Learning Causal Bayesian Network Structures from Experimental Data

Byron Ellis

Baxter Labs in Genetic Pharmacology

Stanford University Medical School

Stanford University, Stanford, CA 94305

email: `bcellis@stanford.edu`

Wing Hung Wong

Department of Statistics

Stanford University, Stanford, CA 94305

email: `whwong@stanford.edu`

Author's Footnote:

Abstract

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

Keywords: MCMC, equi-energy sampling, flow cytometry

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 Bayesian Network has become popular as an analytical framework in causal studies, where the causal relations are encoded by the structure (or topology) of the network. However, despite significant recent progress in algorithm development, the computational inference of network structure is currently still very much an open challenge in computational statistics that has remained infeasible except for cases with a very small number of variables. The purpose of this paper is to develop the statistical methods and computational algorithms for learning Bayesian Network structures from experimental data. In section 2, we review the theory of 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 we 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 samplers ability to cross energy barriers, we developed a novel "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 the effectiveness of the method. On random BN of various sizes, our approach was able to predict causal edges with AUC (area under the ROC curve) approaching 95% while the AUC for direct graph samplers 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 current interest to the emerging field of systems biology. Finally, in section 7, we discuss direction for future research and possible extensions of the methodology.

2. BAYESIAN NETWORKS IN CAUSAL INFERENCE

2.1 Graphical Models

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

$$P(V) = \prod_{i=1}^{n} P\left(V_i \mid \Pi_i^{\mathcal{G}}\right) \tag{1}$$

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, in order for (1) to define a joint distribution, a variable cannot serve as a child of its own descendants, i.e. $\mathcal{G}$ must be acyclic. Thus the structure graph $\mathcal{G}$ for a BN is a directed acyclic graph (DAG). There are many excellent treatments of graphical models and conditional independence structures e.g. Lauritzen and Spiegelhalter (1988), Lauritzen (1996), Pearl (2000*b*).

In this paper we study only the discrete case where 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 k^{th} joint state of its parents, V_i takes its j^{th} value with probability θ_{ijk}. Thus the BN with a fixed structure $\mathcal{G}$ is parameterized by the set of probabilities $\Theta = \left\{\theta_{ijk} \geq 0 : \sum_j \theta_{ijk} = 1\right\}$. 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} equals the number of times when 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

$(\theta_{i1k}, \theta_{i2k}, \ldots, \theta_{ir_ik})$. Thus the data distribution is a product multinomial

$$P(\mathbb{X}, \Theta, \mathcal{G}) = \prod_{i=1}^{n} \prod_{k=1}^{q_i} \binom{N_{i\cdot k}}{N_{i1k} N_{i2k} \cdots N_{ir_ik}} \theta_{i1k}^{N_{i1k}} \cdots \theta_{ir_ik}^{N_{ir_ik}} \tag{2}$$

Example 1 A 4-vertex Bayesian Network

A 4-vertex Bayesian Network, shown in Figure 1, is specified by

$$\begin{aligned} P(v_1, v_2, v_3, v_4) = P(V_2 = v_2 \mid V_3 = v_3, V_4 = v_4)\, P(V_3 = v_3 \mid V_4 = v_4) \\ \times P(V_4 = v_4 \mid V_1 = v_1)\, P(V_1 = v_1) \end{aligned} \tag{3}$$

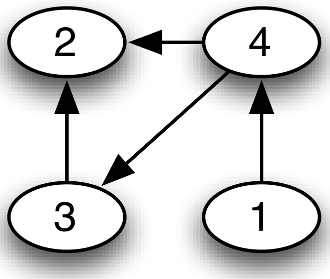

Figure 1: Structure of the Bayesian Network specified by (3).

Suppose 8 independent observations from (3) are obtained and shown in Table 1, then the sufficient statistics N_{ijk} can be obtained from simple counting in the table. For example, $N_{311} = \#\{V_3 = 1, V_4 = 1\} = 2$, $N_{24(1,3)} = \#\{V_2 = 4, (V_3, V_4) = (1,3)\} = 1$.

Since any discrete multivariate distribution can be represented by (1) and (2), the class of Bayesian Network models is too general to be useful, unless suitable assumptions are made on the network structure. 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, i.e. for some $0 < \gamma < 1$,

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

	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

Table 1: 8 observations from the graph defined by (3).

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

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

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

$$\begin{aligned} P(\Theta, \mathcal{G} \mid \mathbb{X}) = \prod_{i=1}^{n} \gamma^{|\Pi_i^{\mathcal{G}}|} \prod_{k=1}^{q_i} & \binom{N_{i\cdot k}}{N_{i1k} N_{i2k} \cdots N_{ir_ik}} \\ & \times \frac{\Gamma(\alpha_{i\cdot k})}{\Gamma(\alpha_{i1k}) \cdots \Gamma(\alpha_{ir_ik})} \\ & \times \theta_{i1k}^{N_{i1k}+\alpha_{i1k}-1} \cdots \theta_{ir_ik}^{N_{ir_ik}+\alpha_{ir_ik}-1} \end{aligned} \tag{6}$$

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

$$P(\mathcal{G} \mid \mathbb{X}) = \prod_{i=1}^{n} \gamma^{|\Pi_i^{\mathcal{G}}|} \prod_{k=1}^{q_i} \frac{\Gamma(\alpha_{i\cdot k})}{\Gamma(\alpha_{i\cdot k} + N_{i\cdot k})} \prod_{j=1}^{r_i} \frac{\Gamma(N_{ijk} + \alpha_{ijk})}{\Gamma(\alpha_{ijk})} \tag{7}$$

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

2.2 Causal Networks and Experimental Data

In general, 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 by a simple example, suppose we want to interpret the directed edges in Figure 1 as causal relations we will immediately run into the difficulty that $p(v_1, v_2, v_3, v_4)$ can also be written as

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

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

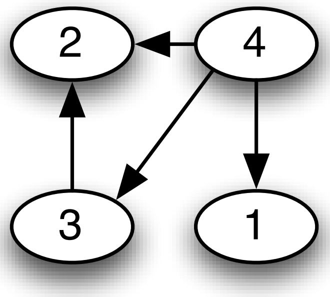

Figure 2: Network implied by an alternative decomposition.

Thus, the use of the BN model in causal inference depends on the existence of a special decomposition (assumed to be (3) in this example and (1) in general). The conditional probabilities in this special decomposition admit an interpretation that is independent of the joint distribution. In fact, we regard these probabilities, referred to as causal probabilities, as the building blocks that allow the specification of many related joint probability distributions under experimental intervention. For example, the causal probability $P(V_4 = a \mid V_1 = b)$ in the specification of $p(\cdot)$ in Example 1 is assumed to be the same as $P(V_4 = a \mid V_1 \text{ set to } b)$, i.e. 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 non-causal probability such as $P(V_1 = b | V_4 = a)$ can only be interpreted as a conditional probability derived from the joint distribution, i.e. 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 given in Example 1. Note that this is not the same as the probability $P(V_1 = b \mid V_4 \text{ set to } a)$, 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, unless the network structure is already known, the causal probabilities cannot be inferred from observational data alone. Causal relations can only be learned from data obtained through experimental interventions (experimental data). Such interventions typically allow us to fix the value of some of the variables and then observe what happens to the other variables. The question, then, is how to compute the posterior distribution for the structure $\mathcal{G}$ given such experimental data. Once the structure is known, it is easy to infer the causal probabilities.

