Annealed Importance Sampling for Structure Learning in Bayesian Networks*

Teppo Niinimäki and **Mikko Koivisto**
HIIT & Department of Computer Science, University of Helsinki, Finland
teppo.niinimaki@cs.helsinki.fi, mikko.koivisto@cs.helsinki.fi

Abstract

We present a new sampling approach to Bayesian learning of the Bayesian network structure. Like some earlier sampling methods, we sample linear orders on nodes rather than directed acyclic graphs (DAGs). The key difference is that we replace the usual Markov chain Monte Carlo (MCMC) method by the method of annealed importance sampling (AIS). We show that AIS is not only competitive to MCMC in exploring the posterior, but also superior to MCMC in two ways: it enables easy and efficient parallelization, due to the independence of the samples, and lower-bounding of the marginal likelihood of the model with good probabilistic guarantees. We also provide a principled way to correct the bias due to order-based sampling, by implementing a fast algorithm for counting the linear extensions of a given partial order.

1 Introduction

To learn the structure of a *Bayesian network* (BN) [Pearl, 1988; 2000] from data, the Bayesian paradigm [Buntine, 1991; Cooper and Herskovits, 1992; Madigan and York, 1995] offers many appealing features, such as an explicit way to enter prior knowledge and full characterization of uncertainty about the quantities of interest, including proper treatment of potential non-identifiability issues. A major drawback of the Bayesian approach is, however, its large computational requirements. Indeed, the space of structures, namely *directed acyclic graphs* (DAGs), grows rapidly with the number of nodes in the network, and—even if a single good DAG could be found relatively fast—exploring the whole landscape of DAGs to some sufficient extent presents a difficult computational challenge. Currently, the fastest exact algorithms scale up to around 25-node instances [Koivisto and Sood, 2004; Koivisto, 2006; Tian and He, 2009].

The Markov chain Monte method has revolutionized applied mathematics in general [Diaconis, 2009] and the practice of Bayesian statistics in particular [Cappé and Robert, 2000], and it has been applied in various forms also to structure learning in BNs. We summarize some corner stones of the developments. Madigan and York [1995] presented a Markov chain that moves in the space of DAGs by simple arc changes. Friedman and Koller [2003] obtained a significantly faster-mixing chain by operating, not directly on DAGs, but in the much smaller and smoother space of node orderings. A drawback of the sampler, *order-MCMC* in the sequel, is that it introduces a bias favoring DAGs that are compatible with many node orderings. Ellis and Wong [2008] enhanced order-MCMC in two dimensions: a sophisticated sampler exploiting tempered distributions was implemented, and a heuristic for correcting the bias was introduced. Niinimäki *et al.* [2011] extended order-MCMC in another dimension, by showing that sampling suitable partial orders, instead of linear orders, further improves the mixing of the Markov chain, with negligible computational overhead. Also other refinements to Madigan and York's sampler have been presented [Eaton and Murphy, 2007; Grzegorczyk and Husmeier, 2008; Corander *et al.*, 2008], however with somewhat more limited advantages over order-MCMC.

While the current MCMC methods for structure learning seem to work well in many cases, they, unfortunately, fail to satisfy some key desiderata:

1. *Guarantees.* We would like to know how the algorithm's output relates to the quantity of interest. For instance, is it a lower bound, an upper bound, or an approximation to within some multiplicative or additive term? (Existing MCMC methods offer such guarantees only in the limit of running the algorithm infinitely many steps.)
2. *Parallelizability.* We would like to fully exploit parallel computation, that is, to run the algorithm in parallel on thousands of processors, preferably without frequent synchronization or communication between the parallel processes. (Existing MCMC methods are designed rather for a small number of very long runs.)
3. *Bias correction.* We would like to take advantage of the reduced state space (whether total or partial orders), yet enable the use of the uniform prior on (Markov-equivalent) DAGs. (Existing MCMC methods rely on heuristic arguments to correct the bias and may become computationally infeasible with small data; we elaborate on this in Section 2.2.)

*This research is supported in part by the Academy of Finland, grants 125637, 218153, and 255675.

The present work makes a step toward satisfying these three desiderata. Motivated by the first desire for guarantees, we seek a sampler such that we know exactly from which distribution the samples are drawn. Here, the method of *annealed importance sampling* (AIS) by Neal [2001] provides an appealing solution. It enables drawing independent and identically distributed samples and computing the associated importance weights, so that the expected value of each weighted sample matches the quantity of interest. In this setting, already a small number of samples may suffice, not only for finding an accurate estimate, but also for showing a relatively tight, high-confidence lower bound on the true value [Gomes *et al.*, 2007; Gogate *et al.*, 2007; Gogate and Dechter, 2011]. The independence of the samples also readily offers an easy way to parallelize the computations, thus satisfying the second desideratum. Finally, we address the issue of bias correction by computing the required correction term explicitly. This amounts to implementing an efficient algorithm for counting the number of topological sortings of a given DAG, or in other words, the linear extensions of the corresponding partial order. The problem is #P-hard [Brightwell and Winkler, 1991], which may have lead to the search for indirect solutions in earlier works. However, as we will show in this paper, a careful implementation of a dynamic programming algorithm allows us to count the number of linear extensions exactly up to around 40-node instances, typically within a few seconds.

