Being Bayesian About Network Structure

A Bayesian Approach to Structure Discovery in Bayesian Networks

Nir Friedman (nir@cs.huji.ac.il)
School of Computer Science & Engineering
Hebrew University
Jerusalem, 91904, Israel

Daphne Koller (koller@cs.stanford.edu)
Computer Science Department
Stanford University
Stanford, CA 94305-9010

Abstract. In many multivariate domains, we are interested in analyzing the dependency structure of the underlying distribution, e.g., whether two variables are in direct interaction. We can represent dependency structures using *Bayesian network* models. To analyze a given data set, Bayesian model selection attempts to find the most likely (MAP) model, and uses its structure to answer these questions. However, when the amount of available data is modest, there might be many models that have non-negligible posterior. Thus, we want compute the Bayesian posterior of a feature, i.e., the total posterior probability of all models that contain it. In this paper, we propose a new approach for this task. We first show how to efficiently compute a sum over the exponential number of networks that are consistent with a fixed order over network variables. This allows us to compute, for a given order, both the marginal probability of the data and the posterior of a feature. We then use this result as the basis for an algorithm that approximates the Bayesian posterior of a feature. Our approach uses a *Markov Chain Monte Carlo* (MCMC) method, but over orders rather than over network structures. The space of orders is smaller and more regular than the space of structures, and has much a smoother posterior "landscape". We present empirical results on synthetic and real-life datasets that compare our approach to full model averaging (when possible), to MCMC over network structures, and to a non-Bayesian bootstrap approach.

Keywords: Bayesian Networks, Structure Learning, MCMC, Bayesian Model Averaging

Abbreviations: BN – Bayesian Network; MCMC – Markov Chain Monte Carlo; PDAG – Partially Directed Acyclic Graph

1. Introduction

Bayesian networks (Pearl, 1988) are a graphical representation of a multivariate joint probability distribution that exploits the dependency *structure* of distributions to describe them in a compact and natural manner. A BN is a directed acyclic graph, in which the nodes correspond to the variables in the domain and the edges correspond to direct probabilistic dependencies between them. Formally, the structure of the network represents a set of *conditional independence assertions* about the distribution: assertions of the form the variables X and Y are independent *given* that we have observed the values of the variables in some set $\mathbf{Z}$. Thus, the network structure allows us to distinguish between the simple notion of correlation and the more interesting notion of direct dependence; i.e., it allows us to state that two variables are correlated, but that the correlation is an indirect one, mediated by other variables. The use of conditional independence is the key to the ability of Bayesian networks to provide a general-purpose compact representation for complex probability distributions.

In the last decade there has been a great deal of research focused on the problem of learning BNs from data (Buntine, 1996; Heckerman, 1998). An obvious motivation for this task is to learn a model that we can then use for inference or decision making, as a substitute for a model constructed by a human expert. In other circumstances, our goal might be to learn a model of the system not for prediction, but for discovering the domain structure. For example, we might want to use BN learning to understand the mechanism by which genes in a cell express themselves in protein, and the causal and dependence relations between the expression levels of different genes (Friedman et al., 2000; Lander, 1999). If we learn the "true" BN structure of our distribution, we reveal many important aspects about our domain. For example, if X and Y are not connected directly by an edge, then any correlation between them is an indirect one: there is some set of variables $\mathbf{Z}$ such that the influence of X on Y is mediated via $\mathbf{Z}$. More controversially, the presence of a directed path from X to Y indicates (under certain assumptions (Spirtes et al., 1993; Heckerman et al., 1997)) that *X causes Y*. The extraction of such *structural features* is often our primary goal in the discovery task, as can be seen by the emphasis in data mining research on discovering *association rules*. In fact, we can view the task of learning the structure of the underlying BN as providing a semantically coherent and well-defined goal for the discovery task.

The most common approach to discovering BN structure is to use learning with model selection to provide us with a single high-scoring model. We then use that model (or its *Markov equivalence class*) as our model for the structure of the domain. Indeed, in small domains with a substantial amount of data, it has been shown that the highest scoring model is orders of magnitude more likely than any other (Heckerman et al., 1997). In such cases, the use of model selection is a good approximation. Unfortunately, there are many domains of interest where this situation does not hold. In our gene expression example, we might have thousands of genes (each of which is modeled as a random variable) and only a few hundred experiments (data cases). In cases like this, where the amount of data is small relative to the size of the model, there are likely to be many models that explain the data reasonably well. Model selection makes a somewhat arbitrary choice between these models. However, structural features (e.g., edges) that appear in this single structure does not necessarily appear in other likely structures; indeed, we have no guarantees that these structural features are even likely relative to the set of possible structures. Furthermore, model selection is sensitive to the particular instances that it was given. Had we sampled another data set of the same size (from the same distribution), model selection would have learned a very different model. For both of these reasons, we cannot simply accept our chosen structure as a true representation of the underlying process.

Given that there are many qualitatively different structures that are approximately equally good, we cannot learn a unique structure from the data. Moreover, in many learning scenarios there are *exponentially* many structures that are "reasonably" good given the data. Thus, enumerating these structures is also impractical. However, there might be certain features of the distribution that are so strong that we can extract them reliably. As an extreme example, if two variables are highly correlated (e.g., deterministically related to each other), it is likely that an edge between them will appear in any high-scoring model. As we discussed above, extracting these structural features is often the primary goal of BN learning.

Bayesian learning allows us to estimate the strength with which the data indicates the presence of a certain feature. The *Bayesian score* of a model is simply its posterior probability given the data. Thus, we can estimate the extent to which a feature, e.g., the presence of an edge, is likely given the data by estimating its probability:

$$P(f \mid D) = \sum_G P(G \mid D) f(G), \tag{1}$$

where G represents a model, and $f(G)$ is 1 if the feature holds in G and 0 otherwise. If this probability is close to 1, then almost any high-scoring model contains the feature . On the other hand, if the probability is low, we know that the feature is absent in the most likely models.

The number of BN structures is super-exponential in the number of random variables in the domain; therefore, this summation can be computed in closed form only for very small domains, or those in which we have additional constraints that restrict the space (as in (Heckerman et al., 1997)). Alternatively, this summation can be approximated by considering only a subset of possible structures. Several approximations have been proposed (Madigan and Raftery, 1994; Madigan and York, 1995). One theoretically well-founded approach is to use Markov Chain Monte Carlo (MCMC) methods: we define a Markov chain over structures whose stationary distribution is the posterior $P(G \mid D)$, we then generate samples from this chain, and use them to estimate Eq. (1). This approach is quite popular, and variants have been used by Madigan and York (1995), Madigan et al. (1996), Giudici and Green (1999), and Giudici et al. (2000).

In this paper, we propose a new approach for evaluating the Bayesian posterior probability of certain structural network properties. The key idea in our approach is the use of an ordering on the network variables to separate the problem into two easier one. An *order* $\prec$ is a total ordering on the variables in our domain, which places a restriction on the structure of the learned BN: if $X \prec Y$, we restrict attention to networks where an edge between X and Y, if any, must go from X to Y. We can now decouple the problem of evaluating the probability over all structures into two subproblems: evaluating the probability for a given order, and summing over the set of possible orders. Our two main technical ideas provide solutions to these two subproblems.

In Section 3, we provide an efficient closed form equation for summing over all (super-exponentially many) networks with at most k parents per node (for some constant k) that are consistent with a fixed order $\prec$. This equation allows us both to compute the overall probability of the data for this set of networks, and to compute the posterior probability of certain structural features over this set. In Section 4, we show how to estimate the probability of a feature over the set of all orders by using an MCMC algorithm to sample among the possible orders. The space of orders is much smaller than the space of network structures; it also appears to be much less peaked, allowing much faster *mixing* (i.e., convergence to the stationary distribution of the Markov chain). We present empirical results illustrating this observation, showing that our approach has substantial advantages over direct MCMC over BN structures. The Markov chain over orders mixes much faster and more reliably than the chain over network structures. Indeed, different runs of MCMC over networks typically lead to very different estimates in the posterior probabilities of structural features, illustrating poor convergence to the stationary distribution; by contrast, different runs of MCMC over orders converge reliably to the same estimates. We also present results showing that our approach accurately detects dominant features even with sparse data, and that it outperforms both MCMC over structures and the non-Bayesian bootstrap approach of Friedman et al. (1999).

2. Bayesian learning of Bayesian networks

2.1. The Bayesian learning framework

Consider the problem of analyzing the distribution over some set of random variables $X_1, \ldots, X_n$, each of which takes values in some domain $Val(X_i)$. We are given a fully observed data set $D = \{\mathbf{x}[1], \ldots, \mathbf{x}[M]\}$, where each $\mathbf{x}[m]$ is a complete assignment to the variables $X_1, \ldots, X_n$ in $Val(X_1, \ldots, X_n)$.

The Bayesian learning paradigm tells us that we must define a prior probability distribution $P(\mathcal{B})$ over the space of possible Bayesian networks $\mathcal{B}$. This prior is then updated using Bayesian conditioning to give a posterior distribution $P(\mathcal{B} \mid D)$ over this space.

For Bayesian networks, the description of a model $\mathcal{B}$ has two components: the structure G of the network, and the values of the numerical parameters θ_G

associated with it. The network structure G is a directed acyclic graph, whose nodes represent the variables in the domain, and whose edges represent direct probabilistic dependencies between them. The BN structure encodes a set of conditional independence assumptions: that each node X is conditionally independent of all of its nondescendants in G given its parents (in G) $\mathrm{Pa}_G(X_i)$. These independence assumptions, in turn, imply many other conditional independence statements, which can be extracted from the network using a simple graphical criterion called *d-separation* (Pearl, 1988). In particular, they imply that a variable X is conditionally independent of all other network variables given its *Markov blanket* — the set consisting of X's parents, its children, and the other parents of its children. Intuitively, the Markov blanket of X is the set of nodes that are, in some sense, directly correlated with X, at least in some circumstances.[1] We use the *family* of a node X to denote the set consisting of X and its parents.

The parameterization θ_G of the network varies. For example, in a discrete Bayesian network of structure G, the parameters θ_G typically define a multinomial distribution $\theta_{X_i|\mathbf{u}}$ for each variable X_i and each assignment of values $\mathbf{u}$ to $\mathrm{Pa}_G(X_i)$. If we consider Gaussian Bayesian networks over continuous domains, then $\theta_{X_i|\mathbf{u}}$ contains the coefficients for a linear combination of $\mathbf{u}$ and a variance parameter.

To define the prior $P(\mathcal{B})$, we need to define a discrete probability distribution over graph structures G, and for each possible graph G, to define a density measure over possible values of parameters θ_G.

The prior over structures is usually considered the less important of the two components. Unlike other parts of the posterior, it does not grow as the number of data cases grows. Hence, relatively little attention has been paid to the choice of structure prior, and a simple prior is often chosen largely for pragmatic reasons. The simplest and therefore most common choice is a uniform prior over structures (Heckerman, 1998). To provide a greater penalty to dense networks, one can define a prior using a probability β that each edge be present; then networks with m edges have prior probability proportional to $\beta^m(1-\beta)^{\binom{n}{2}-m}$ (Buntine, 1991). An alternative prior, and the one we use in our experiments, considers the number of options in determining the families of G. Intuitively, if we decide that a node X_i has k parents, then there are $\binom{n-1}{k}$ possible parents sets. If we assume that we choose uniformly from these, we get a prior:

$$P(G) \propto \prod_{i=1}^{n} \binom{n-1}{|\mathrm{Pa}_G(X_i)|}^{-1}. \tag{2}$$

[1] More formally, the Markov blanket of X is the set of nodes that are directly linked to X in the undirected *Markov network* which is a minimal I-map for the distribution represented by G.