The answer to this key question, first given in Cooper and Yoo (1999), is remarkably simple. To explain the logic, consider case three in Table 1 of Example 1. If this data point is observational, 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 the notation 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 to be 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 Graph-space Sampling

The size of the space of directed acyclic graphs is $2^{O(n^2)}$, which is super-exponential in n, the number of vertices of the graph (Chickering, Heckerman and Meek 2004). For example, for $n = 5, 10$ and 40 there are 29281, 4.17×10^{18} and 1.12×10^{276} graphs respectively.

While large, the space of DAGs is finite and in principle any graph can be reached by repeated applications of local moves chosen from a well-defined set. A minimal set of such moves include adding or deleting single edges (Chickering 2002). Let $[V_j, \ldots, V_i]$ denote a path of directed edges connecting V_j to V_i. We say that this path belongs to $\mathcal{G}$ if all edges in the path are directd edges in $\mathcal{G}$. Then, the set of local moves that add an an edge to graph $\mathcal{G}$ is

$$nbh\,(\mathcal{G})^{+} = \{\text{add } V_i \rightarrow V_j : \text{there is no path in } \mathcal{G} \text{ leading from } V_j \text{ to } V_i\} \tag{10}$$

and the set of local moves that removes an edge from $\mathcal{G}$ is

$$nbh\,(\mathcal{G})^{-} = \{\text{delete } V_i \rightarrow V_j : V_i \rightarrow V_j \in \mathcal{G}\} \tag{11}$$

the set of all possible local moves is defined as $nbh\,(\mathcal{G}) = nbh\,(\mathcal{G})^{+} \cup nbh\,(\mathcal{G})^{-}$.

Madigan and York (1995) proposed the use of the Metropolis-Hastings (MH) algorithm to sample structures according to their posterior distributions (e.g. (7)). Briefly, this algorithm works as follows: From a starting point $\mathcal{G}^{(t)}$ we propose a move uniformly at random from $M \sim nbh\left(\mathcal{G}^{(t)}\right)$ and apply it to the graph to obtain $\mathcal{G}'$ allowing $\mathcal{G}^{(t+1)}$ to be defined as the random variable

$$\mathcal{G}^{(t+1)} = \begin{cases} \mathcal{G}', & \min\left(1, \frac{\pi(\mathcal{G}')|nbh\left(\mathcal{G}^{(t)}\right)|}{\pi(\mathcal{G}^{(t)})|nbh(\mathcal{G}')|}\right) \\ \mathcal{G}^{(t)}, & \text{otherwise} \end{cases} \tag{12}$$

where $\pi(\mathcal{G}) \propto P\left(\mathcal{G} \mid \mathbb{X}\right)$.

The introduction of Markov Chain Monte Carlo methods offered an attractive approach to the computational inference of Bayesian network structures. Because many structures are generated in the sampling process, inference of graph features, such as the presence/absence of an edge, can be obtained by averaging over the sampled structures. For discussion of the advantage of model averaging over inference based on a single model (structure) obtained by heuristic optimization methods, see Cooper and Herskovits (1992) and Madigan, Raftery, York, Bradshaw and Almond (1994).

However, the algorithm only works well for network with a very small number of variables. As a simple example, we consider the space of 4 vertex graphs, which has 543 members allowing for brute force calculations. Figure 3 shows two sets of chains. The first chain is from 200 simulated observations on a simple posterior in the space of 4 vertex DAGS while the second set of 4 chains are run using 750 simulated observations. We see that as the sample size increases, the chains mix very slowly due to the more peaky nature of the posterior in large samples, causing the MH sampler to be trapped in local modes.

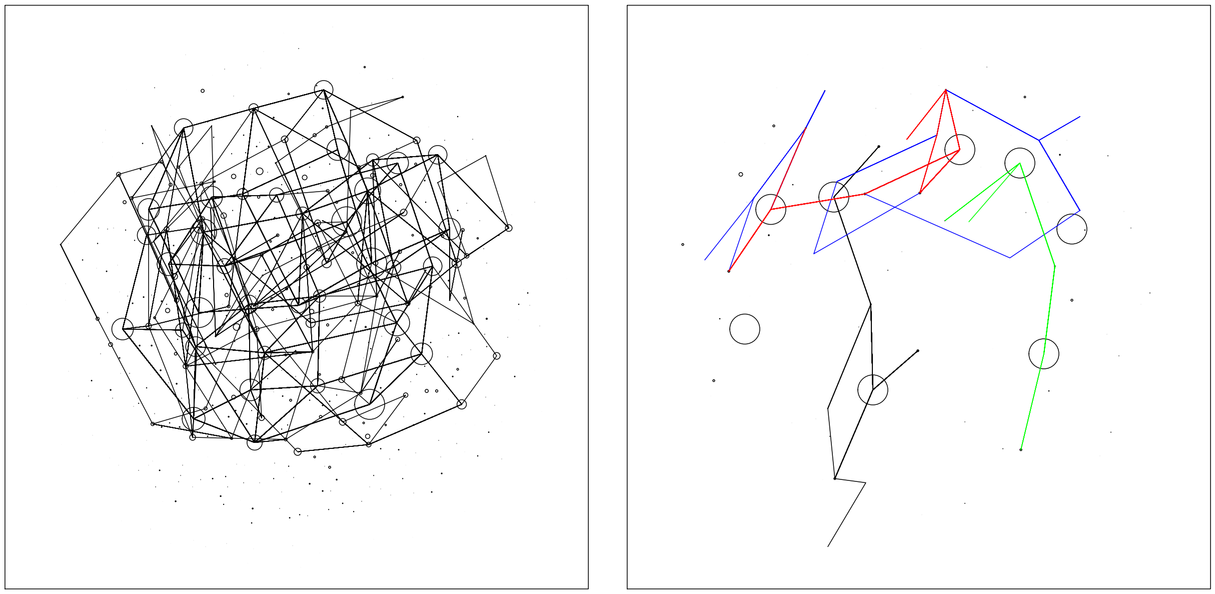

Figure 3: (a) An MCMC of 1000 steps on a posterior distribution obtained from 200 observations.(b) Four MCMC chains of 1000 steps each on a posterior distribution obtained from 750 observations. There are 543 circles representing the graphs with circle size proportional to the probability of the graph.

3.2 Order-space Sampling

As directed acyclic graphs are equivalent to partial orderings of the vertices, there exists at least one total ordering of the vertices, denoted $\sqsubset$, 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(Cooper and Herskovits 1992) exploited this fact to reduce the amount of computation required to find an optimal graph by assuming the knowledge of the ordering of the vertices. Suppose 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 only chosen from elements that precede V_j in the ordering eliminating the need for cycle checking. Thus, the 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 the computation of the posterior probability of a given order $\sqsubset$ defined as a sum over all graphs consistent with this order:

$$P(\sqsubset) \propto \sum_{\mathcal{G} \in \mathcal{G}_{\sqsubset}} P(\mathcal{G}) = \sum_{\mathcal{G} \in \mathcal{G}_{\sqsubset}} \prod_{i=1}^{n} P\left(V_i, \Pi_i^{\mathcal{G}}\right) \tag{13}$$

where $\mathcal{G}_{\sqsubset}$ is defined as

$$\mathcal{G}_{\sqsubset} = \left\{\mathcal{G} : \Pi_i^{\mathcal{G}} \prec_{\sqsubset} V_i\right\} \tag{14}$$

Brute force summation is infeasible since 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 that this summation can also be computed efficiently based on a similar argument as in the optimization case, which the attribute to Buntine (1991). Specifically, Buntine (1991) showed that

$$P(\sqsubset) \propto \prod_{i=1}^{n} \sum_{\Pi_i \in \Pi_i^{\sqsubset}} P(V_i, \Pi_i) \tag{15}$$

where $\Pi_i^{\sqsubset}$ is the collection of admissible parents sets for i in the order $\sqsubset$. This reduces the computational complexity from $O\left(\binom{n}{k}^{n/2}\right)$ to $O\left(n\binom{n}{k}\right)$ where $1 \leq k \leq n-1$ 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-Hasting algorithm to sample the order $\sqsubset$ according to its probability, (15). Note that this step is only possible because $P\left(\sqsubset\right)$ 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, though we have not found that they do much to improve performance.