The purpose of this report is to communicate the main ideas and motivation underlying the proposed approach, demonstrate the potential of the approach, identify its current bottlenecks, and discuss the prospects of future research in the suggested direction. We have targeted our experiments to study rather specific questions using a few benchmark data sets. Extensive experimentation is left for future work.

We note that previously Battle *et al.* [2010] have applied AIS to structure learning in Bayesian networks in a molecular biology application. The key difference to our method is that, instead of node orderings, they sample fully specified network models. They motivate the choice of AIS mainly by the complexity and multi-modality posterior distribution, which often lead to slow mixing of usual MCMC methods.

2 Preliminaries

We begin by describing the structure learning problem, mostly adopting the notation of Niinimäki *et al.* [2011]. We then briefly review Friedman and Koller's [2003] order-MCMC method and extend it based on Geyer's [1991] Metropolis-coupled MCMC (MC^3). In our experiments, reported in Section 3, we use MC^3 as a proxy of a related implementation[1] by Ellis and Wong [2008]. We also discuss the possible drawbacks of Ellis and Wong's [2008] heuristic for correcting the bias of order-MCMC.

2.1 Structure Learning in Bayesian Networks

A *Bayesian network* (BN) represents a joint distribution of a vector of random variables $D = (D_1, \ldots, D_n)$ in terms of a directed acyclic graph (DAG) (N, A), where each node $v \in N = \{1, \ldots, n\}$ corresponds to a random variable D_v, and the arc set $A \subseteq N \times N$ specifies the *parent set* of the node, $A_v = \{u \in N : uv \in A\}$. For each node v and its parent set A_v the BN specifies a local conditional distribution $p(D_v|D_{A_v}, A_v)$, and composes the joint distribution of D as

$$p(D|A) = \prod_{v \in N} p(D_v|D_{A_v}, A_v)\,.$$

Here and henceforth, we identify the DAG with its arc set A, with the understanding that the node set N is fixed while varying configurations for A are of our interest.

Indeed, we treat the arc set A as a random variable and consider the Bayesian approach to learn A from observed values of D, called *data*. In this setting it is customary to extend the BN model to the case where D is not a single vector but a matrix of random variables, consisting of m vectors, so that each D_v is a tuple of m random variables. Usual assumptions of exchangeability or modularity concerning the parametrization of the local conditional distributions, keep the above factorization of $p(D|A)$ valid. Under the most popular specifications, this term can be efficiently computed for a given A [Heckerman *et al.*, 1995]. By choosing some prior distribution $p(A)$, the posterior distribution is obtained via the Bayes rule: $p(A|D) = p(A)p(D|A)/p(D)$, where is $p(D)$ is the *marginal likelihood* of the model, sometimes also called the *evidence* or the *normalization constant*. In addition to this fundamental quantity, the user is often interested in posterior expectations of various structural features $f(A)$. For example, $f(A)$ can be the indicator function of some particular arc of interest, evaluating to 1 if the arc is present in A, and to 0 otherwise [Friedman and Koller, 2003].

It is common to assign a modular prior, that is, $p(A) \propto q(A) = \prod_v q_v(A_v)$ for some nonnegative functions q_v. For instance, the uniform prior on DAGs is obtained by setting simply $q_v(A_v) \equiv 1$. Often the support of the prior is restricted to DAGs in which the *indegree* of every node is at most some constant k, that is, $q_v(A_v) = 0$ when $|A_v| > k$.