Note that the negative logarithm of this prior corresponds to the *description length* of specifying the parent sets, assuming that the cardinality of these sets are known. Thus, we implicitly assume that cardinalities of parent sets are uniformly distributed.

A key property of all these priors is that they satisfy:

- **Structure modularity** The prior $P(G)$ can be written in the form

$$P(G) = \prod_i \rho_{X_i}(\mathrm{Pa}_G(X_i))$$

where $\rho_{X_i}(\mathrm{Pa}_G(X_i))$ is a distribution over the possible parent-sets of X_i.

That is, the prior decomposes into a product, with a term for each variable in our domain. In other words, the choices of the families for the different variables are independent a priori.

Next we consider the prior over parameters, $P(\theta_G \mid G)$. Here, the form of the prior varies depending on the type of parametric families we consider. In discrete networks, the standard assumption is a *Dirichlet* prior over $\theta_{X_i|\mathbf{u}}$ for each variable X_i and each instantiation $\mathbf{u}$ to its parents (Heckerman, 1998). In Gaussian networks, we might use a Wishart prior (Heckerman and Geiger, 1995). For our purpose, we need only require that the prior satisfies two basic assumptions, as presented by Heckerman et al. (1995):

- **Global parameter independence:** Let $\theta_{X_i|\mathrm{Pa}_G(X_i)}$ be the parameters specifying the behavior of the variable X_i given the various instantiations to its parents. Then we require that

$$P(\theta_G \mid G) = \prod_i P(\theta_{X_i|\mathrm{Pa}_G(X_i)} \mid G) \tag{3}$$

- **Parameter modularity:** Let G and G' be two graphs in which $\mathrm{Pa}_G(X_i) = \mathrm{Pa}_{G'}(X_i) = \mathbf{U}$ then

$$P(\theta_{X_i|\mathbf{U}} \mid G) = P(\theta_{X_i|\mathbf{U}} \mid G') \tag{4}$$

Once we define the prior, we can examine the form of the posterior probability. Using Bayes rule, we have that

$$P(G \mid D) \propto P(D \mid G)P(G).$$

The term $P(D \mid G)$ is the *marginal likelihood* of the data given G, and is defined as the integral of the likelihood function over all possible parameter values for G.

$$P(D \mid G) = \int P(D \mid G, \theta_G)P(\theta_G \mid G)d\theta_G$$

The term $P(D \mid G, \theta_G)$ is simply the probability of the data given a specific Bayesian network. When the data is *complete*, this term is simply a product of conditional probabilities.

Using the above assumptions, one can show (see (Heckerman et al., 1995)):

THEOREM 2.1.: *If D is complete and* $P(G)$ *satisfies* parameter independence *and* parameter modularity *, then*

$$P(D \mid G) = \prod_i \int \prod_m P(x_i[m] \mid \mathrm{pa}_G(X_i)[m], \theta_{X_i|\mathrm{Pa}_G(X_i)}) P(\theta_{X_i|\mathrm{Pa}_G(X_i)}) d\theta_{X_i|\mathrm{Pa}_G(X_i)}.$$

If the prior also satisfies structure modularity, we can also conclude that the posterior probability decomposes:

$$P(G \mid D) \propto P(D \mid G)P(G) = \prod_i score(X_i, \mathrm{Pa}_G(X_i) \mid D) \qquad (5)$$

where

$$score(X_i, \mathbf{U} \mid D) = \rho_{X_i}(\mathbf{U}) \int \prod_m P(x_i[m] \mid \mathbf{u}[m], \theta_{X_i|\mathbf{U}}) P(\theta_{X_i|\mathbf{U}}) d\theta_{X_i|\mathbf{U}}$$

For standard priors such as Dirichlet or Wishart, $score(X_i, \mathrm{Pa}_G(X_i))$ has a simple closed form solution that is easily computed from the prior and certain *sufficient statistics* over the data. (E.g., in the case of multinomials with a Dirichlet prior, the sufficient statistics are simply the counts of the different events $x_i, \mathbf{u}$ in the data.)

We note that the parameter prior can have a substantial impact on the posterior distribution over structures. For example, in Dirichlet priors, the greater the "strength" of the parameter prior (the equivalent sample size defined by the hyperparameters), the greater the bias towards the distribution induced by the hyperparameters, leading structures that resemble that distribution to have a higher score. Of course, as the amount of data in the training set grows, the impact of the prior shrinks, but the impact can be quite significant for small data sets. This issue is fundamental to the Bayesian approach, including the use of the Bayesian score in standard BN structure search, and is outside the scope of this paper.

2.2. Bayesian model averaging

Recall that our goal is to compute the posterior probability of some feature $f(G)$ over all possible graphs G. This is equal to:

$$P(f \mid D) \;=\; \sum_G f(G) P(G \mid D)$$

The problem, of course, is that the number of possible BN structures is super-exponential: $2^{\Theta(n^2)}$, where n is the number of variables.[2]

We can reduce this number by restricting attention to structures G where there is a bound k on the number of parents per node. This assumption, which we will make throughout this paper, is a fairly innocuous one. There are few applications in which very large families are called for, and there is rarely enough data to support robust parameter estimation for such families. From a more formal perspective, networks with very large families tend to have low score. Let $\mathcal{G}_k$ be the set of all graphs with indegree bounded by some constant k. Note that the number of structures in $\mathcal{G}_k$ is still super-exponential: $2^{\Theta(kn \log n)}$.[3]

Thus, exhaustive enumeration over the set of possible BN structures is feasible only for tiny domains (4–5 nodes). One solution, proposed by several researchers (Madigan and Raftery, 1994; Madigan and York, 1995; Heckerman et al., 1997), is to approximate this exhaustive enumeration by finding a set $\mathcal{G}$ of high scoring structures, and then estimating the relative mass of the structures in $\mathcal{G}$ that contains f:

$$P(f \mid D) \approx \frac{\sum_{G \in \mathcal{G}} P(G \mid D) f(G)}{\sum_{G \in \mathcal{G}} P(G \mid D)}. \tag{6}$$

This approach leaves open the question of how we construct $\mathcal{G}$. The simplest approach is to use model selection to pick a single high-scoring structure, and then use that as our approximation. If the amount of data is large relative to the size of the model, then the posterior will be sharply peaked around a single model, and this approximation is a reasonable one. However, as we discussed in the introduction, there are many interesting domains (e.g., our biological application) where the amount of data is small relative to the size of the model. In this case, there is usually a large number of high-scoring models, so using a single model as our set $\mathcal{G}$ is a very poor approximation.

A simple approach to finding a larger set is to record all the structures examined during the search, and return the high scoring ones. However, the

[2] Recall that the $\Theta(f(n))$ denotes both an asymptotic lower bound and an asymptotic upper bound (up to a constant factor). In this case, the number of BN structures is at least $2^{\binom{n}{2}}$, because we have at least this many subgraphs for any complete graph over the n nodes. We have at most $3^{\binom{n}{2}}$ structures, because for each possible pair of nodes X_i, X_j we have either no edge, an edge $X_i \rightarrow X_j$, or an edge $X_i \leftarrow X_j$. Hence, we we have that the number of possible structures is both $2^{\Omega(n^2)}$ and $2^{O(n^2)}$.

[3] For each node X_i, we have at most $\binom{n}{k}$ possible families, so that the number of possible networks is $\binom{n}{k}^n \leq n^{kn} = 2^{kn \log n}$, giving us the upper bound. For the lower bound, consider a fixed ordering on the number of nodes, and consider each of the nodes X_i that appear in the second half of the ordering. For each of these, we have $\binom{n/2}{k}$ possible families, which, for k constant, is $\Omega(n^k)$. Considering the choice of family only for these nodes, the number of possible structures is at least $\binom{n/2}{k}^{n/2}$, which is $2^{\Omega(kn \log n)}$.

set of structures found in this manner is quite sensitive to the search procedure we use. For example, if we use greedy hill-climbing, then the set of structures we will collect will all be quite similar. Such a restricted set of candidates also show up when we consider multiple restarts of greedy hill-climbing and beam-search. This is a serious problem since we run the risk of getting estimates of confidence that are based on a biased sample of structures.

Madigan and Raftery (1994) propose an alternative approach called *Occam's window*, which rejects models whose posterior probability is very low, as well as complex models whose posterior probability is not substantially better than a simpler model (one that contains a subset of the edges). These two principles allow them to prune the space of models considered, often to a number small enough to be exhaustively enumerated. Madigan and Raftery also provide a search procedure for finding these models.

An alternative approach, proposed by Madigan and York (1995), is based on the use of *Markov chain Monte Carlo (MCMC)* simulation. In this case, we define a Markov Chain over the space of possible structures, whose stationary distribution is the posterior distribution $P(G \mid D)$. We then generate a set of possible structures by doing a random walk in this Markov chain. Assuming that we continue this process until the chain converges to the stationary distribution, we can hope to get a set of structures that is representative of the posterior. Related approaches have also been adopted by other researchers. Giudici and Green (1999) and Giudici et al. (2000) propose an MCMC approach over *junction trees* — undirected graphical models that are *decomposable*, i.e., where graph is triangulated. Green (1995) and Giudici et al. (2000) also extend the MCMC methodology to cases where closed-form integration over parameters is infeasible, by defining a *reversible jump* Markov Chain that traverses the space of parameters as well as structure. Madigan et al. (1996) provide an approach for MCMC sampling over the space of *PDAGs* (Partially Directed Acyclic Graphs), representing equivalence classes over network structures.

These MCMC solutions are the only approach that can, in principle, approximate true Bayesian model averaging by sampling from the posterior over network structures. They have been demonstrated with success on a variety of small domains, typically with 4–14 variables. However, there are several issues that potentially limit its effectiveness for large domains involving many variables. As we discussed, the space of network structures grows super-exponentially with the number of variables. Therefore, the domain of the MCMC traversal is enormous for all but the tiniest domains.[4] More importantly, the posterior distribution over structures is often quite peaked, with neighboring structures having very different scores. The reason is that even

[4] For the experiments done so far, the larger domains (those with more than 7–8 variables) were typically associated with a large set of structural constraints limiting the set of possible structures.

small perturbations to the structure — a removal of a single edge — can cause a huge reduction in score. Thus, the "posterior landscape" can be quite jagged, with high "peaks" separated by low "valleys". In such situations, MCMC is known to be slow to mix, requiring many samples to reach the posterior distribution. In Section 5 we provide experimental evidence indicating that these difficulties do, indeed, arise in practice.

3. Closed form for known order

In this section, we temporarily turn our attention to a somewhat easier problem. Rather than perform model averaging over the space of *all* structures, we restrict attention to structures that are consistent with some known total order $\prec$. In other words, we restrict attention to structures G where if $X_i \in \mathrm{Pa}_G(X_j)$ then $i \prec j$. This assumption was a standard one in the early work on learning Bayesian networks from data (Cooper and Herskovits, 1992).

3.1. Computing the marginal likelihood

We first consider the problem of computing the probability of the data given the order:

$$P(D \mid \prec) = \sum_{G \in \mathcal{G}_k} P(G \mid \prec) P(D \mid G) \tag{7}$$

Note that this summation, although restricted to networks with bounded in-degree and consistent with $\prec$, is still exponentially large: the number of such structures is still $2^{\Theta(kn \log n)}$.[5]

The key insight is that, when we restrict attention to structures consistent with a given order $\prec$, the choice of family for one node places no additional constraints on the choice of family for another. Note that this property does not hold without the restriction on the order; for example, if we pick X_i to be a parent of X_j, then X_j cannot in turn be a parent of X_i.