- Given an order $\sqsubset$ generated as above, sample a network structure G from the space of DAGS compatible with this order according to the conditional probability distribution. This is easy as each V_i can be independently sampled from $\sqsubset$ by only considering the proper term of the product in (15) to obtain each Π_i.

3.3 Order-Graph Sampling

While 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 is caused by 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 pre-specified 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 from that given by (13). 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 2 vertex problem, which has 3 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 $\frac{6}{8}\left(\frac{1}{6}+\frac{2}{6}\right) = \frac{3}{8}$ and $\frac{5}{8}$ respectively. Calculating the probabilities of the graph *via* the orderings gives

$$P(V_1 \to V_2) = P(V_1V_2)\,P(V_1 \to V_2 \mid V_1V_2) = \frac{3}{8}\frac{1}{3} = \frac{1}{8}$$

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

$$P(V_2 \leftarrow V_1) = P(V_2V_1)\,P(V_1 \leftarrow V_2 \mid V_2V_1) = \frac{5}{8}\frac{3}{5} = \frac{3}{8}$$

As we can see, the true probabilities are such that $P(\mathcal{G}_1) < P(\mathcal{G}_2) < P(\mathcal{G}_3)$ while 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 $\mathcal{G}'$ be a random graph generated by the FK algorithm. Then

$$E[f(\mathcal{G}')] = \frac{\sum_{\mathcal{G}} |\sqsubset_{\mathcal{G}}| f(\mathcal{G}) P(\mathcal{G})}{\sum_{\mathcal{G}} |\sqsubset_{\mathcal{G}}| P(\mathcal{G})} \tag{16}$$

Proof.

$$\begin{aligned}
E\left[f(\mathcal{G}')\right] &= \sum_{\sqsubset} \frac{P(\sqsubset)}{\sum_{\sqsubset} P(\sqsubset)} \sum_{\mathcal{G}\in\mathcal{G}^{\sqsubset}} f(\mathcal{G}) \frac{P(\mathcal{G})}{\sum_{\mathcal{G}\in\mathcal{G}^{\sqsubset}} P(\mathcal{G})} \\
&= \frac{\sum_{\sqsubset}\sum_{\mathcal{G}\in\mathcal{G}^{\sqsubset}} f(\mathcal{G}) P(\mathcal{G})}{\sum_{\sqsubset} P(\sqsubset)} \\
&= \frac{\sum_{\mathcal{G}} \left|\sqsubset_{\mathcal{G}}\right| f(\mathcal{G}) P(\mathcal{G})}{\sum_{\mathcal{G}} \left|\sqsubset_{\mathcal{G}}\right| P(\mathcal{G})}
\end{aligned}$$

□

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

$$P(\mathcal{G}') \propto \mid \sqsubset_{\mathcal{G}} \mid P(\mathcal{G}) \tag{17}$$

which is generally biased due to the extra factor $|\sqsubset_{\mathcal{G}}|$. In small problems like our 4 vertex example, we know how many orders are consistent with each graph allowing us to correct the bias introduced by the overlap. However, in large problems the calculation of the overlap is infeasible as 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

$$\sum_{i=1}^{k} P(\mathcal{G}_i \mid \sqsubset) \geq (1-\varepsilon) \tag{18}$$

Let $\mathcal{U}$ be the union of the graph samples from all orders. We treat $\mathcal{U}$ as an importance weighted sample, where the weight from a graph $\mathcal{G}_i$ in $\mathcal{U}$ is $\frac{P(\mathcal{G}_i)}{\sum_{\mathcal{G}\in\mathcal{U}} P(\mathcal{G})}$ and $P(\mathcal{G})$ is given by (7). By ensuring that we sample most of the graphs that make up the posterior score of an order and assuming that our sampler obtains samples from high scoring regions of the order space, which contain these high probability graphs, we can assure ourselves that we have sampled most of the high scoring regions of the graph space as well. Figure 4 shows that in the 4 vertex problem, this method eliminates the bias induced by order sampling.

To assess the waiting time until (18) to have been satisfied, let g_i, $i = 1, \ldots, I$ be the distinct graphs in $\sqsubset$ ordered such that $p_i = P(g_i \mid \sqsubset)$ satisfy $p_1 \geq p_2 \geq \cdots \geq p_I$. Draw graphs from $\sqsubset$, with replacement, according to $\underset{\sim}{p}$.

Let

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

which has expectation $E\left[I_i(n)\right] = (1-p_i)^n$.

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 $E\left[U_n\right] = \sum_{i=1}^{I} p_i(1-p_i)^n$.

Now, assume that $p_{i+1} \leq \alpha p_i$ for some $0 < \alpha < 1$ and choose k large enough to satisfy (18), namely,

$$k \geq \frac{\log\left(\frac{\varepsilon(1-\alpha)}{p_1}\right)}{\log(\alpha)}$$

Then,

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

choosing n so that $k(1-p_k)^n \leq \varepsilon$ it follows that if

$$n \geq \frac{\log\left(\frac{\varepsilon}{k}\right)}{\log\left(1-p_k\right)}$$

then we have

$$E\left[U_n\right] \leq 2\varepsilon \tag{19}$$

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

Lemma 2

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

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

where

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

then $P\left(U_n > \gamma\right) \leq 2\gamma$

Proof. Applying the Markov Inequality we have

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

where $E\left[U_n\right] \leq 2\varepsilon$ from (19). The result follows by setting $\varepsilon = \gamma^2$. $\square$

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 0.81. Since sampling graphs given an order is a fast computation, this step consumes negligible time compared to the time required to sample the necessary orders. Finally, we note that, before sampling, one can first compute the graph with the highest probability within an order. This can often reduce the number of samples required for (18). To see the difference between the biased Order MCMC results and our bias-corrected samples we can compute the posterior probability for all graphs in the 4 vertex problem exactly as well as using Order MCMC and Order-Graph, shown in Figure 4.

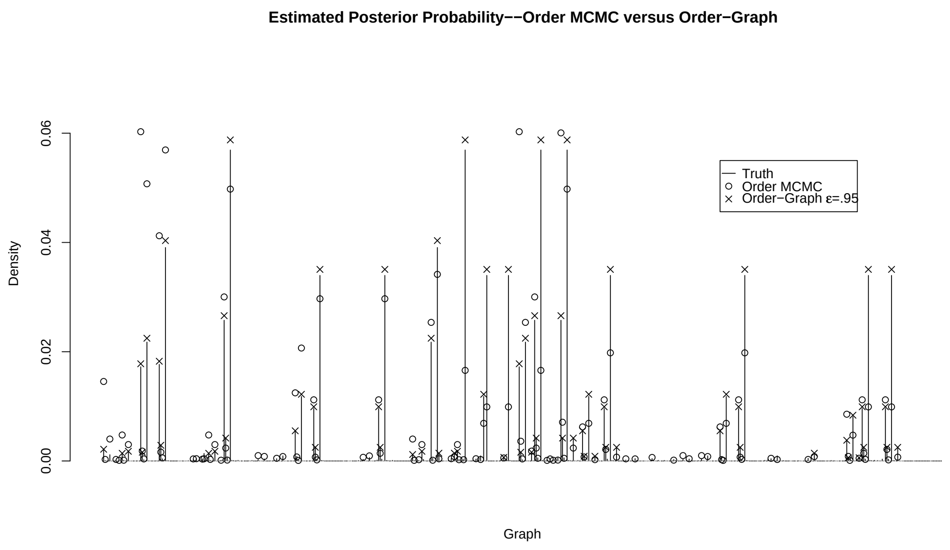

Figure 4: Order MCMC posterior probability estimates and Order-Graph posterior probability estimates, as compared to the true posterior probabilities for the 4 vertex problem.

