algorithm for a single iteration) for 100,000 burn-in iterations and collected 100,000 iterations for processing. We ran the greedy search algorithm with 100,000 unique starting locations to run in roughly the same amount of time. Finally, we ran the OGEE sampler for 15,000 burn-in order samples and 15,000 collected order samples. The graph sampling phase of OGEE given an order requires a negligible amount of additional time with at most 1,000 graphs sampled from each order and merged into a unique set.

Table 2 gives the average AUC results for the OGEE, DAG–IS, and DAG–EE samplers. As we can see, the OGEE sampler performed with 500 observations had better performance than the importance sampler with 5,000 observations. In this case it seems that the importance sampler's performance is directly tied to the ability of the optimization routine to encounter a good starting location, because a single graph often dominates all other graphs in the sample, effectively wasting the information from most samples. We also found that the DAG–EE sampler did not achieve adequate mixing even with the introduction of the equi-energy sampler. In fact, although it can perform well in "gold standard" networks such as ALARM, DAG MCMC never appears to mix well when learning randomly generated graphs.

5.4 Observational Data With Incorrectly Specified In-Degree

A limitation our algorithm is the need to place an upper bound on the maximum size of a parent set of each vertex. Although for many problems, it may be reasonable to assume a sparse graph such that most vertices will not have a parent set larger than the maximum in-degree, it is entirely possible that we will encounter the situation in which the parent

sets of a few vertices will exceed the number of parents allowed by our algorithm. To assess the algorithms' performance when some parent sets are larger than the assumed maximum in-degree, we generated random graphs with 10 vertices and a maximum in-degree of 9 (i.e., unlimited) for each node. We ran each algorithm with a maximum in-degree of 3 for all tests. Given that DAG–EE performed very poorly even with the correct maximum in-degree, here we compared only DAG–IS and OGEE.

Using 50 simulations with random data and graphs for each simulation, we obtained examples with between 0 and 12 edges beyond the possible number of edges of any single graph sampled under the maximum in-degree bound. The mean AUC across both the order graph and importance samplers remained relatively unchanged at .89 and .71, with a general downward trend as the number of edges was increased, with .68 being the worst case for the order graph sampler (although this appears to be an outlier, because the mean AUC for graphs with 12 extra edges was .85).

For comparison, we consider the oracle K2 algorithm. Under the usual K2 algorithm (Cooper and Herskovits 1992), each vertex adds a parent that improves the node score until no further improvement can be made or until the maximum in-degree k is reached. In the oracle K2 case, we provided each node with the true set of parents, which were chosen at random until either all parents were selected or the maximum in-degree was reached [i.e., $|\Pi_i| = \min(k, k_i^{true})$ for each node]. We then constructed an ROC classifier from this graph by rank-ordering the selected parents first and then randomly distributing the remaining parents, a best-case optimization scenario. We then calculated the AUC for the oracle K2 and compared it with the OGEE and DAG–IS results, as shown in Figure 7. The OGEE sampler outperformed oracle K2 in 48% of the cases, whereas DAG–IS always underperformed both oracle K2 and OGEE. Although not shown, the OGEE sampler also outperformed order-graph parallel tempering (OGPT), which outperformed oracle K2 in only 36% of cases.

5.5 Effects of Experimental Data

This simulation compared the changes in performance of the OG and importance samplers as the amount of experimental data increases, particularly when little data are available. In this case we used 100 observations for graphs with 10 vertices and an in-degree of 3, both true and assumed (i.e., $k = K$). Using 0%, 20%, 50%, 80%, and 100% experimental data, we performed 15 independent simulations for each algorithm. The results, summarized in Figure 8, show a general improvement for both algorithms as the amount of experimental data is increased, but that OGEE is clearly superior to DAG–IS for any given amount of experimental data.

6. AN APPLICATION TO FLOW CYTOMETRY DATA

6.1 Introduction

In this example, we investigated the performance of our sampling techniques in analyzing experimental data from a polychromatic flow cytometry experiment originally presented by Sachs, Perez, Peér, Lauffenburger, and Nolan (2005). Polychromatic flow cytometry allows for the simultaneous probing of the phosphorylation state of a number of different proteins within a cell. This is accomplished by first "fixing" (i.e., killing) the cell and then can staining each protein with a fluorescent marker protein, which then can be scanned by a laser tuned to a particular wavelength as each cell flows past in a column of fluid. Importantly, because this does not require cell lysis, as is the case in microarray experiments, data are collected on a cell-by-cell basis, leading to large amounts of data in each experiment.