Therefore, we can choose a structure G consistent with $\prec$ by choosing, independently, a family $\mathbf{U}$ for each node X_i. The parameter modularity assumption in Eq. (4) states that the choice of parameters for the family of X_i is independent of the choice of family for another family in the network. Hence, summing over possible graphs consistent with $\prec$ is equivalent to summing over possible choices of family for each node, each with its parameter prior.

[5] Our lower bound in footnote 3 was derived in the case of a fixed ordering, and the matching upper bound certainly continues to hold in the more restricted case. Clearly, the number of structures in this case is substantially lower, but that difference expresses only in the different constant factor in the exponent, which is obscured by the Θ notation. (Note that a constant factor in the exponent corresponds to a different base for the exponent, a very significant difference.)

Given our constraint on the size of the family, the possible parent sets for the node X_i is

$$\mathcal{U}_{i,\prec} = \{\mathbf{U} \,:\, \mathbf{U} \prec X_i, |\mathbf{U}| \leq k\}.$$

where $\mathbf{U} \prec X_i$ is defined to hold when all nodes in $\mathbf{U}$ precede X_i in $\prec$. Let $\mathcal{G}_{k,\prec}$ be the set of structures in $\mathcal{G}_k$ consistent with $\prec$. Using Eq. (5), we have that

$$\begin{aligned} P(D \mid \prec) &= \sum_{G \in \mathcal{G}_{k,\prec}} \prod_i score(X_i, \mathrm{Pa}_G(X_i) \mid D) \\ &= \prod_i \sum_{\mathbf{U} \in \mathcal{U}_{i,\prec}} score(X_i, \mathbf{U} \mid D) \ . \end{aligned} \tag{8}$$

Intuitively, the equality states that we can sum over all networks consistent with $\prec$ by summing over the set of possible families for each node, and then multiplying the results for the different nodes. This transformation allows us to compute $P(D \mid \prec)$ very efficiently. The expression on the right-hand side consists of a product with a term for each node X_i, each of which is a summation over all possible families for X_i. Given the bound k over the number of parents, the number of possible families for a node X_i is at most $\binom{n}{k} \leq n^k$. Hence, the total cost of computing Eq. (8) is at most $n \cdot n^k = n^{k+1}$.

We note that the decomposition of Eq. (8) was first mentioned by Buntine (1991), but the ramifications for Bayesian model averaging were not pursued. The concept of Bayesian model averaging using a closed-form summation over an exponentially large set of structures was proposed (in a different setting) by Pereira and Singer (1999).

The computation of $P(D \mid \prec)$ is useful in and of itself; as we show in the next section, computing the probability $P(D \mid \prec)$ is a key step in our MCMC algorithm.

3.2. Probabilities of features

For certain types of features f, we can use the technique of the previous section to compute, in closed form, the probability $P(f \mid \prec, D)$ that f holds in a structure given the order and the data.

In general, if $f(\cdot)$ is a feature. We want to compute

$$P(f \mid \prec, D) = \frac{P(f, D \mid \prec)}{P(D \mid \prec)}.$$

We have just shown how to compute the denominator. The numerator is a sum over all structures that contain the feature and are consistent with the order:

$$P(f, D \mid \prec) = \sum_{G \in \mathcal{G}_{k,\prec}} f(G) P(G \mid \prec) P(D \mid G) \tag{9}$$

The computation of this term depends on the specific type of feature f.

The simplest situation is when we want to compute the posterior probability of a particular choice of parents $\mathbf{U}$. This in effect require us to sum over all graphs where $\mathrm{Pa}_G(X_i) = \mathbf{U}$. In this case, we can apply the same closed form analysis to (9). The only difference is that we restrict $\mathcal{U}_{j,\prec}$ to be the singleton $\{\mathbf{U}\}$. Since the terms that sum over the parents of X_k for $k \neq j$ are not disturbed by this constraint, they cancel out from the equation.

PROPOSITION 3.1.**:**

$$P(\mathrm{Pa}_G(X_i) = \mathbf{U} \mid D, \prec) = \frac{\mathrm{score}(X_i, \mathbf{U} \mid D)}{\sum_{\mathbf{U}' \in \mathcal{U}_{i,\prec}} \mathrm{score}(X_i, \mathbf{U}' \mid D)}. \tag{10}$$

A slightly more complex situation is when we want to compute the posterior probability of the *edge feature* $X_i \to X_j$. Again, we can apply the same closed form analysis to (9). The only difference is that we restrict $\mathcal{U}_{j,\prec}$ to consist only of subsets that contain X_i.

PROPOSITION 3.2.**:**

$$P(X_j \in \mathrm{Pa}_G(X_i) \mid \prec, D) = \frac{\sum_{\{\mathbf{U} \in \mathcal{U}_{i,\prec} \,:\, X_j \in \mathbf{U}\}} \mathrm{score}(X_i, \mathbf{U} \mid D)}{\sum_{\mathbf{U} \in \mathcal{U}_{i,\prec}} \mathrm{score}(X_i, \mathbf{U} \mid D)}$$

A somewhat more subtle computation is required to compute the posterior of the *Markov feature* $X_i \stackrel{M}{\sim} X_j$, denoting that X_i is in the Markov blanket of X_j; this feature holds if G contains the edge $X_i \to X_j$, or the edge $X_j \to X_i$, or there is a variable X_k such that both edges $X_i \to X_k$ and $X_j \to X_k$ are in G.

Assume, without loss of generality, that X_i precedes X_j in the order. In this case, X_i can be in X_j's Markov blanket either if there is an edge from X_i to X_j, or if X_i and X_j are both parents of some third node X_l. We have just shown how the first of these probabilities $P(X_j \in \mathrm{Pa}_G(X_i) \mid D, \prec)$, can be computed in closed form. We can also easily compute the probability $P(X_i, X_j \in \mathrm{Pa}_G(X_l) \mid D, \prec)$ that both X_i and X_j are parents of X_l: we simply restrict $\mathcal{U}_{l,\prec}$ to families that contain both X_i and X_j. The key is to note that as the choice of families of different nodes are independent, these are all independent events. Hence, X_i and X_j are not in the same Markov blanket only if all of these events fail to occur. Thus,

PROPOSITION 3.3.**:**

$$\begin{aligned} &P(X_i \stackrel{M}{\sim} X_j \mid D, \prec) = \\ &\quad 1 - (1 - P(X_j \in \mathrm{Pa}_G(X_i) \mid D, \prec)) \cdot \prod_{X_l \succ X_j} (1 - P(X_i, X_j \in \mathrm{Pa}_G(X_l) \mid D, \prec)) \end{aligned}$$

Unfortunately, this approach cannot be used to compute the probability of arbitrary structural features. For example, we cannot compute the probability that there exists some directed path from X_i to X_j, as we would have to consider all possible ways in which a path from X_i to X_j could manifest itself through our exponentially many structures.

We can overcome this difficulty using a simple sampling approach. Eq. (10) provides us with a closed form expression for the exact posterior probability of the different possible families of the node X_i. We can therefore easily sample entire networks from the posterior distribution given the order: we simply sample a family for each node, according to the distribution in Eq. (10). We can then use the sampled networks to evaluate any feature, such as the existence of a causal path from X_i to X_j.

4. MCMC methods

In the previous section, we made the simplifying assumption that we were given a predetermined order. Although this assumption might be reasonable in certain cases, it is clearly too restrictive in domains where we have very little prior knowledge (e.g., our biology domain). We therefore want to consider structures consistent with all $n!$ possible orders over BN nodes. Here, unfortunately, we have no elegant tricks that allow a closed form solution. Therefore, we provide a solution which uses our closed form solution of Eq. (8) as a subroutine in a Markov Chain Monte Carlo algorithm (Metropolis et al., 1953). This hybrid algorithm is a form of Rao-Blackwellized Monte Carlo sampling algorithm (Casella and Robert, 1996). Related approaches, called *mixture estimators* were proposed and analyzed by Gelfand and Smith (1990) and by Liu et al. (1994) (see discussion below). This approach is somewhat related to the work of Larrañaga et al. (1996), which proposes the use of a genetic algorithm to search for a high-scoring order; there, however, the score of an order is the score of a single high-scoring structure (as found by the K2 algorithm of Cooper and Herskovits (1992)), and the overall purpose is model selection rather than model averaging. Furthermore, genetic algorithms, unlike MCMC, are not guaranteed to generate samples from the posterior distribution.

4.1. THE BASIC ALGORITHM

We introduce a uniform prior over orders $\prec$, and define $P(G \mid \prec)$ to be of the same nature as the priors we used in the previous section. It is important to note that the resulting prior over structures has a different form than our original prior over structures. For example, if we define $P(G \mid \prec)$ to be uniform, we have that $P(G)$ is not uniform: graphs that are consistent with more orders are more likely. For example, a Naive Bayes graph is consistent with $(n-1)!$ orders, whereas any chain-structured graph is consistent with only one. As one consequence, our induced structure distribution is not *hypothesis* *equivalent* (Heckerman et al., 1995), in that different network structures that are in the same equivalence class often have different priors. For example, the chain $X \to Y \to Z$ is associated with a unique order, whereas the equivalent structure $X \leftarrow Y \to Z$ is associated with two orders, and is therefore twice as likely a priori. However, as Heckerman et al. observe, hypothesis equivalence is often too strong an assumption (e.g., in causal settings). They propose *likelihood equivalence* as a substitute, a property which clearly holds in our setting.

In general, while this discrepancy in priors is unfortunate, it is important to see it in proportion. The standard priors over network structures are often used not because they are particularly well-motivated, but rather because they are simple and easy to work with. In fact, the ubiquitous uniform prior over structures is far from uniform over PDAGs (Markov equivalence classes) — PDAGs consistent with more structures have a higher induced prior probability. One can argue that, for causal discovery, a uniform prior over PDAGs is more appropriate; nevertheless, a uniform prior over networks is most often used for practical reasons. Finally, the prior induced over our networks does have some justification: one can argue that a structure which is consistent with more orders makes fewer assumptions about causal order, and is therefore more likely a priori (Wallace et al., 1996).

We now construct a Markov chain $\mathcal{M}$, with state space O consisting of all $n!$ orders $\prec$; our construction will guarantee that $\mathcal{M}$ has the stationary distribution $P(\prec\mid D)$. We can then simulate this Markov chain, obtaining a sequence of samples $\prec_1, \ldots, \prec_T$. We can now approximate the expected value of any function $g(\prec)$ as:

$$\boldsymbol{E}[g \mid D] \approx \frac{1}{T} \sum_{t=1}^{T} g(\prec_t).$$

Specifically, we can let $g(\prec)$ be $P(f \mid \prec, D)$ for some feature (edge) f. We can then compute $g(\prec_t) = P(f \mid \prec_t, D)$, as described in the previous section.

It remains only to discuss the construction of the Markov chain. We use a standard Metropolis algorithm (Metropolis et al., 1953). We need to guarantee two things:

- that the chain is *reversible*, i.e., that $P(\prec \mapsto \prec') = P(\prec' \mapsto \prec)$;
- that the stationary distribution of the chain is the desired posterior distribution $P(\prec\mid D)$.