4. IMPLEMENTING THE ORDER-GRAPH SAMPLER

4.1 Single Queue Equi-Energy Sampling

While the Order MCMC approach greatly improves mixing compared to MCMC in the space of DAGs it is 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 5, which shows the energies of 10 different chains run for $100,000$ iterations. We can also see this problem in simulated data, such as the 32 chains of a simulated 10 vertex data set shown in Figure 6.

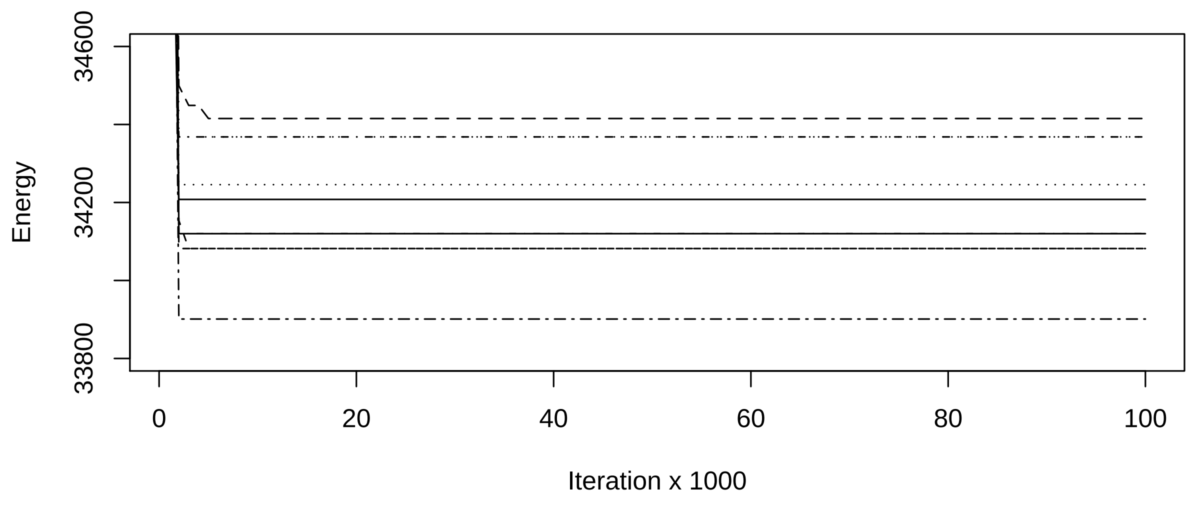

Figure 5: Energies for samples from 10 runs of the Order MCMC sampler in an 11 vertex problem. As we can see there are several local minima for the sampler often quite far from the global minimum energy, which is discovered by one run of the sampler. Though not visible due to scale, each chain is still moving in a small region around the trapping point with an acceptance rate of around 30%.

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 have chosen to use Equi-Energy sampling(Kou, Zhou and Wong 2006) for our software implementation. Briefly, like other population-based MCMC approaches the Equi-Energy sampler relies on a temperature

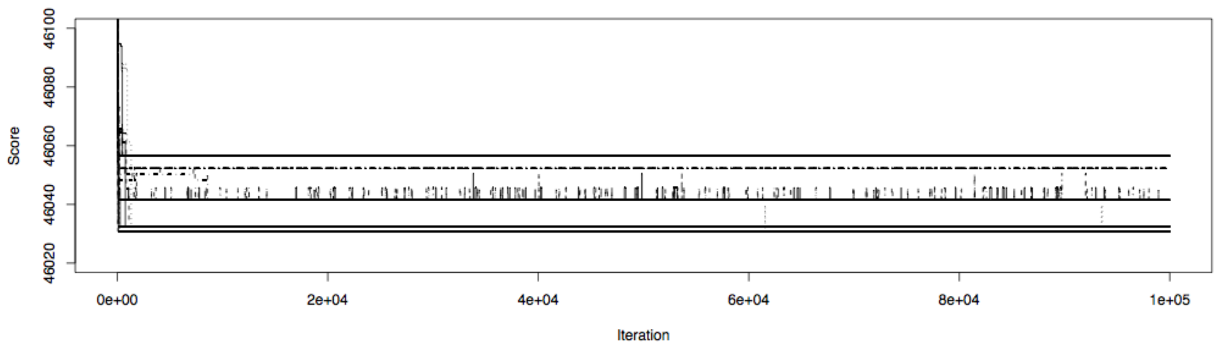

Figure 6: Energies for another set of Order MCMC chains run on a simulated 10 vertex data set. In this case 32 chains are run for 100, 000 iterations with a similar trapping phenomenon. In this data set the trapping locations are apparently closer in order space allowing for some crossing between local minima. However, most of the chains once trapped in a local minima tend to stay there.

ladder $1 = T_1 < \cdots < T_I$ that are used to define I Boltzmann distributions. In addition, the EE Sampler uses a ladder of truncation energies $H_{min} = H_1 < \cdots < H_I < \infty$ such that tempered distributions are defined as,

$$\pi_i(x) = \exp\left(\frac{-\max\left[H(x), H_i\right]}{T_i}\right) \tag{20}$$

Each chain is then started in a staggered fashion such that the i^{th} chain starts $B + N$ iterations after the $i+1$ chain, where $B+N$ are the number of burn-in and collected iterations of the chain of interest $\pi_1(x)$. The chains themselves run using the normal Metropolis-Hastings moves, though better results are usually obtained from scaling the step-size of the higher temperature chains as 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 Order-Graph Sampler the step size adjustment takes the form of a generalized swap move. In the i^{th} 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 "cynlindrical 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 this highest temperature and lowest temperature chains. In our experience, this simple rule gives satisfactory results for most problems.

In the original Equi-Energy sampler, communication between the i^{th} and the $i+1^{th}$ chain takes places through the use of a set of energy rings. After burn-in, the samples from the i^{th} chain are recorded in a set of energy bins $\tilde{D}_i^0, \ldots, \tilde{D}_i^I$ where $\tilde{D}_i^j$ contains samples with energies in the range $(H_i, H_{i+1}]$ with $H_0 = -\infty$ and $H_{I+1} = \infty$. These bins are used to help creating fast mixing for the $i+1^{\text{th}}$ chain through the equi-energy jump. In addition to the usual Metropolis moves, the $i+1^{\text{th}}$ chain will with probability q propose an equi-energy jump. The jump is proposed by selecting a sample y from the same bin that would hold the i^{th} chain's current state $x_i^{(t)}$ from among the $\tilde{D}_i^0, \ldots, \tilde{D}_i^I$ bins. This ensures that y will be relatively close in the energy space while potentially quite distant in the state space, such as the space of total orderings or graphs.