This experiment involved an investigation into the interaction between the major mitogen-activated protein kinase (MAPK) pathways in human CD4+ T cells. These pathways generally serve to amplify the transmission of signals originating at the cell surface (in this case from the cell surface receptors CD3 and CD28) to the nucleus to effect a change in the genetic program. Figure 9 provides an overview of the potential network structure derived from interactions reported in the literature without respect to a particular cell type. In a particular cellular subpopulation, it is possible that certain interactions are nonexistent or weak, whereas there also may be cross-talk interactions that have not yet been investigated. As such, it is impossible to consider this structure as anything more than a general qualitative guide for the selection of targets for monitoring.

This experiment consisted of 11 targets selected from those shown in Figure 9, each labeled with a distinct fluorescent marker. The experimenters perturbed the system with nine

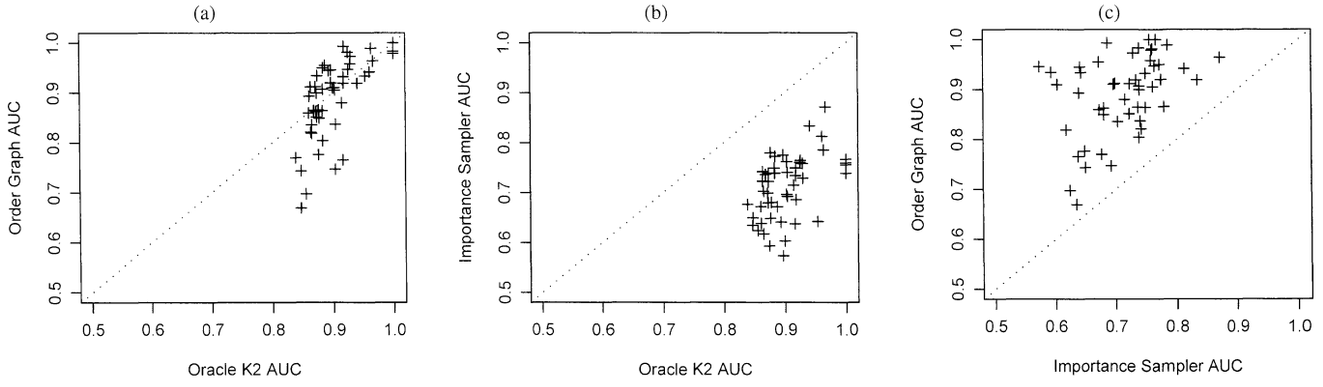

Figure 7. Performance of (a) the OGEE sampler versus an oracle K2 optimizer, (b) the DAG–IS versus the oracle K2, and (c) the order graph versus the importance sampler.

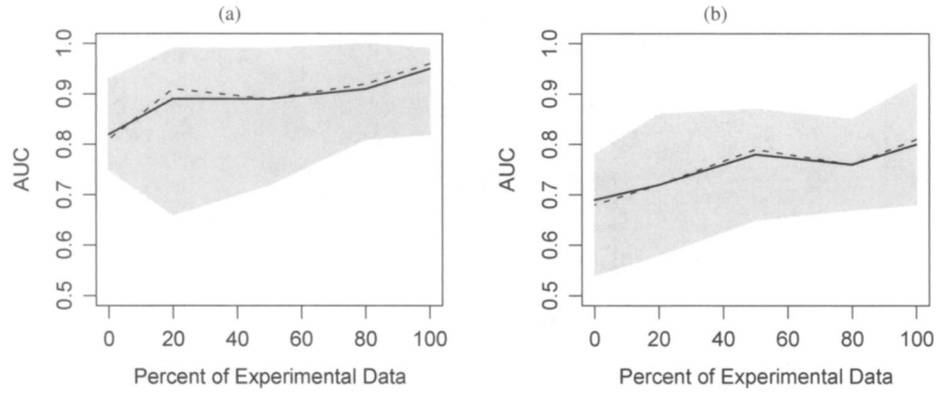

Figure 8. (a) Classifier performance in terms of AUC as the amount of experimental data is increased. The shaded region shows the range of results from 15 simulations, whereas the solid line indicates the mean AUC and the dashed line indicates the median AUC. (b) The same simulations using the importance sampler. In all cases, 100 observations were used with 0%, 20%, 50%, 80%, and 100% of the data obtained through a random experiment.