We accomplish this goal using a standard Metropolis sampling. For each order $\prec$, we define a *proposal probability* $q(\prec'|\prec)$, which defines the probability that the algorithm will "propose" a move from $\prec$ to $\prec'$. The algorithm then *accepts* this move with probability

$$\min\left[1, \frac{P(\prec' \mid D)q(\prec \mid \prec')}{P(\prec \mid D)q(\prec' \mid \prec)}\right].$$

It is well known that the resulting chain is reversible and has the desired stationary distribution (Gilks et al., 1996).

We consider several specific constructions for the proposal distribution, based on different neighborhoods in the space of orders. In one very simple construction, we consider only operators that flip two nodes in the order (leaving all others unchanged):

$$(i_1 \ldots i_j \ldots i_k \ldots i_n) \mapsto (i_1 \ldots i_k \ldots i_j \ldots i_n).$$

4.2. COMPUTATIONAL ISSUES

Although our closed form solution to the marginal likelihood and to the probabilities of the different structural features allows us to perform the computation in time polynomial in n, it can still be quite expensive, especially for large networks and reasonable size k. We utilize several ideas and approximations to reduce the complexity of these computations.

Our first set of ideas serve to reduce the scope of the summation both for the marginal likelihood and for the computation of feature probabilities. For each node X_i, we restrict attention to at most C other nodes as *candidate parents* (for some fixed C). We select these C nodes in advance, before any MCMC step, as follows: for each potential parent X_j, we compute the score of the single edge $X_j \to X_i$; we then select the C nodes X_j for which this score was highest. Note that C is different from k: C is the size of the set of nodes that could *potentially* be parents of a node X_i, whereas k is an upper bound on the size of the parent set actually chosen for X_i from among the set of C candidate parents.

Second, for each node X_i, we precompute the score for some number F of the highest-scoring families. The parents in these families are selected from among the C candidate parents for X_i. Again, this procedure is executed once, at the very beginning of the process. The list of highest-scoring families is sorted in decreasing order; let ℓ_i be the score of the worst family in X_i's list. As we consider a particular order, we extract from the list all families consistent with that order. We know that all families not in the list score no better than ℓ_i. Thus, if the best family extracted from the list is some factor γ better than ℓ_i, we choose to restrict attention to the families extracted from the list, under the assumption that other families will have negligible effect relative to these high-scoring families. If the score of the best family extracted is not that good, we do a full enumeration.

When performing exhaustive enumeration, we prune families that augment low-scoring families with low-scoring edges. Specifically, assume that for some family $\mathbf{U}$, we have that $score(X_i, \mathbf{U} \mid D)$ is substantially lower than other families enumerated so far. In this case, families that extend $\mathbf{U}$ are likely to be even worse. More precisely, we define the *incremental value* of a parent Y for X_i to be its added value as a single parent: $\Delta(Y;X_i) = score(X_i,Y) - score(X_i)$. If we now have a family $\mathbf{U}$ such that, for all other possible parents Y, $score(X_i,\mathbf{U}) + \Delta(Y;X_i)$ is lower than the best family found so far for X_i, we prune all extensions of $\mathbf{U}$.

In addition to reducing the scope of the summation, we can further reduce the cost of our MCMC algorithm, by observing that, when we take a single MCMC step in the space, we can often preserve much of our computation. In particular, let $\prec$ be an order and let $\prec'$ be the order obtained by flipping i_j and i_k. Now, consider the terms in Eq. (8); those terms corresponding to nodes i_ℓ in the order $\prec$ that precede i_j or succeed i_k do not change, as the set of potential parent sets $\mathcal{U}_{i_l,\prec}$ is the same. Furthermore, the terms for i_l that are between i_j and i_k also have a lot in common — all parent sets $\mathbf{U}$ that contain neither i_j nor i_k remain the same. Thus, we only need to subtract

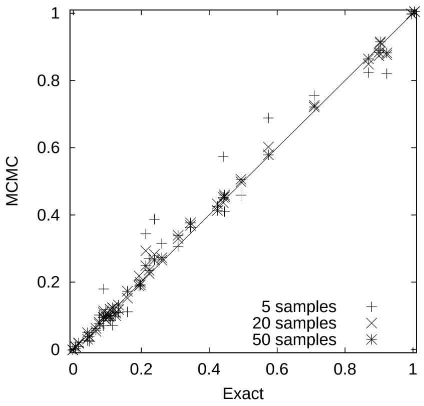
$$\sum_{\{\mathbf{U} \in \mathcal{U}_{i,\prec} \,:\, \mathbf{U} \ni X_{i_j}\}} score(X_i, \mathbf{U} \mid D)$$

and add

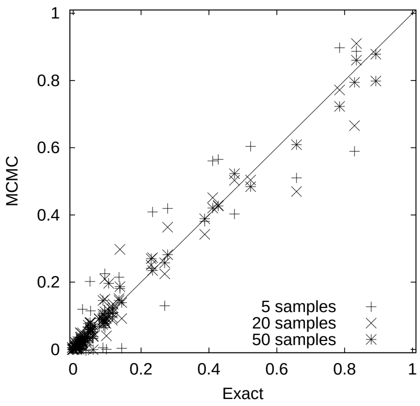
$$\sum_{\{\mathbf{U} \in \mathcal{U}_{i,\prec'} \,:\, \mathbf{U} \ni X_{i_k}\}} score(X_i, \mathbf{U} \mid D).$$

Having collected a set of order samples using our MCMC algorithm, we can use them to estimate the probability of the various structural features. However, this process can be quite expensive, especially when we are interested in the probabilities of all $\Theta(n^2)$ (edge or Markov) features. To reduce the computational burden, we perform this computation using only a small set of sampled orders. To make sure that we got a representative set of orders, we did not simply use the first orders generated by the MCMC process after a burn-in phase; rather, after the burn-in phase we continued running the MCMC process, collecting an order sample at fixed intervals (e.g., every 100 steps). This process results in samples from the chain that are closer to independent, thereby allowing us to provide a lower-variance estimate of the probability using a smaller number of samples.[6]

[6] An even better estimate would be obtained if we could use all of the samples generated by the MCMC process, but the computational cost of estimating feature probabilities for all of them would be prohibitive.



Markov



Edges

Figure 1. Comparison of posterior probabilities for the exact posterior over orders (x-axis) versus order-MCMC (y-axis) in the *Flare* dataset with 100 instances. The figures show the probabilities for all Markov features and edge features.

5. Experimental Results

We evaluated our approach in a variety of ways. We first compare it with a full Bayesian model averaging, in those domains small enough to permit an exhaustive enumeration of BN structures. Most importantly, we compare it with the more practical and most common approach to Bayesian model averaging: using MCMC directly over BN structures (Madigan and York, 1995). In this approach, a Metropolis-Hastings Markov chain is defined whose states correspond to individual BN structures. Each step in the chain corresponds to a local transformation on the structure: adding, deleting, or reversing an edge. The proposal distribution is uniform over these local transformations, and the acceptance probability is defined using the Bayesian score, in a way that guarantees that the stationary distribution of the chain is the posterior $P(G \mid D)$. We call our approach *order-MCMC* and the MCMC over BN structure *structure-MCMC*. Our primary measure for comparing the different approaches is via the probability that they give to the structural features we discuss above: edge features and Markov features.

Evaluating the Sampling Process. Our first goal is to evaluate the extent to which the sampling process reflects the result of true Bayesian model averaging. We first compared the estimates made by order-MCMC to estimates given by the full Bayesian averaging over networks. We experimented on the *Flare* dataset (Murphy and Aha, 1995), that has nine discrete variables, most of which take 2 or 3 values. We ran the MCMC sampler with a burn-in period of 1,000 steps and then proceeded to collect either 5, 20, or 50 order samples at fixed intervals of 100 steps. (We note that the burn-in time and sampling interval are probably excessive, but they ensure that we are sampling very close to the stationary probability of the process.) The results are shown in Figure 1. As we can see, the estimates are very robust. In fact, for Markov features even a sample of 5 orders gives a surprisingly decent estimate. This surprising success is due to the fact that a single sample of an order contains information about exponentially many possible structures. For edges we obviously need more samples, as edges that are not in the direction of the order necessarily have probability 0. With 20 and 50 samples we see a very close correlation between the MCMC estimate and the exact computation for both types of features.

Mixing rate. We then considered larger datasets, where exhaustive enumeration is not an option. For this purpose we used synthetic data generated from the *Alarm* BN (Beinlich et al., 1989), a network with 37 nodes. Here, our computational heuristics are necessary. We used the following settings: k (max. number of parents in a family) $= 3$;[7] C (max. number of potential parents) $= 20$; F (number of families cached) $= 4000$; and γ (difference in score required in pruning) $= 10$. Note that $\gamma = 10$ corresponds to a difference of 2^{10} in the posterior probability of the families. Different families have huge differences in score, so a difference of 2^{10} in the posterior probability is not uncommon.

Our first goal was the comparison of the mixing rate of the two MCMC samplers. For structure-MCMC, we used a burn in of 100,000 iterations and then sampled every 25,000 iterations. For order-MCMC, we used a burn in of 10,000 iterations and then sampled every 2,500 iterations. In both methods we collected a total of 50 samples per run. We note that, computationally, structure-MCMC is faster than order-MCMC. In our current implementation, generating a successor network is about an order of magnitude faster than generating a successor order. We therefore designed the runs in Figure 2 to take roughly the same amount of computation time.

In both approaches, we experimented with different initializations for the MCMC runs. In the uninformed initialization, we started the structure-MCMC with an empty network and the order-MCMC with a random order . In the informed initialization, we started the structure-MCMC with the *greedy network* — the BN found by greedy hill climbing search over network structures (Heckerman, 1998) and the order-MCMC with an order consistent with that structure.

One phenomenon that was quite clear was that order-MCMC runs *mix* much faster. That is, after a small number of iterations, these runs reached a "plateau" where successive samples had comparable scores. Runs started in

[7] We note that the maximum number of parents in a family in the original *Alarm* network is 3, hence our choice of $k = 3$.

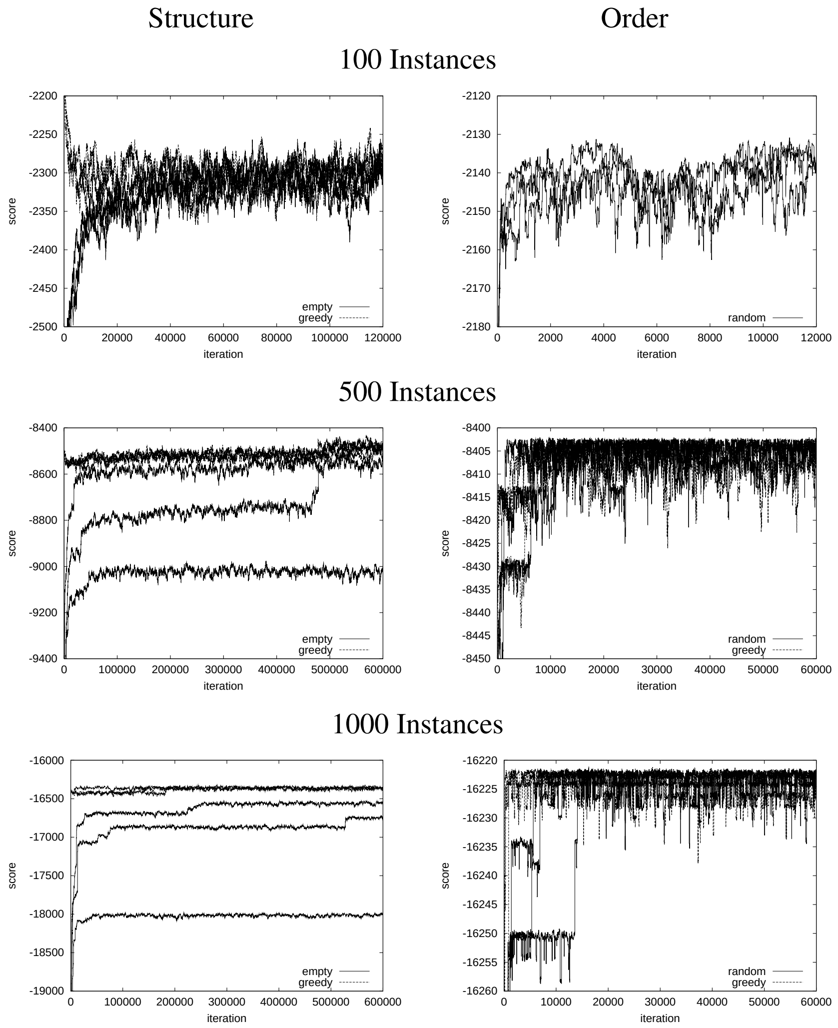


Figure 2. Plots of the progression of the MCMC runs. Each graph shows plots of 6 independent runs over *Alarm* with either 100, 500, and 1000 instances. The graph plot the score ($\log_2(P(D \mid G)P(G))$ or $\log_2(P(D \mid \prec)P(\prec))$) of the "current" candidate (y-axis) for different iterations (x-axis) of the MCMC sampler. In each plot, three of the runs are initialized with an uniformed network or order, and the others with the greedy network or an ordering consistent with it.