However, as shown in Figure 7, having separate sets of energy bins between each chain can still slow down communication of low energy states discovered in high temperature chains to the low temperature chains. This seems to be especially prevalent in situations such as Order sampling where we observe empty regions of the energy space, particularly between a small set of very low energy states and a much larger set of states with relatively higher energy. To allow for instant communication between all of the chains we modify the standard Equi-Energy sampler with a variant we call the "Single Queue Variant." In this variant, rather than a set of rings between each pair of chains, the i^{th} chain can now propose a sample from any of the higher temperature chains by having the i^{th} chain draw samples from the union of the relevant energy bin from all higher temperature chains. This requires a change to the transition kernel of the energy jump acceptance probability calculations as the samples are now drawn from a mixture of all higher temperature distributions rather than a single distribution. We track the number of samples, M_i, placed into the energy rings

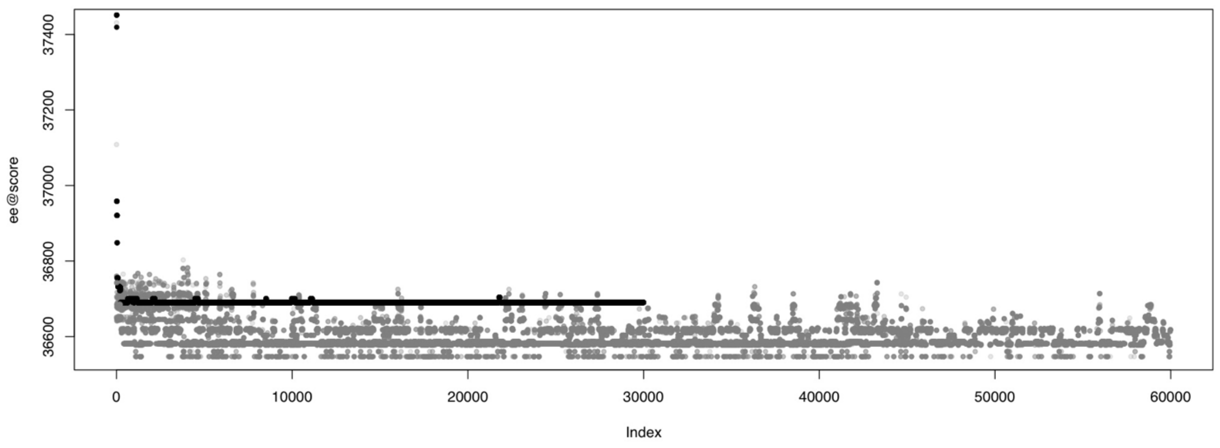

Figure 7: An example of the Order-Graph Sampler's Equi-Energy implementation becoming trapped in an 11 node problem when using the standard implementation of the Equi-Energy Sampler. Despite the fact that higher temperature chains, shown in gray, find a "good" set of states, the temperature of interest, shown in black, fails to converge properly. The reason is that a low energy states discovered by a high temperature chain is only used in the equi-energy jumps of the next highest energy chain, it is not directly accessible by lower temperature chains.

by each chain, so that the proposal transition kernel is now computed as

$$Q(y; x_i^{(t)}) = \sum_{j=i+1}^{I} w_j \pi_j(y) \tag{21}$$

where $w_j = M_j / \sum_{k=i+1}^{I} M_k$.

4.2 Efficient Order Scoring

The key to any Order sampler, be it Friedman-Koller Order MCMC or the population-based samplers of Order-Graph MCMC, 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 it's 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 so this scheme is sufficient for fast calculation. Order scoring, on the other hand, can result in a large number of changes and each scoring pass involves a large number of family checks making the cache-based structure a less obvious choice for efficient calculation.

Our implementation instead relies on a stream-based approach similar to the Google MapReduce (Dean and Ghemawat 2004) architecture. We precompute 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^j and a vector of n values containing $\log P(V_i, \Pi^j)$ or $-\infty$ if $i \in \Pi^j$. To calculate an order probability, consider the total ordering, $\sqsubset$. We begin by initializing an intermediate probability vector $(s_1, \ldots, s_n)$ for each $V_1, \ldots, V_n$. Then we calculate the position vector $(p_1, \ldots, p_n)$ such that p_i is the index of V_i in $\sqsubset$. We then consider each record in the database in an arbitrary order, possibly in parallel, first calculating the rightmost position of the elements in Π^j, $p_{max} = \max_{k \in \Pi^j} p_k$. We then update the intermediate scoring vector as follows,

$$s_i = \log\left(\exp(s_i) + P\left(V_i, \Pi^j\right)\right) \quad i = \sqsubset_{p_{max}}, \ldots, \sqsubset_n \tag{22}$$

for each entry and finally calculate $\log P(\sqsubset) = \sum_{i=1}^{n} s_i$. An example of a single step in this process is given in Figure 8.

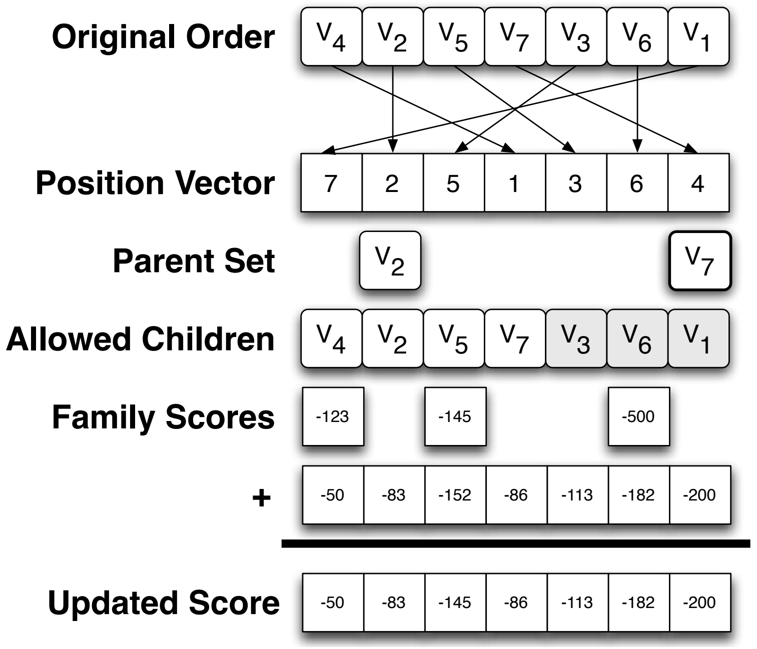

Figure 8: A single step in the order scoring process.

This approach achieved its performance through the use of a large set of pre-computed scores that are likely to be needed and eliminating the large number of random access lookups that are 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). Additionally, 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, for example, accommodate parent set sizes larger than k in place of smaller, but low scoring, parent sets over time.

5. ALGORITHMIC VALIDATION

To validate our algorithm's performance we conducted a number of simulation experiments using our algorithm and several other approaches that work directly in the DAG space. In these experiments we generated random graphs having either 8, 10, 12 or 14 vertices and $5,000$ observations from each random graph. We then sample each set of observations to obtain artificial sample sizes of 100, 500, 1000, 2500 as well as the full 5000 observation datasets. To obtain an average performance each combination of vertices and observations is replicated 5 times with different values. In all cases, a prior of $P(\mathcal{G}) = \gamma^{-|\mathbb{E}|}$ with $\gamma = 10$ was used.

For comparison we employ three different algorithms. The first, of course, is our Order-Graph Sampler using the Equi-Energy sampler for the Order MCMC phase. The second is a direct DAG sampler using the Equi-Energy approach to improve mixing. Finally, we use an importance re-weighting scheme similar to the one employed by the second phase of the OGEE sampler. In this sampler we simply use a Greedy Search algorithm to generate "optimal" graphs from a large number of random starting locations. We then re-normalize this collection of graphs to obtain a properly weighted distribution on graphs.

In all three cases we choose 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 is roughly matched running on identical hardware and identical operating systems (an Apple XServe cluster in this case). The DAG EE sampler was run for $100,000$ burn-in iterations and $100,000$ iterations collected for processing, this being the fastest algorithm for a single iteration. The greedy search algorithm was run with $100,000$ unique starting locations to run in roughly the same amount of time. Finally, the OGEE sampler was run for $15,000$ burn-in order samples and $15,000$ collected order samples. The graph sampling process given an order requires a negligible amount of time with at most 1000 graphs sampled from each order and merged into a unique set.