chemicals known to inhibit or activate a specific targeted protein along with a chemical known to activate CD3 and CD28 for the purpose of activating the entire pathway.

6.2 Original Analysis

The original analysis conducted by Sachs et al. (2005) used the importance sampler as comparison in the previous section on simulation. The source data, which are publicly available, were discretized into "low," "medium," and "high," using an information-preserving technique (Hartemink 2001). For those inhibitory perturbations that block the activity of a target protein without affecting its phosphorylation status (measured by the FACS instrument), the signal for the target was set at a "low" value.

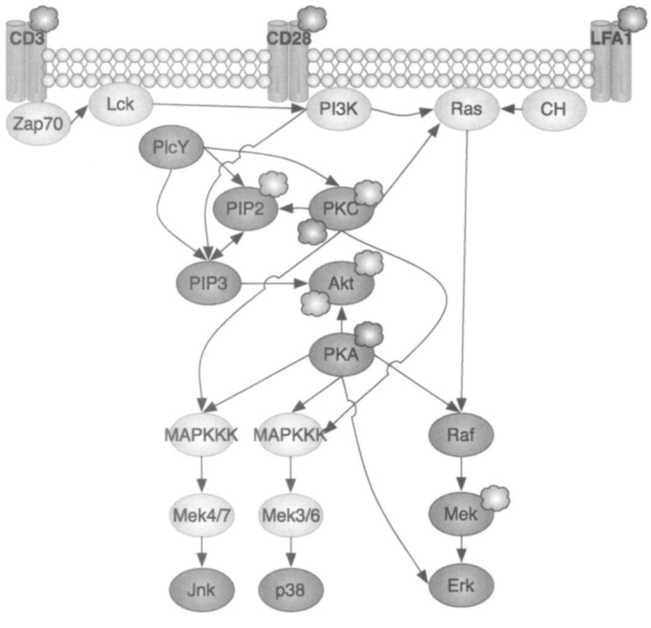

Figure 9. A nonspecific map of the MAPK cascades under study. The proteins being probed are indicated by the darker shading, whereas those involved in the pathway but unprobed are indicated by lighter shading. Note that this map is generic and not specific to the cell type of interest in the study.

After preprocessing, 500 data sets were generated. Each data set consisted of 600 cells sampled from each of the 9 experiments. Simulated annealing was used to obtain an optimal DAG from each of the data sets. Thus, after 500 simulated annealing runs, they now had a set of 500 DAGs each with an associated score. To estimate the marginal feature probabilities, bootstrap samples were generated from this data set according to their scores.

6.3 Analytic Setup

Much like in our experiment with the random graphs, we created a competitive experimental setting for our analysis, although we used a bootstrap-based technique similar to the original analysis rather than the parallel tempering or DAG equi-energy approaches. Our bootstrap implementation followed those approaches as closely as possible (Friedman, Nachman, and Peér 1999) although we used the same efficient scoring and database mechanisms as in our OG sampler, to ensure fair comparisons between running time and other performance metrics.

For the simulated annealing technique, we used parameters as close as possible to those of the original Sachs analysis and generated 500 separate graphs from data sets sampled with replacement from the complete data. We then ran each annealing chain for 100,000 iterations with a starting temperature of 50 and a finishing temperature of 1 with a log-cooling schedule.

6.4 Results

Because no ground truth is available in this test, we chose to use a 10-fold cross-validation approach to assess the relative performance of our two algorithms. In each round, we removed 10% of the data uniformly from each experimental condition. For example, with 600 observations per experiment, we removed 60 observations from each of the 9 experimental conditions. We then obtained the predicative probability for each algorithm within each subset through model averaging. The results, reported in Figure 10, demonstrate consistently better performance from the OG sampler, as was the case in the simulated results of the last section.