different places (including random order and orders seeded from the results of a greedy-search model selection) rapidly reached the same plateau. On the other hand, MCMC runs over network structures reached very different levels of scores, even though they were run for a much larger number of iterations. Figure 2 illustrates this phenomenon for examples of *Alarm* with 100, 500, and 1000 instances. Note the substantial difference in the scale of the *y*-axis between the two sets of graphs.

In the case of 100 instances, both MCMC samplers seemed to mix. Structure-MCMC mixes after about 20,000–30,000 iterations, while order-MCMC mixes after about 1,000–2,000 iterations. On the other hand, when we examine 500 samples, order-MCMC converges to a high-scoring plateau, which we believe is the stationary distribution, within 10,000 iterations. By contrast, different runs of the structure-MCMC stayed in very different regions of the in the first 500,000 iterations. The situation is even worse in the case of 1,000 instances. In this case, structure-MCMC started from an empty network does not reach the level of score achieved by the runs starting from the structure found by greedy hill climbing search. Moreover, these latter runs seem to fluctuate around the score of the initial seed, never exploring another region of the space. Note that different runs show differences of 100–500 bits. Thus, the sub-optimal runs sample from networks that are at least 2^{100} less probable!

Effects of Mixing. This phenomenon has two explanations. Either the seed structure is the global optimum and the sampler is sampling from the posterior distribution, which is “centered” around the optimum; or the sampler is stuck in a local “hill” in the space of structures from which it cannot escape. This latter hypothesis is supported by the fact that runs starting at other structures (e.g., the empty network) take a very long time to reach similar level of scores, indicating that there is a very different part of the space on which stationary behavior is reached. We now provide further support for this second hypothesis.

We first examine the posterior computed for different features in different runs. Figure 3 compares the posterior probability of Markov features assigned by different runs of structure-MCMC. Let us first consider the runs over 500 instances. Here, although different runs give a similar probability estimate to most structural features, there are several features on which they differ radically. In particular, there are features that are assigned probability close to 1 by structures sampled from one run and probability close to 0 by those sampled from the other. While this behavior is less common in the runs seeded with the greedy structure, it occurs even there. This phenomenon suggests that each of these runs (even runs that start at the same place) gets trapped in a different local neighborhood in the structure space. Somewhat surprisingly, a similar phenomenon appears to occur even in the case of 100 instances, where the runs appeared to mix. In this case, the overall correlation between the runs is, as we might expect, weaker: with 100 instances, there are many more high-scoring structures and therefore the variance of the sampling process is higher. However, we once again observe features which have probability close to 0 in one run and close to 1 in the other. These discrepancies are not

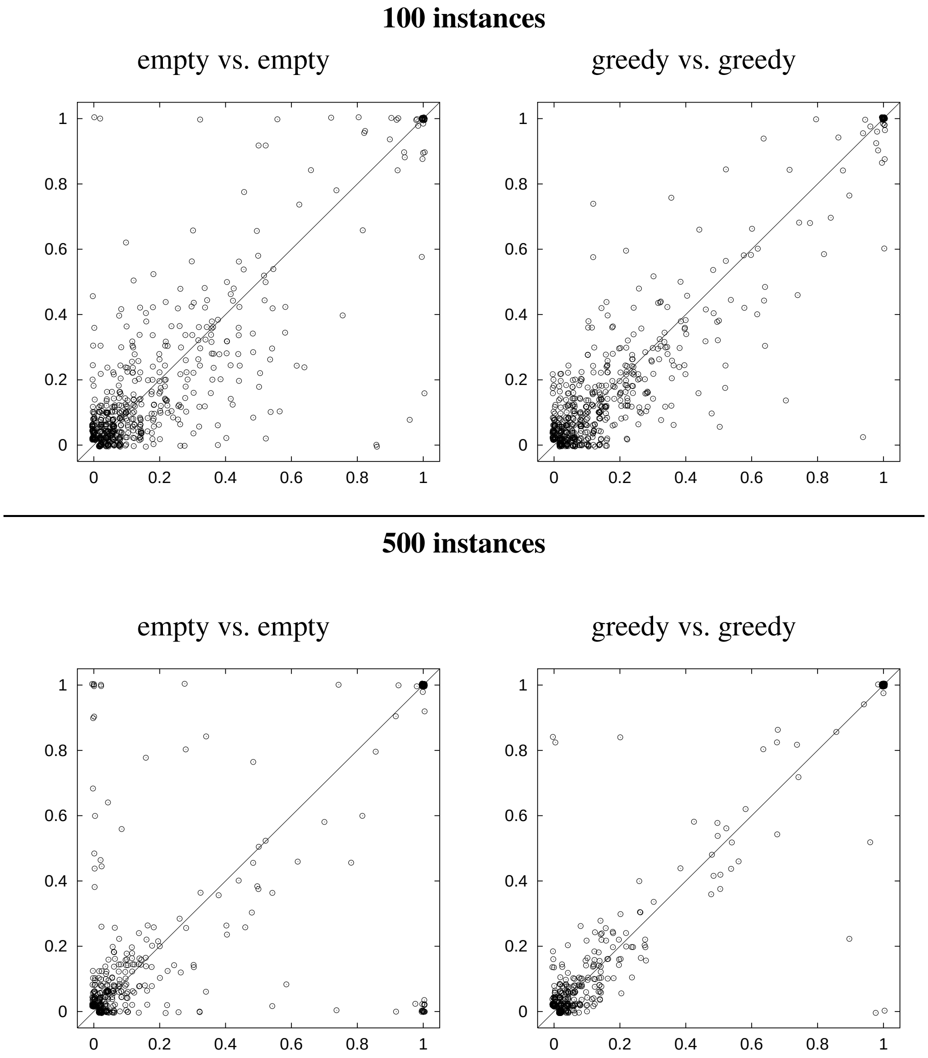


Figure 3. Scatter plots that compare posterior probability of Markov features on the *Alarm* dataset, as determined by different runs of structure-MCMC. Each point corresponds to a single Markov feature; its x and y coordinates denote the posterior estimated by the two compared runs. The position of points is slightly randomly perturbed to visualize clusters of points in the same position.

as easily explained by the variance of the sampling process. Therefore, even for 100 instances, it is not clear that structure-MCMC mixes.

By contrast, comparison of the predictions of different runs of order-MCMC are tightly correlated. To test this, we compared the posterior estimates of Markov features and Path features. The latter represent relations of the form “there is a directed path from X to Y” in the PDAG of the network structure.

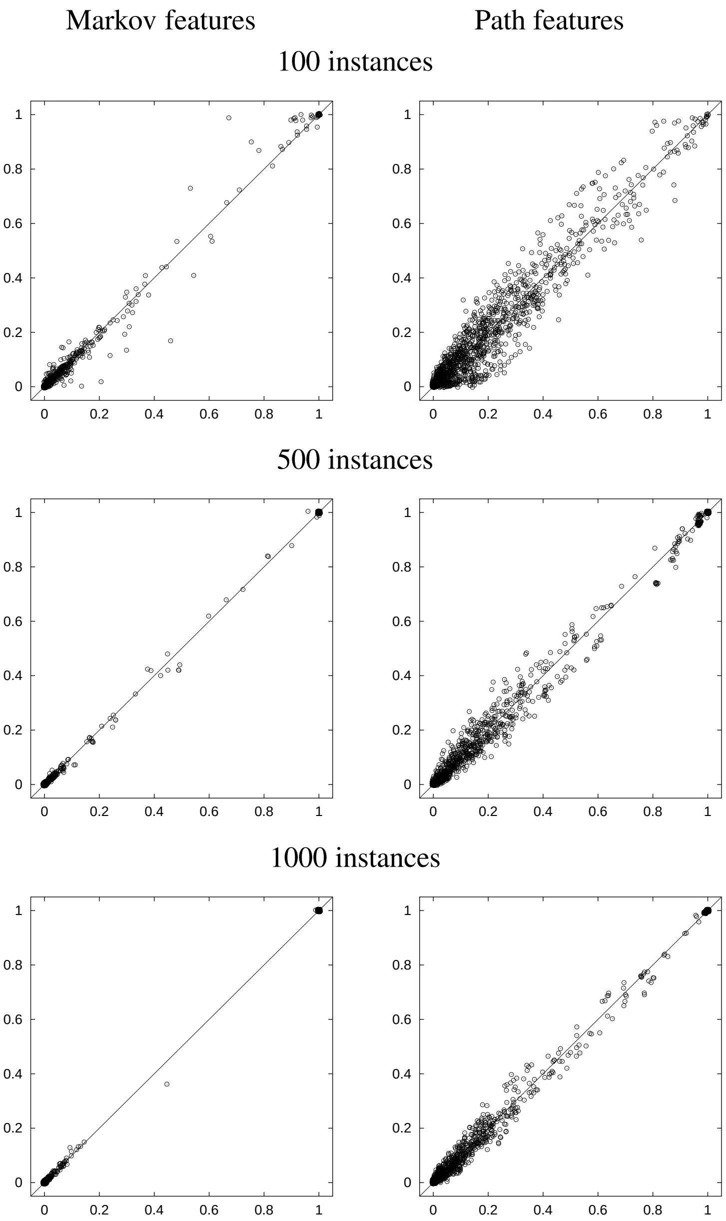


Figure 4. Scatter plots that compare posterior probability of Markov and Path features on the *Alarm* domain as determined by different runs of order-MCMC. Each point corresponds to a single feature; its x and y coordinates denote the posterior estimated by the greedy seeded run and a random seeded run respectively.

As discussed in Section 3, we cannot provide a closed form expression for the posterior of such a feature given an order. However, we can sample networks from the order, and estimate the feature relative to those. In our experiments, we sampled 20 networks from each order. Figure 4 compares two runs, one starting from an order consistent with the greedy structure and the other from a random order. We can see that the predictions are very similar, both for the small dataset and the larger one. The predictions for the Path features have somewhat higher variance, which we attribute to the additional randomness of sampling structures from the ordering. The very high degree of correlation between the two runs reaffirms our claim that they are indeed sampling from similar distributions. That is, they are sampling from the exact posterior.