	8	10	12	14
100	0.73	0.69	0.73	0.71
500	0.82	0.81	0.90	0.87
1000	0.87	0.84	0.92	0.89
2500	0.88	0.86	0.93	0.89
5000	0.91	0.91	0.97	0.94

Table 2: The mean AUC for the ROC curves obtained from the OGEE Sampler for each combination of vertices and observations.

5.1 Experimental Results

To compare these posterior samples, we employed a popular technique described in the original Order MCMC paper (Friedman and Koller 2003). In this case, the posterior graph sample is used to calculate pairwise feature probabilities for the entire graph (i.e. $P(V_i \rightarrow V_j)\ \forall i, j$). These marginal features are then rank-ordered and a cutoff-threshold is selected and the features above that threshold are used to construct a final graph.

By allowing this threshold to vary from 1 down to 0, we can construct receiver operator curves (ROC), which is a common method for assessing these sorts of threshold classifiers (Fawcett 2003) by calculating the area under this curve (AUC) with 1.0 being an ideal classifier. Typically, an AUC above 0.9 considered to be "excellent" and an AUC above 0.8 considered to be "good." Tables 2, 3 and 4 show the mean of 5 AUC calculations for each combination of nodes and data under each of the three algorithms. As we can see, the OGEE algorithm performs quite well even with relatively moderate amounts of data achieving "good" classification results quite quickly, handily outperforming both the DAG EE and importance re-weighting algorithms.

Interestingly, the importance re-weighting technique seems to in turn outperform the DAG EE approach. The reason for this seems to be that the DAG EE sampler is still not using enough starting locations and the algorithm becomes trapped at relatively low

	8	10	12	14
100	0.69	0.64	0.62	0.67
500	0.70	0.72	0.71	0.72
1000	0.77	0.72	0.75	0.73
2500	0.77	0.77	0.74	0.72
5000	0.73	0.76	0.76	0.74

Table 3: The mean AUC for the ROC curves obtained from the direct importance sampling method for each combination of vertices and observations.

	8	10	12	14
100	0.48	0.51	0.47	0.50
500	0.46	0.55	0.50	0.52
1000	0.49	0.54	0.52	0.51
2500	0.48	0.58	0.50	0.50
5000	0.49	0.58	0.50	0.53

Table 4: The mean AUC for the ROC curves obtained from the Equi-Energy DAG sampler for each combination of vertices and observations.

probability locations. The importance re-weighting algorithm, on the other hand, gets to attempt a large number of starting locations and, typically, "gets lucky" by finding one particularly high probability graph that will dominate the posterior probability. In effect, the sample is more like a single incredibly fortunate sample rather than the $100,000$ graphs that have actually been sampled.

6. AN APPLICATION TO FLOW CYTOMETRY DATA

6.1 Introduction

In this example, we will investigate 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 staining each protein with a fluorescent marker protein which can then be scanned by a laser tuned to a particular wavelength as each cell flows past in a column of fluid. Importantly, this does not require cell lysis as is the case in microarray experiments so data is collected on a cell by cell basis, leading to large amounts of data in each experiment.

The researchers in this particular experiment chose to investigate elements of the three major Mitogen-Activated Protein Kinase (MAPK) pathway in human CD4+ T-cells, part of the immune system. In general, MAPK pathways are involved in a process known as "signal transduction" (Figure 9). External stimuli are sense by cell surface markers (such as the CD3 and CD28 in Figure 10). This signal then travels through a cascade of protein modifications (deactivation and activation) of signaling proteins known, generically, as MAPKKK, MAPKK and MAPK, which eventually leads to changes in nuclear transcription.

In this particular experiment, 11 well-studied proteins were selected from the MAPK pathways for fluorescent labeling. This pathway was then perturbed by 9 different stimuli, each targeting a different protein in the selected pathway. The pathway itself is shown in Figure 10 with the appropriate proteins highlighted, while the experimental stimuli and their effects are summarized in Table 5.

6.2 Original Analysis

The original analysis conducted by Sachs et al. (2005) was also a Bayesian Network reconstruction. In this case they employed the Simulated Annealing approach discussed briefly

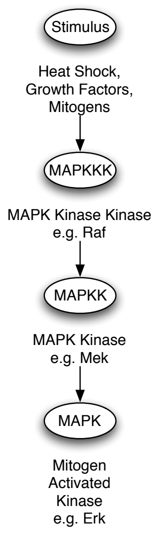
Stimulus
Heat Shock, Growth Factors, Mitogens
MAPKKK
MAPK Kinase Kinase e.g. Raf
MAPKK
MAPK Kinase e.g. Mek
MAPK
Mitogen Activated Kinase e.g. Erk

Figure 9: An overview of the basic structure of MAPK pathways.

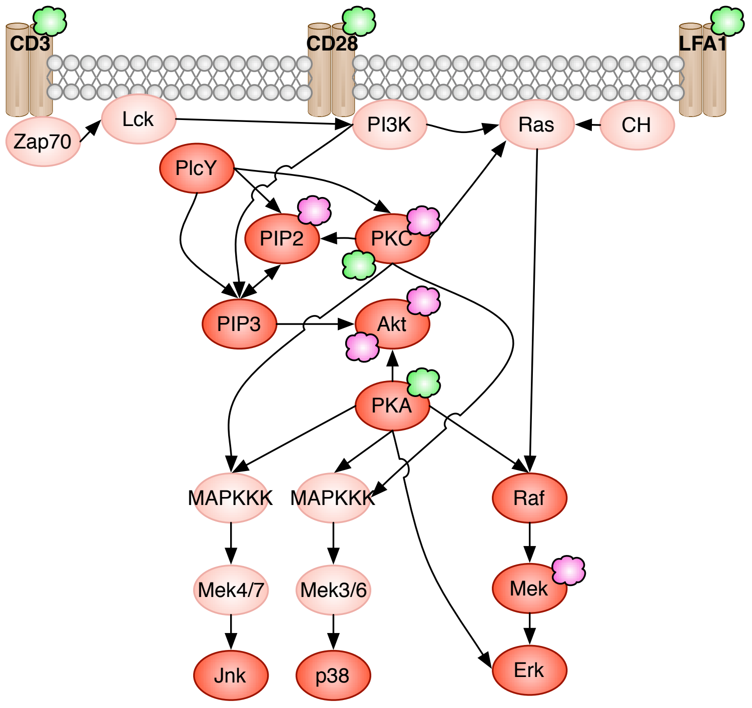

Figure 10: The known signal transduction experiment. Proteins being probed are indicated by the darker shading, while those involved in the pathway but un-probed are indicated by lighter shading.

	Stimulus	Effect
1.	CD3, CD28	General stimulation
2.	ICAM2	General stimulation
3.	Akt-inhibitor	Inhibits Akt (-Akt)
4.	G0076	Inhibits PKC (-PKC)
5.	Psi	Inhibits PIP_2 (-PIP_2)
6.	U0126	Inhibits Mek (-Mek)
7.	Ly	Inhibits Akt (-Akt)
8.	PMA	Activates PKC (+PKC)
9.	$\beta_2 cAMP$	Activates PKA (+PKA)

Table 5: A summary of the 9 experimental stimuli and their effect on the proteins of interest.

in Chapter 3. The original data set, shown in Figure 11 is continuous data that was processed by first eliminating outliers, we assume after a log-scale transform though this is not explicitly stated. The data were then discretized using an information preserving technique (Hartemink 2001) and adjusted such that perturbed vertices always reflect their perturbed values. This is necessary because some of the perturbation methods, particularly the inhibition, only affect the activity of the protein and not their phosphorolation so while the activity of the protein appears to vary, its activity is in fact controlled.