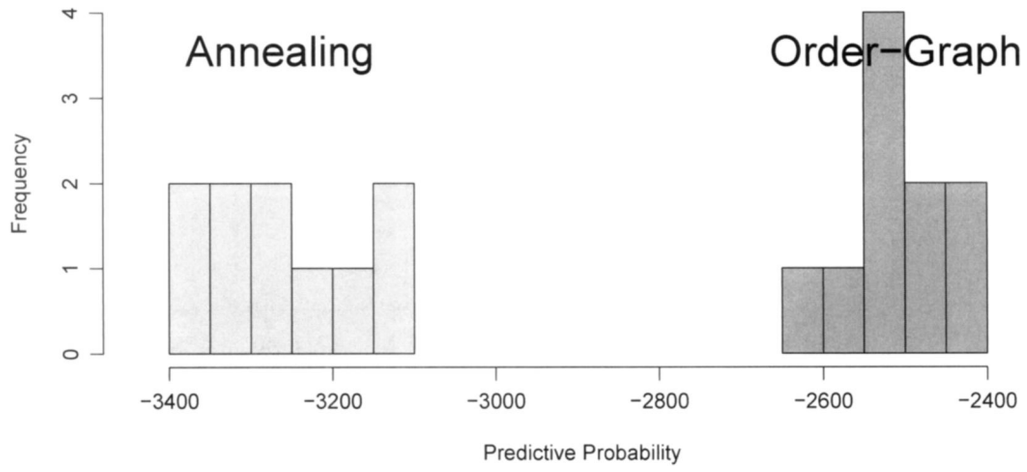

Figure 10. A histogram of the scores obtained by each algorithm during cross-validation. The lighter bars represent the simulated annealing results; and the darker bars, the order graph results.

Similarly, by pooling the results for the feature probabilities from each posterior sample, we could construct a "mean" graph from each algorithm. These graphs, given in Figure 11, show that the OG sampler obtained an average graph qualitatively more similar to the nonspecific graph given in Figure 9 than to that obtained from the importance sampling algorithm. For instance, the well-known "Raf → Mek → Erk" pathway is recovered by OGEE, but not by DAG–IS. In addition, the OGEE sampler required a running time between 3 and 5 minutes, whereas the DAG–IS sampler required between 3 and 5 hours on the same computing hardware.

7. CONCLUSIONS AND FUTURE WORK

With the development of the OG sampling approach, we now can address complete-data problems with reasonable confidence in problems of moderate size. We have accomplished this through a combination of methods, including extending the already powerful order MCMC approach with equi-energy sampling, bias-corrected graph sampling, and stream-based computation. Currently, our implementation precomputes the scores for all (V_i, Π_i) combinations subject to a bound on $|\Pi_i|$. Although, as seen in simulations, this restriction does not necessarily significantly inhibit our ability to construct a high-quality classification capable of outperforming an equivalent pure optimization approach, better performance perhaps can be found through algorithmic improvement. As part of future work, one promising approach may be to use an adaptive database that assigns a maximum in-degree of k_i to each vertex independently. During the tuning process, the individual k_i then could be calculated to allow some vertices to have larger parent sets, whereas others might have their parent sets reduced. This is similar in spirit to other algorithms such as the sparse candidate algorithm, also proposed by Friedman.

Another challenge is the missing-data case. For example, we may wish to study signal transduction systems with more components than the maximum number of colors allowable by current flow cytometers. To study these larger systems, we are now developing algorithms that allow us to perform experiments with data that are "missing by design," such that the targets of the dyes are varied under the same set of experimental conditions to obtain readings for different proteins under the same conditions. The hope is that the large number of single-cell samples will allow us to "mesh" our experiments to construct a pos-