We believe that the difference in mixing rate is due to the smoother posterior landscape of the space of orders. In the space of networks, even a small perturbation to a network can lead to a huge difference in score. By contrast, the score of an order is a lot less sensitive to slight perturbations. For one, the score of each order is an aggregate of the scores of a very large set of structures; hence, differences in scores of individual networks can often cancel out. Furthermore, for most orders, we are likely to find a consistent structure which is not too bad a fit to the data; hence, an order is unlikely to be uniformly horrible.

The disparity in mixing rates is more pronounced for larger datasets. The reason is quite clear: as the amount of data grows, the posterior landscape becomes "sharper" since the effect of a single change in the structure is amplified across many samples. As we discussed above, if our dataset is large enough, model selection is often a good approximation to model averaging. However, it is important to note that 500 instances for *Alarm* are not enough to peak the posterior sharply enough that model selection is a reliable approach to discovering structure. We can see that by examining the posterior probabilities in Figure 4. We see that the posterior probability for most Markov features is fairly far from 0 or 1. As Markov features are invariant for all networks in the same Markov equivalence class (PDAG), this phenomenon indicates that there are several PDAGs that have high score given the data. By contrast, in the case of 1000 instances, we see that the probability of almost all features is clustered around 0 or 1, indicating that model selection is likely to return a fairly representative structure in this case.

A second form of support for the non-mixing conjecture is obtained by considering an even smaller data set: the *Boston-housing* data set, from the UCI repository (Murphy and Aha, 1995), is a continuous domain with 14 variables and 506 samples. Here, we considered linear Gaussian networks, and used a standard Wishart parameter prior. We started the structure-MCMC on the structure obtained from greedy hill-climbing search. We started the order-MCMC on an order consistent with that structure. As usual, as shown in Figure 6(a), structure-MCMC does not converge. However, as shown in Figure 6(b), the runs of order-MCMC are also somewhat more erratic, indicating a more jagged posterior landscape even over orders. In a way, this is not

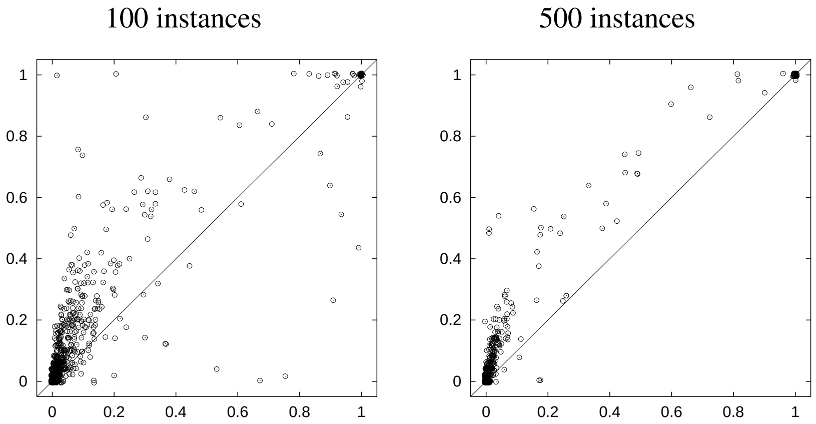
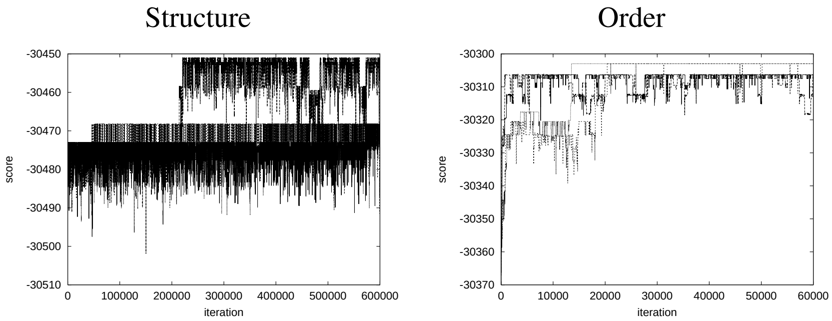


Figure 5. Scatter plots that compare posterior probability of Markov features on the *Alarm* domain as determined by the two different MCMC samplers. Each point corresponds to a single Markov feature; its x and y coordinates denote the posterior estimated by the greedy seeded run of order-MCMC and structure-MCMC, respectively.



(a) (b)

Figure 6. Plots of the progression of the MCMC runs on the *Boston-housing* data set. Each graph shows plots of 4 independent runs. All the runs are seeded with the network found by searching over network structures.

surprising, given the large number of instances and small domain. In Figure 7, we see that, as above, different runs of structure-MCMC lead to very different answers, whereas different runs of order-MCMC are very consistent.

More interesting is the examination of the feature probabilities themselves. Figure 8(a) shows a comparison between the feature probabilities of structure-MCMC and those of the structure returned by greedy search, used as the starting point for the chain. We can see that most of the structures traversed by the MCMC search are very similar to the greedy seed. By contrast, Figure 8(b) shows that order-MCMC traverses a different region of the space,

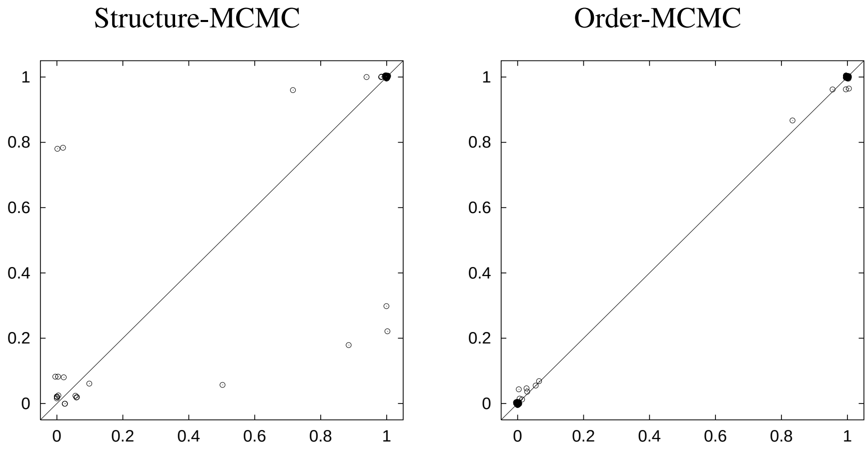
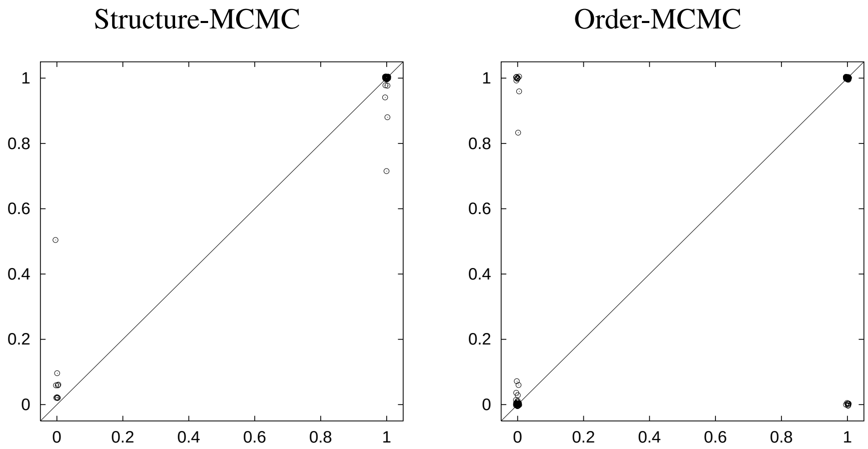


Figure 7. Scatter plots that compare posterior probability of Markov on the *Boston-housing* data set, as determined by different runs of structure-MCMC and order-MCMC.



(a) (b)

Figure 8. Scatter plots that compare posterior probability of Markov features on the *Boston-housing* data set, as determined by different runs of structure-MCMC and order-MCMC, to the probabilities according to the initial seed of the MCMC runs. The *x*-axis denotes whether the feature appears in the seed network: 1 if it appear and 0 if does not. The *y*-axis denote the estimate of the posterior probability of the feature based on the MCMC sampling.

leading to very different estimates. It turns out that the structure found by the greedy search is suboptimal, but that structure-MCMC remains stuck in a local maximum around that point. By contrast, the better mixing properties of order-MCMC allow is to break out of this local maximum, and to reach a substantially higher-scoring region. Thus, even in cases where there is a dominant global maximum, order-MCMC can be a more robust approach than greedy hill-climbing, structure-MCMC, or their combination.

Comparison of Estimates. We now compare the estimates of the two approaches on the *Alarm* data set. We deliberately chose to use the smaller data sets for two reasons: to allow structure-MCMC a better chance to mix, and to highlight the differences resulting from the different priors used in the two approaches. The results are shown in Figure 5. We see that, in general, the estimates of the two methods are not too far apart, although the posterior estimate of the structure-MCMC is usually larger.

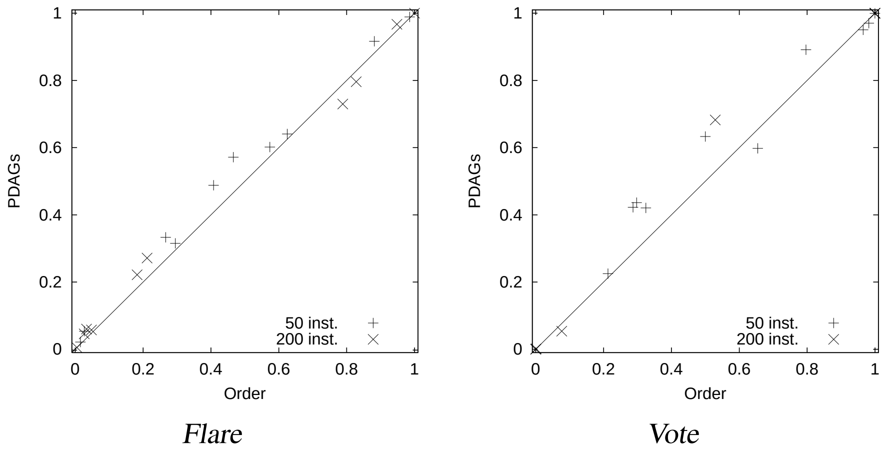


Figure 9. Comparison of posterior probabilities for different Markov features between full Bayesian averaging using: orders (x-axis) versus PDAGs (y-axis) for two UCI datasets (5 variables each).

We attribute these discrepancies in the posterior to the different structure prior we employ in the order-MCMC sampler. To test this conjecture, in a way that decouples it from the effects of sampling, we chose to compare the exact posterior computed by summing over *all* orders to the posterior computed by summing over all equivalence classes of Bayesian networks (PDAGs) (i.e., we counted only a single representative network for each equivalence class.) Of course, in order to do the exact Bayesian computation we need to do an exhaustive enumeration of hypotheses. For orders, this enumeration is possible for as many as 10 variables, but for structures, we are limited to domains with 5–6 variables. We took two data sets — *Vote* and *Flare* — from the UCI repository (Murphy and Aha, 1995) and selected five variables from each (all of which are discrete). We generated datasets of sizes 50 and 200, and computed the full Bayesian averaging posterior for these datasets using both methods. Figure 9 compares the results for both datasets. We see that the two approaches are well correlated, but that the prior does have some effect.