Motivated by computational issues, Friedman and Koller [2003] introduced another class of priors, in which the model is augmented with a new variable, a *linear order* on the nodes, $L \subseteq N \times N$, or *node ordering* less formally. We call a linear order L an *extension* of A if $A \subseteq L$. Now, we assign a joint prior $p(A, L)$ that vanishes whenever $A \not\subseteq L$, and $p(A, L) \propto q(A)\rho(L)$ otherwise, where $q(A)$ is as before and $\rho(L)$ is some nonnegative function; in the sequel, we take simply $\rho(L) \equiv 1$. The motivation of this construction stems from the factorization

$$\begin{aligned} p(D|L) &= \sum_A p(A|L)p(D|A) \\ &\propto \prod_v \sum_{A_v \subseteq L_v} q_v(A_v)p(D_v|D_{A_v}, A_v)\,, \end{aligned} \quad (1)$$

where $L_v = \{u \in N : uv \in L\}$ is the set of nodes that precede v in the order. By modeling different node orderings as mutually exclusive events, the factorization enables efficient treatment of DAGs via node orderings. A drawback of this augmentation is, however, that it introduces a systematic bias

[1] The software used by Ellis and Wong [2008] is not publicly available at the moment (W.H. Wong, personal discussion).

from the simpler modular prior on DAGs. Namely, letting $\ell(A)$ denote the *number of linear extensions* of A, we have $p(A) \propto q(A) \sum_{L \supseteq A} \rho(L) = q(A)\ell(A)$. That is, the prior is biased due to the extra factor $\ell(A)$, which is reflected also in the posterior; for more thorough discussions, see the references [Friedman and Koller, 2003; Koivisto and Sood, 2004; Eaton and Murphy, 2007; Ellis and Wong, 2008].

2.2 Order-MCMC and Correction

Friedman and Koller's [2003] order-MCMC samples first node orderings and then DAGs that are compatible with the sampled orderings:

1. *Sample node orderings along a Markov chain.* Start from a random node ordering. To move, propose the swapping of the positions of two random nodes in the current order. Accept the proposal according to the Metropolis–Hastings ratio, such that the stationary distribution of the resulting Markov chain is the posterior of node orderings. Evaluate the posterior probability of a node ordering efficiently using the factorization (1). The accepted states yield a sample of node orderings, $L^{(1)}, L^{(2)}, \ldots, L^{(T)}$, possibly after thinning and discarding some burn-in period.
2. *Sample DAGs from the sampled orders.* For each sampled node ordering L, generate a DAG compatible with the order from the posterior distribution $p(A|L, D)$. Again this can be done efficiently by exploiting the independence of the parent sets A_v given the order L. This produces a sample of DAGs $A^{(1)}, A^{(2)}, \ldots, A^{(T)}$.

If the feature of interest f is modular, i.e., $f(A) = \prod_v f_v(A_v)$, then an estimate for the posterior expectation of f is obtained using only the sampled orderings, as $\sum_t f(L^{(t)})/T$, where $f(L)$ denotes the posterior expectation of $f(A)$ under the constraint $A \subseteq L$, that is, $f(L) = \sum_A f(A)p(A|D, L)$. This, again, can be evaluated relatively fast for any given L due to a factorization similar to (1). For non-modular features, the sampled DAGs are used, the estimate being simply $\sum_t f\big(A^{(t)}\big)/T$.

In principle, the bias due to favoring DAGs that have many linear extensions can be corrected by simple reweighting. The bias is removed in the estimate

$$\sum_t \frac{f\big(A^{(t)}\big)}{\ell\big(A^{(t)}\big)} \Big/ \sum_t \frac{1}{\ell\big(A^{(t)}\big)} \,.$$

A difficulty here is, however, that computing $\ell(A)$ for a given DAG A is #P-hard in general [Brightwell and Winkler, 1991].

To circumvent the computation of the terms $\ell\big(A^{(t)}\big)$, Ellis and Wong [2008] proposed the following heuristic: From each unique sampled order L, sample a set of unique DAGs such that their total posterior mass, conditionally on L, is at least $1 - \epsilon$. Let $\mathcal{U}$ be the union of all the DAGs so obtained, over all the sampled orders L. Treat $\mathcal{U}$ as an importance-weighted sample, in which the weight of a DAG A is given by $p(A)/\sum_{A \in \mathcal{U}} p(A)$.

While this heuristic corrects the bias reliably when the sampled orders and DAGs cover nearly all the mass of the posterior, it may fail in other typical cases. To see this, consider what happens when $\epsilon = 0$ and one samples only a single order L. Then, a calculation shows that the expected value of the "corrected" estimator for the posterior expectation of $f(A)$ is still proportional to $\sum_A f(A)p(A|D)\ell(A)$, thus unchanged and biased. For another example, consider a scenario where the posterior probability is $1/2$ for a special DAG A_0 and $1/(2M)$ for some other M DAGs. Assume also that $f(A_0) = 1$ and $f(A) = 0$ for $A \neq A_0$, and that every DAG in question has exactly one linear extension. Now, suppose $T = M$ DAGs are sampled, again with $\epsilon = 0$. Then, with high probability, nearly one half of the DAGs equal A_0 the rest $T/2$ being unique. Since only unique DAGs are kept for the estimator, the corrected estimate is about $(1/2)/(1/2 + T/(4M)) = 2/3$, whereas the correct value is $1/2$. Taking fewer samples rapidly worsens the situation. A further concern with the heuristic is its computational requirements when the number of nodes is large, say 25 or larger, but the number of data points is small: the posterior is flat, and consequently, a very large number of DAGs need to be sampled from each order to gather the required mass of $1 - \epsilon$. (We note that Ellis and Wong [2008] limit their experiments to instances of at most 14 nodes and at least 100 data points.)