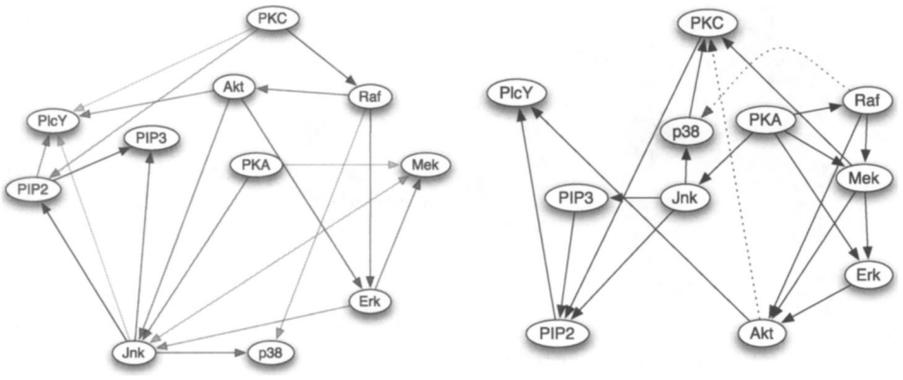

Figure 11. (A comparison) The average graph obtained by the importance sampler using the bootstrap algorithm. The lighter edges denote smaller probabilities. (Right) The average graph obtained using the OG sampler. All edges appeared in at least 90% of the graphs, with the exception of the dotted edges, which appeared in >50%, but <80% of cases.

terior graph sample for all of the elements. The complete-data algorithms discussed in this article will be useful as a component of the algorithms in the incomplete-data inference, a problem likely to be a major focus of computational development in the coming years.

[Received October 2006. Revised January 2008.]

REFERENCES

Beinlich, I., Suermondt, G., Chavez, R., and Cooper, G. F. (1989), "The ALARM Monitoring System: A Case Study With Two Probabilistics Inference Techniques for Belief Networks," in *The Second European Conference on Artificial Intelligence in Medicine*, New York: Springer-Verlag, pp. 247–256.

Birge, L., and Massart, P. (1993), "Convergence Rates of Minimal Contrast Estimators," *Probability Theory and Related Fields*, 97, 113–150.

Brightwell, G., and Winkler, P. (1991), "Counting Linear Extensions Is #P–Complete," in *TOC'91: Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing*, New York: ACM Press, pp. 175–181.

Buntine, W. L. (1991), "Theory Refinement on Bayesian Networks," in *Proceedings of the 7th Annual Conference on Uncertainty in Artificial Intelligence (UAI-91)*, San Francisco, CA: Morgan Kaufmann, pp. 56–60.

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

Cooper, G. F., and Yoo, C. (1999), "Causal Discovery From a Mixture of Experimental and Observational Data," in *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA: Morgan Kaufmann, pp. 116–125.

Dash, D., and Cooper, G. F. (2004), "Model Averaging for Prediction With Discrete Bayesian Networks," *Journal of Machine Learning Research*, 5, 1177–1203.

Dean, J., and Ghemawat, S. (2004), "MapReduce: Simplified Data Processing on Large Clusters," in *OSDI'04: Sixth Symposium on Operating System Design and Implementation*, Berkeley: USENIX, pp. 137–150.

Fawcett, T. (2003), "ROC Graphs: Notes and Practical Considerations for Data Mining Researchers," Technical Report HPL-2003-4, HP Laboratories, available at *http://citeseer.ist.psu.edu/fawcett03roc.html*.

Friedman, N. (2004), "Inferring Cellular Networks Using Probabilistic Graphical Models," *Science*, 303, 799–805.

Friedman, N., and Koller, D. (2003), "Being Bayesian About Network Structure: A Bayesian Approach to Structure Discovery in Bayesian Networks," *Machine Learning*, 50, 95–126.

Friedman, N., Nachman, I., and Peér, D. (1999), "Learning Bayesian Network Structure From Massive Datasets: The 'Sparse Candidate' Algorithm," in *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA: Morgan Kaufmann, pp. 206–215.

Goldszmidt, M., and Pearl, J. (1992), "Default Ranking: A Practical Framework for Evidential Reasoning, Belief Revision and Update," in *Proceedings of the 3rd International Conference on Knowledge Representation and Reasoning*, San Mateo, CA: Morgan Kaufmann, pp. 661–672.