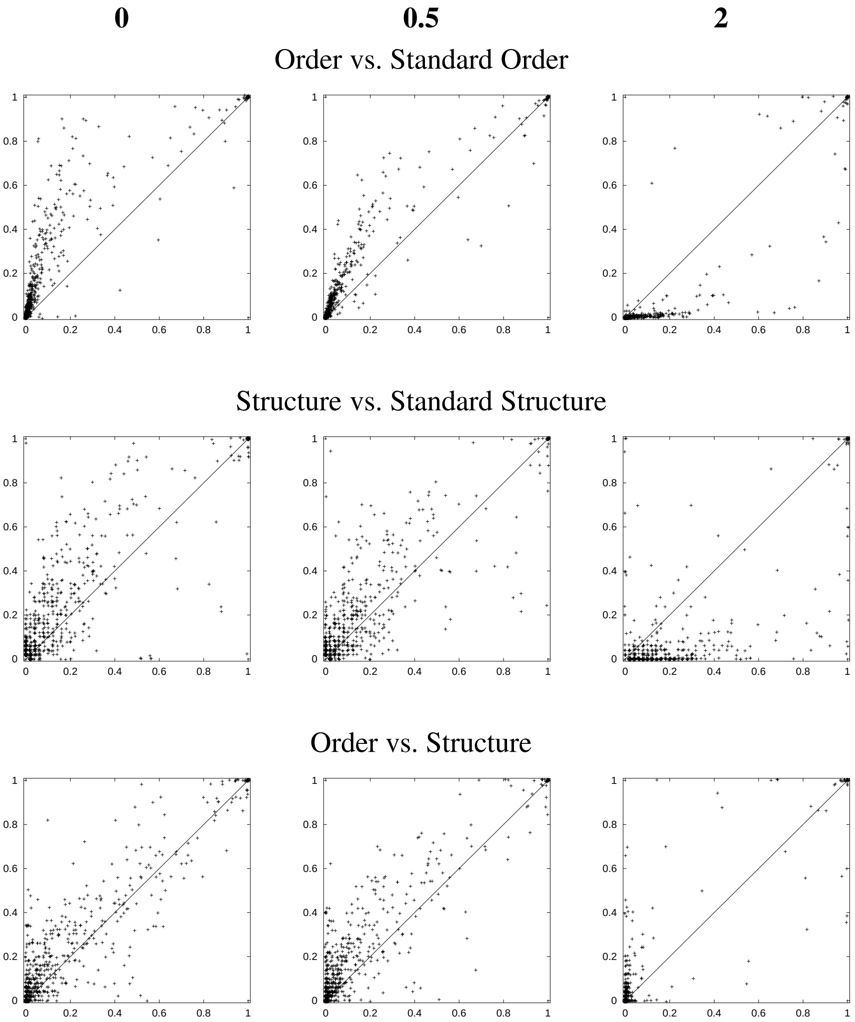


Figure 10. Comparison of the posterior of Markov features when we change the structure prior strength for *Alarm* with 100 instances. The top row compares the modified prior (y-axis) in order-MCMC against the standard prior (x-axis). The middle row makes an analogous comparison for structure-MCMC. The bottom compares the modified prior with order (x-axis) against the modified prior with structures (y-axis). Each column corresponds to a different weighting of the prior, as denoted at the top of the column.

To gain a better understanding of the general effect of a structure prior, we examined the sensitivity of Bayesian model averaging to changes in the prior. Recall that our experiments use the MDL prior shown in Eq. (2), whether for $P(G)$ (in structure-MCMC) or for $P(G \mid \prec)$ (in order-MCMC). We ran the same experiment, raising this prior to some power — 0, $\frac{1}{2}$, or 2. Note that a power of 0 corresponds to a uniform prior, over structures in the structure-

MCMC case and over structures within an order in the order-MCMC case. By contrast, a power of 2 corresponds to an even more extreme penalty for large families. Figure 10 shows the comparison of the modified priors to the "standard" case. As we can expect, a stronger structure prior results in lower posterior for features while a uniform structure prior is more prone to adding edges and thus most features have higher posterior. Thus, we see that the results of a structure discovery algorithm are always sensitive to the structure prior, and that even two very reasonable (and common) priors can lead to very different results. This effect is at least as large as the effect of using our order-based structure prior. Given that the choice of prior in BN learning is often somewhat arbitrary, there is no reason to assume that our order-based prior is less reasonable than any other.

Structure Reconstruction. This phenomenon raises an obvious question: given that the approaches give different results, which is better at reconstructing features of the generating model. To test this, we label Markov features in the *Alarm* domain as *positive* if they appear in the generating network and *negative* if they do not. We then use our posterior to try and distinguish "true" features from "false" ones: we pick a threshold t, and predict that the feature f is "true" if $P(f) > t$. Clearly, as we vary the the value of t, we will get different sets of features. At each threshold value we can have two types of errors: *false positives* — positive features that are misclassified as negative, and *false negatives* — negative features that are classified as positive. Different values of t achieve different tradeoffs between these two type of errors. Thus, for each method we can plot the *tradeoff curve* between the two types of errors. Note that, in most applications of structure discovery, we care more about false positives than about false negatives. For example, in our biological application, false negatives are only to be expected — it is unrealistic to expect that we would detect all causal connections based on our limited data. However, false positives correspond to hypothesizing important biological connections spuriously. Thus, our main concern is with the left-hand-side of the tradeoff curve, the part where we have a small number of false positives. Within that region, we want to achieve the smallest possible number of false negatives.

We computed such tradeoff curves for *Alarm* data set with 100, 500, and 1000 instances for two types of features: Markov features and Path features. Figure 11 displays ROC curves comparing order-MCMC, structure-MCMC, and the *non-parametric Bootstrap* approach of Friedman et al. (1999), a non-Bayesian simulation approach to estimate "confidence" in features. The curves represent the average performance over ten repetitions of the experiment — we sampled ten data sets from the *Alarm* data set, and for each each threshold t we report the average number of errors of both types.

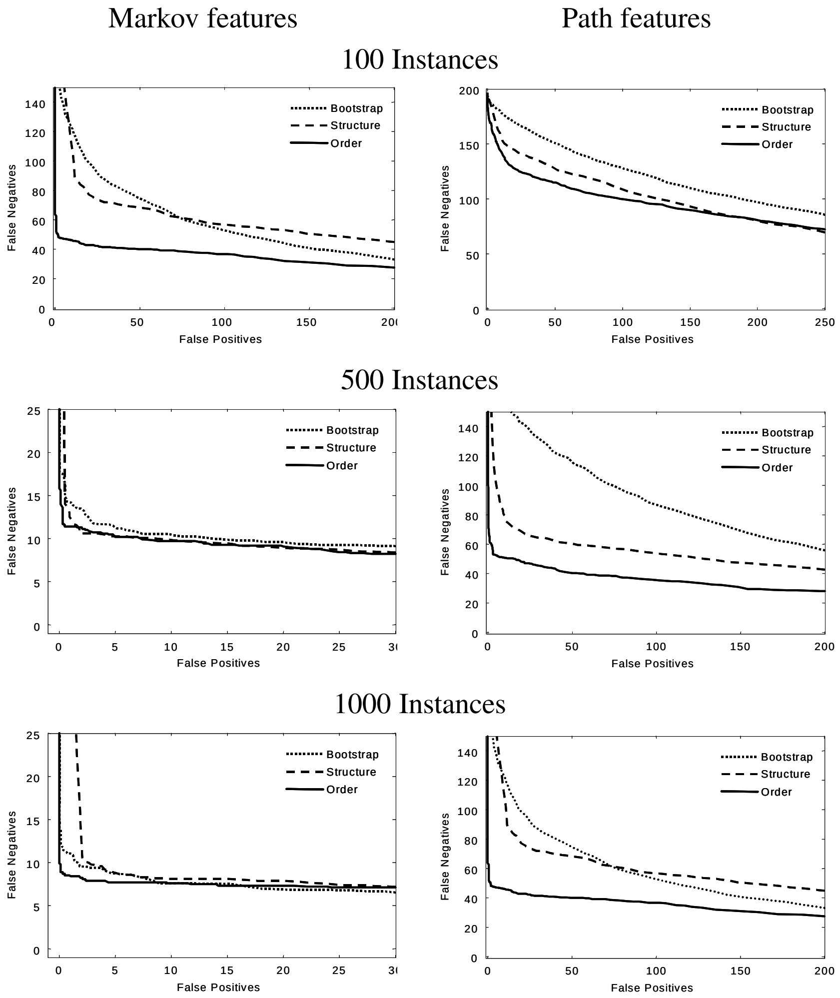


Figure 11. Classification tradeoff curves for different methods on datasets of varying sizes sampled from the *Alarm* network. The x-axis and the y-axis denote *false positive* and *false negative* errors, respectively. The curve is achieved by different threshold values in the range $[0, 1]$. Each graph contains three curves, each collected over 50 samples: order-MCMC, with order samples collected every 200 iterations; structure-MCMC, with structure samples collected every 1000 iterations; and 50 network structures generated by the non-parametric bootstrap sampling method.

As we can see, in all cases order-MCMC does as well or better than the other approaches, with marked gains in three cases. In particular, for t larger than 0.4, order-MCMC makes no false positive errors for Markov features on the 1000-instance data set. We believe that features it misses are due to weak interactions in the network that cannot be reliably learned from such a small data set.

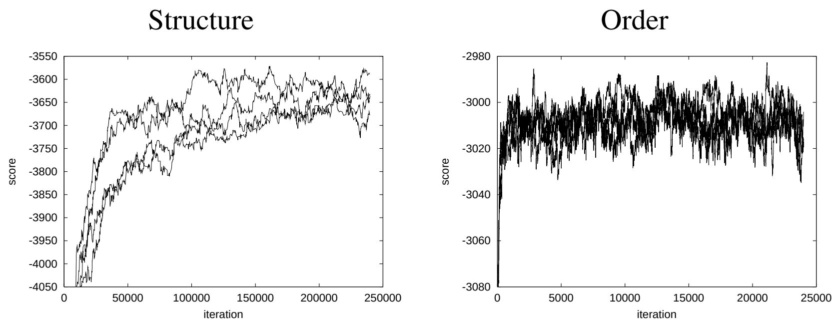


Figure 12. Plots of the progression of the MCMC runs on the *Genetics* data set. Each graph shows plots of 4 independent runs. All the runs are seeded with the network found by searching over network structures.

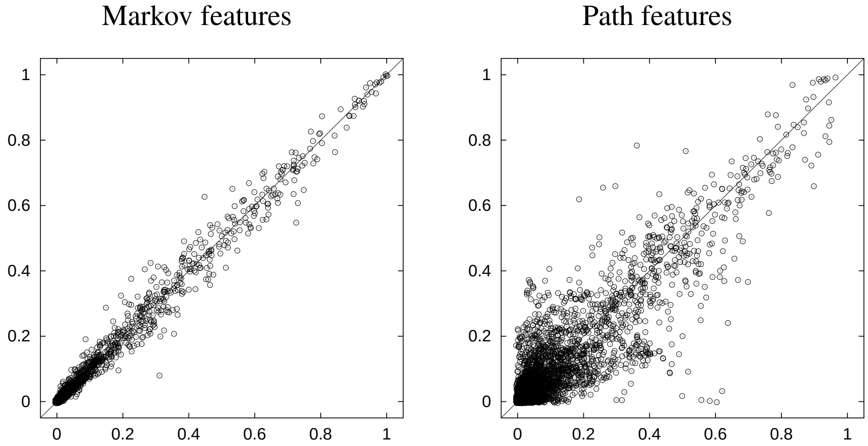


Figure 13. Scatter plots that compare posterior probability of Markov and path features on the *Genetics* data set, as determined by different runs of order-MCMC.