After pre-processing, 500 data sets are generated. Each data set consists 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 have a set of 500 DAGs each with an associated score. To estimate the marginal feature probabilities Bootstrap samples are generated from this data set according to their scores.

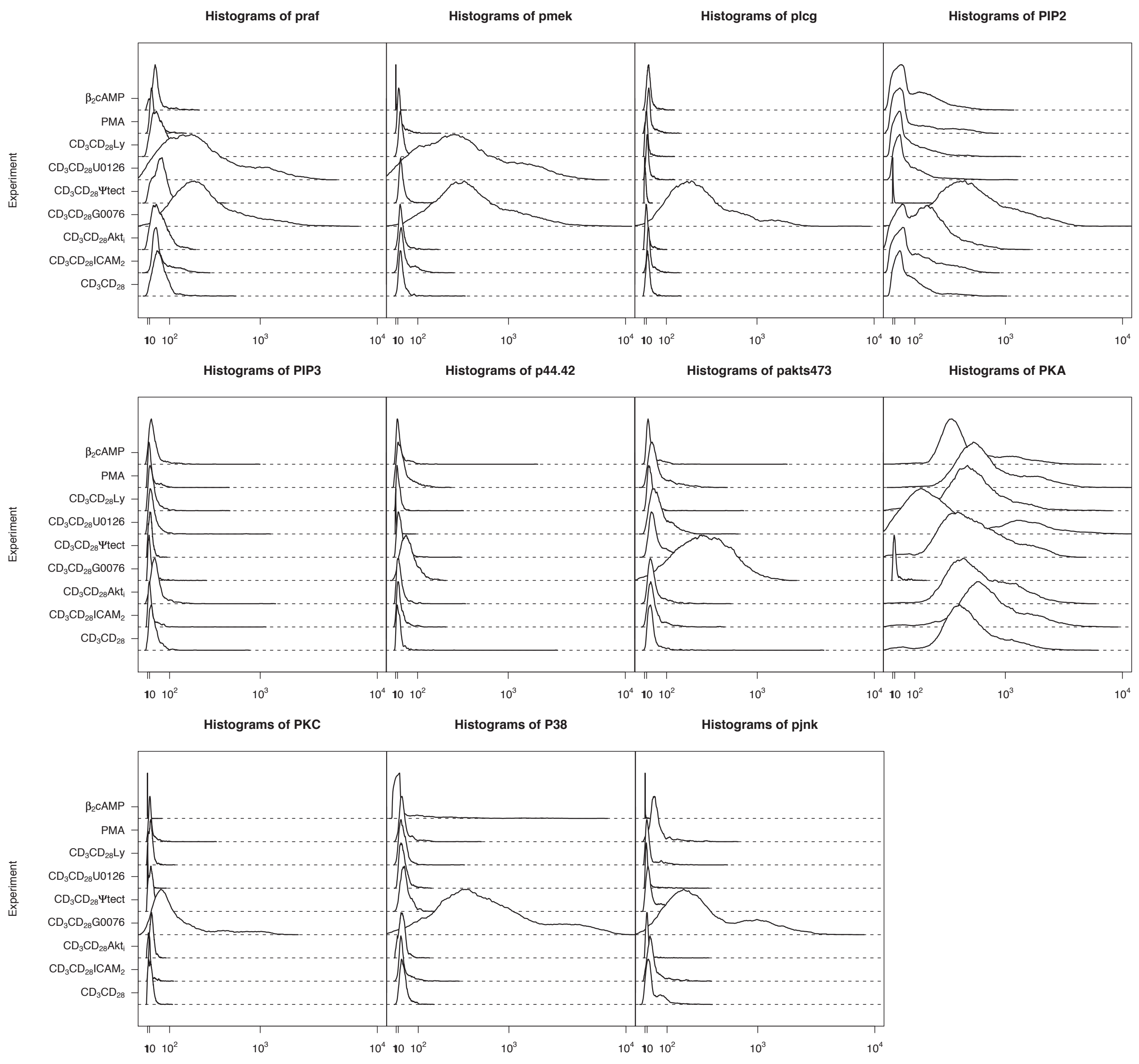

Figure 11: The data original data after a standard "Logicle" transformation, which is more or less a log transform when values are far away from 0 and linear close to 0. This has the effect of compressing the first two order of magnitudes into an area quite close to 0, which is consistent with a correction for an artifact introduced by the flow cytometry process that otherwise appears as a spurious second clustering of cells.

6.3 Analytic Setup

Much like our experiment with the random graphs we have created a competitive experimental setting for our analysis, though we will be using a Bootstrap-based technique similar to the original analysis rather than the Parallel Tempering or DAG Equi-Energy approaches. Our Bootstrap implementation follows as closely as possible (Friedman, Nachman and Peér 1999) though we employ the same efficient scoring and database mechanisms as our Order-Graph sampler to ensure fair comparisons between running time and other performance metrics.

For the simulated annealing technique we use parameters as close as possible to the original Sachs analysis, in this case 500 separate graphs will be generated from datasets sampled with replacement from the complete data. Each annealing chain is then run for $100,000$ iterations with a starting temperature of 50 and a finishing temperature of 1 with a log cooling schedule.

6.4 Results

With no "true" graph available as was the case in simulation, we compare the algorithmic performance using a 10-fold cross validation approach. To simplify the cross validation, a single data set consisting of 600 cells from each experiments is generated and that data set is divided into 10 subsets containing the same number of cells from each experiment (i.e. each subset contains 60 observations from each of the 9 experiments). Nine of these subsets are then used to estimate a posterior distribution of graphs which are then used to predict the remaining 10^{th} data set. In the case of the simulated annealing algorithm, which requires resampling, the 500 data sets are sampled with replacement from the 9 subsets rather than the entire data as was done in the original analysis. This is repeated for each of the 10 subsets used for prediction for both algorithms. We can then obtain an average predictive probability via model averaging for each of the 10 subsets with the scores shown in Figure 12. As we can see, the Order-Graph sampler is consistently superior to the simulated annealing approach.

To get an idea of the graphs being considered by each algorithm we also constructed

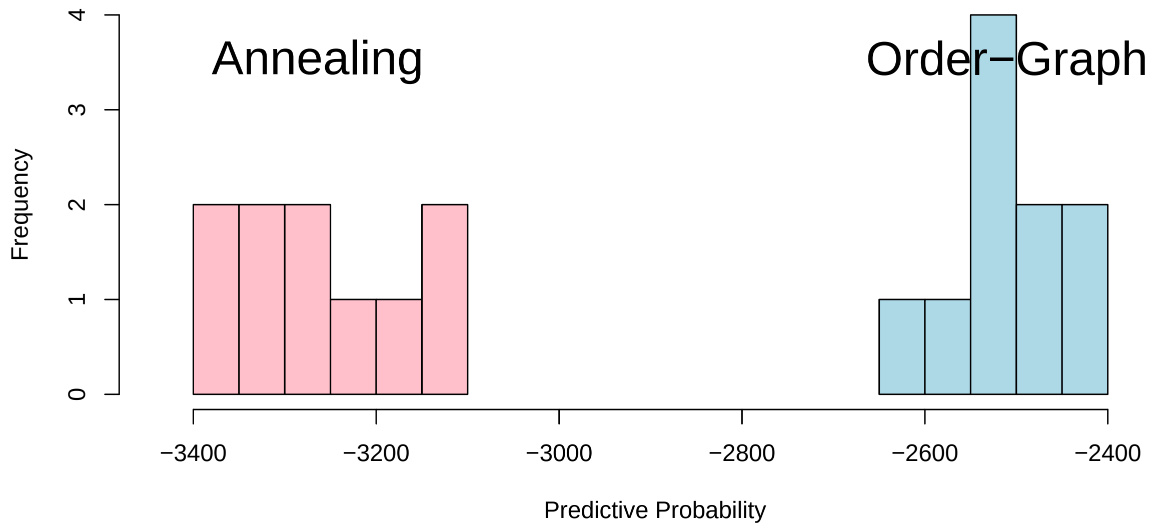

Figure 12: A histogram of the scores obtained by each algorithm during Cross Validation. The lighter bars are the simulated annealing results and the darker bars the Order-Graph results.