Hartemink, A. J. (2001), "Principled Computational Methods for the Validation of and Discovery of Genetic Regulatory Networks," unpublished doctoral thesis, Massachusetts Institute of Technology.

Holland, P. W. (1988), "Causal Inference, Path Analysis, and Recursive Structural Equations Models," in *Sociological Methodology*, ed. C. C. Clogg, Washington, DC: American Sociological Association, pp. 449–484.

Ide, J. S., and Cozman, F. G. (2002), "Testing MCMC Algorithms With Randomly Generated Bayesian Networks," in *I Workshop de Teses e Dissertaqes em Inteligência Artificial*, Recife, Pernambuco, Brazil. Available at *http://www.pmr.poli.usp.br/ltd/People/jside/IdeCozman_wtdia02.pdf*.

Kou, S., Zhou, Q., and Wong, W. H. (2006), "Equi-Energy Sampler: Applications in Statistical Inference and Statistical Mechanics," *The Annals of Statistics*, 34, 1581–1619.

Lauritzen, S. L. (1996), *Graphical Models*, Oxford, U.K.: Clarendon Press.

Lauritzen, S. L., and Spiegelhalter, D. J. (1988), "Local Computations With Probabilities on Graphical Structures and Their Application to Expert Systems," *Journal of the Royal Statistical Society*, Ser. B, 50, 157–224.

Murphy, K. P. (2001), "Bayes Net Toolbox for MATLAB," *Interface of Computing Science and Statistics*, 33, 331–350.

Neyman, J. (1990), "Sur les applications de la thar des probabilities aux experiences Agaricales: Essay des principle" [English translation of excerpts by D. Dabrowska and T. Speed], *Statistical Science*, 5, 463–472.

Pearl, J. (1988), *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, San Mateo: Morgan Kaufmann.

——— (1993a), "Aspects of Graphical Models Connected With Causality," in *Proceedings of the 49th Session of the International Statistical Institute*, Florence, Italy: International Statistical Institute, pp. 399–401.

——— (1993b), "Graphical Models: Causality and Intervention," *Statistical Science*, 8, 266–273.

——— (2000a), *Causality: Models, Reasoning and Inference*, Cambridge, U.K.: Cambridge University Press.

——— (2000b), "The Logic of Counterfactuals in Causal Inference," *Journal of the American Statistical Association*, 95, 428–435.

Peér, D. (2005), "Bayesian Network Analysis of Signaling Networks: A Primer," *Science STKE*, 281:PL4, 1–12.

Robins, J. (1986), "A New Approach to Causal Inference in Mortality Studies With Sustained Exposure Periods: Application to Control of the Healthy Worker Survivor Effect," *Math Modeling*, 7, 1393–1512.

——— (1987), "A Graphical Approach to the Identification and Estimation of Causal Parameters in Mortality Studies With Sustained Exposure Periods," *Journal of Chronic Diseases*, 40, 1395–1615.

Robins, J. M., and Wasserman, L. (1999), "On the Impossibility of Inferring Causation From Association Without Background Knowledge," in *Computation, Causation and Discovery*, eds. P. Glymour and G. F. Cooper, Menlo Park, CA/Cambridge, MA: AAAI Press/MIT Press, pp. 305–321.

Rubin, D. (1978), "Bayesian Inference for Causal Effects: The Role of Randomization," *The Annals of Statistics*, 6, 34–58.

Sachs, K., Perez, O., Peér, D., Lauffenburger, D. A., and Nolan, G. P. (2005), "Causal Protein-Signalling Networks Derived From Multiparameter Single-Cell Data," *Science*, 308, 523–529.

Shen, X., and Wasserman, L. (2001), "Rates of Convergence of Posterior Distributions," *The Annals of Statistics*, 29, 687–714.

Spirtes, P., Glymour, C., and Scheines, R. (1993), *Causation, Prediction, and Search* (2nd ed.), Cambridge, MA: MIT Press.

Wong, W. H., and Shen, X. (1995), "Probability Inequalities for Likelihood Ratios and Convergence Rates of Sieve MLEs," *The Annals of Statistics*, 23, 339–362.

Wright, S. (1923), "The Theory of Path Coefficients: A Reply to Niles' Criticism," *Genetics*, 8, 239–255.