Application to Gene Expression Data. As stated in the introduction our goal is to apply structure estimation methods for causal learning from gene expression data. We tested our method on a relatively small genetic data set of Friedman et al. (2000). This data set is derived from a larger data set of *S. cerevisiae* cell-cycle measurements reported in Spellman et al. (1998). The data set contains 76 samples of 250 genes. Friedman et al. discretized each measurement into three values ("under-expressed", "normal", "over-expressed").

We applied order-MCMC, using an informed greedy initialization. For these runs, we used : k (max. number of parents in a family) $= 3$; C (max. number of potential parents) $= 45$; F (number of families cached) $= 4000$; and γ (difference in score required in pruning) $= 10$. (The choice of $k = 3$ is imposed by computational limitations, induced by the large number of variables

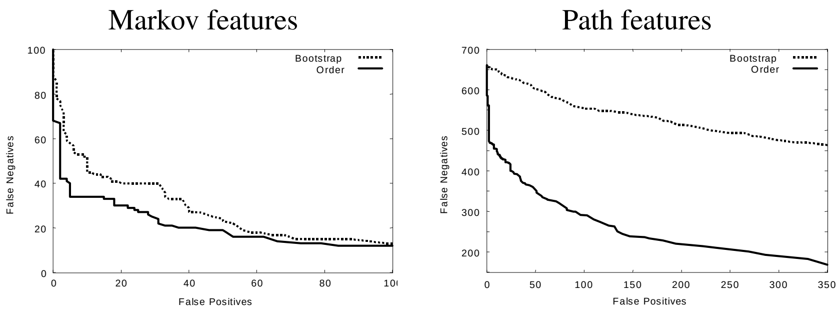


Figure 14. Classification tradeoff curves for different methods on the simulated *Genetics* data set. The *x*-axis and the *y*-axis denote *false positive* and *false negative* errors, respectively. The curve is achieved by different threshold values in the range $[0,1]$. .

in the domain.) We used a burn-in period of 4000 iterations, and then sampled every 400 iterations collecting 50 samples in each run.

Figure 12 shows the progression of runs of the two MCMC methods on this data. As we can see, order-MCMC mixes rapidly (after a few hundred iterations). On the other hand, structure-MCMC seems to be mixing only after 200,000 iterations. Figure 13 shows comparison of estimates from two different runs of the order based MCMC sampler. As in the other data sets, the estimates for Markov features based on the two different runs are very similar. In this case, we also show the estimates for path features, which are obtained (as discussed in Section 3.2) by sampling specific networks from the distribution over networks for a given order, and then evaluating the presence or absence of a path in each sampled networks. In this case, we sampled 500 networks from our 50 sampled orderings. The variance of this estimator is, as can be expected, much higher; nevertheless, the estimates are still quite well-correlated.

Since we want to use this tool for scientific discovery, we want to evaluate how well Bayesian structure estimation performs in this domain. To do so we performed the following simulation experiments. We sampled 100 instances from the network found by structure search on the genetics data. We then applied the order based MCMC sampler and the bootstrap approach and evaluated the success in reconstructing features of the generating network. Figure 14 shows the tradeoff between the two types of errors for these two methods in predicting Markov and path features. As we can see, order-MCMC clearly outperforms the bootstrap approach.

We should stress that the simulation is based on a network that is probably simpler than the underlying structure (since we learned it from few samples). Nonetheless, we view these results as an indication that using Bayesian estimates is more reliable in this domain. A discussion of the biological conclusions from this analysis are beyond the scope of this paper.

6. Discussion and future work

We have presented a new approach for expressing the posterior distribution over BN structures given a data set, and thereby for evaluating the posterior probability of important structural features of the distribution. Our approach is based on two main ideas. The first is a clean and computationally tractable expression for the posterior of the data given a known order over network variables. The second is Monte Carlo sampling algorithm over orders. We have shown experimentally that this approach mixes substantially faster than the standard MCMC algorithm that samples structures directly.

Once we have generated a set of orders sampled from the posterior distribution, we can use them in a variety of ways. As we have shown, we can estimate the probabilities of certain structural features — edge features or adjacency in Markov neighborhoods — directly in closed form for a given order. For other structural features, we can estimate their probability by sampling network structures from each order, and testing for the presence or absence of the feature in each structure.

We have shown that the estimates returned by our algorithm, using either of these two methods, are substantially more robust than those obtained from standard MCMC over structures. To some extent, if we ignore the different prior used in these two approaches, this phenomenon is due to the fact that mixture estimators have lower variance than estimators based on individual samples (Gelfand and Smith, 1990; Liu et al., 1994). More significantly, however, we see that the results of MCMC over structures are substantially less reliable, as they are highly sensitive to the region of the space to which the Markov chain process happens to gravitate.

We have also tested the efficacy of our algorithm for the task of recovering structural features which we know are present. We have shown that our algorithm is always more reliable at recovering features than MCMC over structures, and in all but one case also more reliable than the bootstrap approach of Friedman et al. (1999).

We believe that this approach can be extended to deal with data sets where some of the data is missing, by extending the MCMC over orders with MCMC over missing values, allowing us to average over both. If successful, we can use this combined MCMC algorithm for doing full Bayesian model averaging for prediction tasks as well. Finally, we plan to apply this algorithm in our biology domain, in order to try and understand the underlying structure of gene expression.

Acknowledgements

The authors thank Yoram Singer for useful discussions and Harald Steck, Nando de Freitas, and the anonymous reviewers for helpful comments and references. This work was supported by ARO grant DAAH04-96-1-0341 under the MURI program "Integrated Approach to Intelligent Systems", by DARPA's *Information Assurance* program under subcontract to SRI International, and by Israel Science Foundation (ISF) grant 244/99. Nir Friedman was also supported through the generosity of the Michael Sacher Trust Alon Fellowship, and Sherman Senior Lectureship. Daphne Koller was also supported through the generosity of the Sloan Foundation and the Powell Foundation. The experiments reported here were performed on computers funded by an ISF infrastructure grant.

References

Beinlich, I., G. Suermondt, R. Chavez, and G. Cooper: 1989, 'The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks'. In: *Proc. 2'nd European Conf. on AI and Medicine*. Berlin: Springer-Verlag.

Buntine, W. L.: 1991, 'Theory refinement on Bayesian networks'. In: B. D. D'Ambrosio, P. Smets, and P. P. Bonissone (eds.): *Proc. Seventh Annual Conference on Uncertainty Artificial Intelligence (UAI '91)*. San Francisco: Morgan Kaufmann, pp. 52–60.

Buntine, W. L.: 1996, 'A guide to the literature on learning probabilistic networks from data'. *IEEE Transactions on Knowledge and Data Engineering* **8**, 195–210.

Casella, G. and C. Robert: 1996, 'Rao-Blackwellisation of sampling schemes'. *Biometrika* **83**(1), 81–94.

Cooper, G. F. and E. Herskovits: 1992, 'A Bayesian method for the induction of probabilistic networks from data'. *Machine Learning* **9**, 309–347.

Friedman, N., M. Goldszmidt, and A. Wyner: 1999, 'Data Analysis with Bayesian Networks: A Bootstrap Approach'. In: *Proc. UAI*. pp. 206–215.

Friedman, N., M. Linial, I. Nachman, and D. Pe'er: 2000, 'Using Bayesian Networks to Analyze Expression Data'. *Computational Biology*. To appear. A preliminary version appeared in *Prof. Fourth Fourth Annual International Conference on Computational Molecular Biology*, 2000, pages 127–135.

Gelfand, A. and A. Smith: 1990, 'Sampling based approaches to calculating marginal densities'. *Journal American Statistical Association* **85**, 398–409.

Gilks, W., S. Richardson, and D. Spiegelhalter: 1996, *Markov Chain Monte Carlo Methods in Practice*. CRC Press.

Giudici, P. and P. Green: 1999, 'Decomposable graphical Gaussian model determination'. *Biometrika* **86**(4), 785–801.

Giudici, P., P. Green, and C. Tarantola: 2000, 'Efficient model determination for discrete graphical models'. *Biometrika*. To appear.

Green, P.: 1995, 'Reversible jump Markov chain Monte Carlo computation and Bayesian model determination'. *Biometrika* **82**, 711–732.

Heckerman, D.: 1998, 'A tutorial on learning with Bayesian networks'. In: M. I. Jordan (ed.): *Learning in Graphical Models*. Dordrecht, Netherlands: Kluwer.

Heckerman, D. and D. Geiger: 1995, 'Learning Bayesian networks: a unification for discrete and Gaussian domains'. In: P. Besnard and S. Hanks (eds.): *Proc. Eleventh Conference on Uncertainty in Artificial Intelligence (UAI '95)*. San Francisco: Morgan Kaufmann, pp. 274–284.

Heckerman, D., D. Geiger, and D. M. Chickering: 1995, 'Learning Bayesian Networks: The combination of knowledge and statistical data'. *Machine Learning* **20**, 197–243.

Heckerman, D., C. Meek, and G. Cooper: 1997, 'A Bayesian Approach to Causal Discovery'. Technical report. Technical Report MSR-TR-97-05, Microsoft Research.

Lander, E.: 1999, 'Array of hope'. *Nature Genetics* **21**(1), 3–4.

Larrañaga, P., C. Kuijpers, R. Murga, and Y. Yurramendi: 1996, 'Learning Bayesian network structures by searching for the best ordering with genetic algorithms'. *IEEE Transactions on System, Man and Cybernetics* **26**(4), 487–493.

Liu, J., W. Wong, and A. Kong: 1994, 'Coveriance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes'. *Biometrika* **81**(1), 27–40.

Madigan, D., S. Andersson, M. Perlman, and C. Volinsky: 1996, 'Bayesian model averaging and model selection for Markov equivalence classes of acyclic graphs'. *Communications in Statistics: Theory and Methods* **25**, 2493–2519.

Madigan, D. and E. Raftery: 1994, 'Model selection and accounting for model uncertainty in graphical models using Occam's window'. *Journal Americal Statistical Association* **89**, 1535–1546.

Madigan, D. and J. York: 1995, 'Bayesian graphical models for discrete data'. *International statistical Review* **63**, 215–232.

Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller: 1953, 'Equation of state calculation by fast computing machines'. *Journal of Chemical Physics* **21**, 1087–1092.

Murphy, P. M. and D. W. Aha: 1995, 'UCI Repository of Machine Learning Databases'. `http://www.ics.uci.edu/$\sim$mlearn/MLRepository.html`.

Pearl, J.: 1988, *Probabilistic Reasoning in Intelligent Systems*. San Francisco, Calif.: Morgan Kaufmann.

Pereira, F. and Y. Singer: 1999, 'An efficient extension to mixture techniques for prediction and decision trees'. *Machine Learning* **36**(3), 183–199.

Spellman, P., G. Sherlock, M. Zhang, V. Iyer, K. Anders, M. Eisen, P. Brown, D. Botstein, and Futcher: 1998, 'Comprehensive identification of cell cycle-regulated genes of the yeast *Sacccharomyces cerevisiae* by microarray hybridization'. *Molecular Biology of the Cell* **9**, 3273–3297.

Spirtes, P., C. Glymour, and R. Scheines: 1993, *Causation, Prediction and Search*, No. 81 in Lecture Notes in Statistics. New York: Springer-Verlag.

Wallace, C., K. Korb, and H. Dai: 1996, 'Causal Discovery via MML'. In: *Proc. 13th International Conference on Machine Learning*. pp. 516–524.