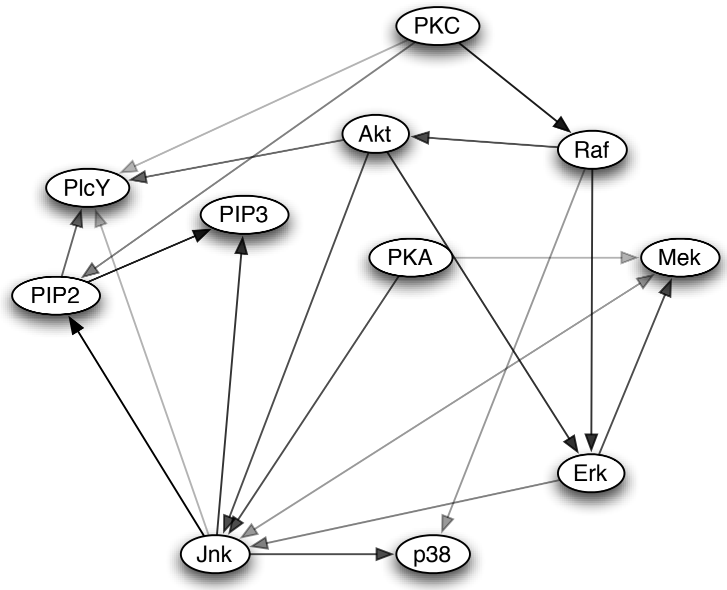

Figure 13: The simulated annealing graph reconstruction.

"mean graphs" for each algorithm by first constructing a mean graph according to each empirical distribution generated by our Cross Validation scheme. We then took the average of those 10 average graphs to obtain a global average for each algorithm. These graphs, both containing 22 edges, are shown in Figure 13 for simulated annealing and in Figure 14 for the Order-Graph sampler. The running time and cross-validation results are reported in Table 6, the most striking result being the time differential between the two techniques. To generate 500 graph samples via simulated annealing required between 3 and 4 average depending on server conditions whereas the Order-Graph sampler required between 3 and 5 minutes to generate $15,000$ graph samples. Additionally, when compared with the "known" graph shown in Figure 10 it seems that the Order Graph sampler's graph appears to be qualitatively closer to the graph obtained via literature search.

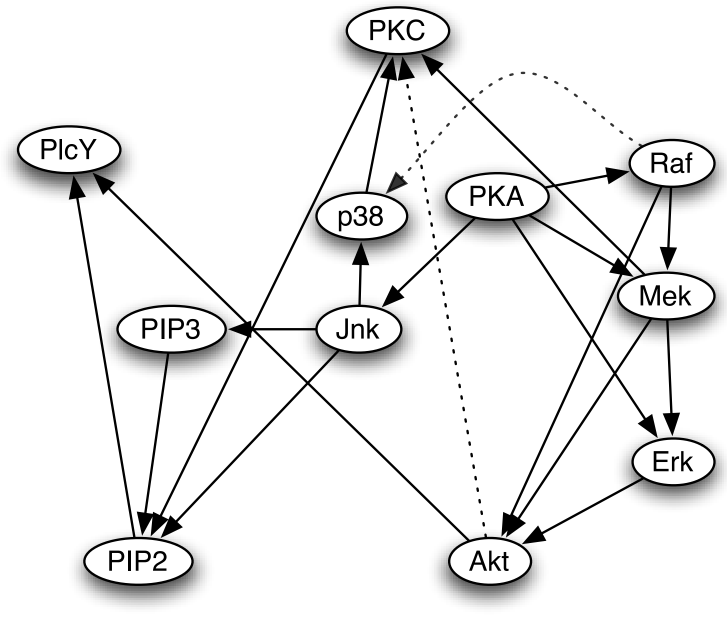

Figure 14: The average graph constructed from the average of each subset in the cross validation experiment

	Simulated Annealing	Order-Graph
Running Time	3-4 hours	3-5 minutes
Average Score	-3264.1 (93.52)	-2510.6 (63.39)

Table 6: The results for the cross validation experiment.

7. CONCLUSIONS AND FUTURE WORK

With the development of the Order-Graph sampling approach, we can now address complete data problems with reasonable confidence in problems of moderate size. This has been accomplished through a combination of methods: extending the already powerful Order MCMC approach with equi-energy sampling, bias-corrected graph sampling, and stream-based computation. Currently, our implementation pre-computes the scores for all (V_i, Π_i) combinations subject to a bound on $|\Pi_i|$. This bound on the maximum in-degree can be relaxed by adapting the database to exchange non-favorable combinations with selected combinations with higher in-degrees than the initial bound. This should be explored in future implementations.

Perhaps the most important future challenge is to consider the missing data case. As an example, we may want 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 is "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 posterior graph sample for all of the elements. The complete-data algorithms discussed in this paper 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.

8. ACKNOWLEDGMENTS

This research was supported by NSF grant DMS-0505732 to Wing Hung Wong. We would like to thank Rob Tibshirani for pointing us towards multiparameter flow cytometry as a rich source of data and Trevor Hastie for suggesting the cross-validation approach for assessing the quality of obtained graphs.

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*, Springer-Verlag.

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*, 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).*

Chickering, D. M. (2002), "Optimal Structure Identification With Greedy Search," *Journal of Machine Learning Research*, 3, 507–554.

Chickering, D. M., Heckerman, D., and Meek, C. (2004), "Large-Sample Learning of Bayesian Networks is NP-Hard," *Journal of Machine Learning Research*, 5, 287–1330.

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* (*Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence* 1999).

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.*

Fawcett, T. (2003), ROC Graphs: Notes and Practical Considerations for Data Mining Researchers,, Technical Report HPL-2003-4, HP Laboratories, 1501 Page Mill Road, Palo Alto, CA 94304.
URL: *citeseer.ist.psu.edu/fawcett03roc.html*

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

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* (*Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence* 1999).

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

Kou, S., Zhou, Q., and Wong, W. H. (2006), "Equi-Energy Sampler: Applications in Statistical Inference and Statistical Mechanics," *Annals of Statistics*, 34(4).
URL: *http://arxiv.org/abs/math.ST/0507080*

Lauritzen, S. L. (1996), *Graphical Models* Oxford: 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. Series B (Methodological)*, 50(2), 157–224.

Madigan, D., Raftery, A. E., York, J., Bradshaw, J. M., and Almond, R. G. (1994), Strategies for Graphical Model Selection,, in *Proceedings of the 4th International Workshop on Artificial Intelligence and Statistics*.

Madigan, D., and York, J. (1995), "Bayesian graphical models for discrete data," *International Statistical Review*, 63, 215–232.

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

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

Pearl, J. (2000*b*), "The Logic of Counterfactuals in Causal Inference," *Journal of the American Statistical Association*, 95(450), 428–435.

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

Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (1999).

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.