2.3 MC^3

Extending order-MCMC with tempering techniques can yield still better mixing and also a good estimator for the marginal likelihood $p(D)$. Here we consider one such technique, *Metropolis-coupled MCMC* (MC^3) [Geyer, 1991]. In MC^3 several Markov chains, indexed by $0, 1, \ldots, r$, are simulated in parallel, each chain i having its own stationary distribution p_i. The idea is to take p_0 as a "hot" distribution, e.g., the uniform distribution, and then let the p_i be increasingly "cooler" and closer approximations of the posterior p, putting finally $p_r = p$. Usually, powering schemes of the form

$$p_i \propto p^{\beta_i} \,, \quad 0 \leq \beta_0 < \beta_1 < \ldots < \beta_r = 1$$

are used. For instance, Geyer and Thompson [1995] suggest harmonic stepping, $\beta_i = 1/(r+1-i)$; in our experiments we have used linear stepping, $\beta_i = i/r$. In addition to running the chains in parallel, every now and then we propose a swap of the states L_i and L_j of two randomly chosen chains i and $j = i + 1$. The proposal is accepted with probability

$$\min\left\{1, \frac{p_i(L_j)p_j(L_i)}{p_i(L_i)p_j(L_j)}\right\}.$$

We note that each p_i needs to be known only up to some constant factor, that is, we can efficiently evaluate a function g_i that is proportional to p_i. We denote by Z_i the normalization constant g_i/p_i.

Having T samples from each chain, the ratio Z_r/Z_0 can be estimated by the telescoping product of the estimates

$$\frac{Z_i}{Z_{i-1}} \approx \frac{1}{T} \sum_t g_i\big(L_{i-1}^{(t)}\big)/g_{i-1}\big(L_{i-1}^{(t)}\big) \,, \quad i = 1, \ldots, r \,.$$

When p_0 is taken as the uniform distribution, the constant Z_0 is known or easy to estimate, and so an estimate of the ratio Z_r/Z_0 gives us an estimate of the marginal likelihood $p(D)$.

3 AIS on Node Orderings

We next describe our application of the AIS method, our approach to correct the bias by explicitly counting the linear extensions of a given DAG, and the lower bounding method based on Markov's inequality. We also include experimental results on a few selected data sets.

3.1 Sampling Node Orderings

AIS produces a sample of linear orders $L^{(1)}, \ldots, L^{(T)}$, and corresponding importance weights $w^{(1)}, \ldots, w^{(T)}$. Like in MC^3, a sequence of distributions $p_0, p_1, \ldots, p_r$ are introduced, such that sampling from p_0 is easy, and as i increases, the distributions p_i provide gradually improving approximations to the posterior distribution p, until finally p_r equals p. For each p_i we assume the availability of a corresponding function g_i that is proportional to p_i and that can be evaluated fast at any given point. To sample $L^{(t)}$, we first sample a sequence of linear orders, $L_0, L_1, \ldots, L_{r-1}$ along a Markov chain, starting from p_0 and moving according to suitably defined transition kernels τ_i, as follows:

Generate L_0 from p_0.
Generate L_1 from L_0 using τ_1.
$\vdots$
Generate L_{r-1} from L_{r-2} using τ_{r-1}.

The transition kernels τ_i are constructed by a simple Metropolis move: at state L_{i-1} a new state L' is proposed by swapping the positions of two random nodes in the order; the proposal is accepted as the state L_i with probability $\min\{1, g_i(L')/g_i(L_{i-1})\}$, and otherwise L_i is set to L_{i-1}. It follows that τ_i leaves p_i invariant. Finally, we set $L^{(t)} = L_{r-1}$ and assign the importance weight as

$$w^{(t)} = \frac{g_1(L_0)}{g_0(L_0)} \frac{g_2(L_1)}{g_1(L_1)} \cdots \frac{g_r(L_{r-1})}{g_{r-1}(L_{r-1})} .$$

Such a weighted sample in hand, the ratio of the normalization constants $Z = g_r/p$ and $Z_0 = g_0/p_0$ can be estimated by $Z/Z_0 \approx \sum_t w^{(t)}/T$. Likewise, the expectation of a function f with respect to the posterior p can be estimated by $\sum_t w^{(t)} f(L^{(t)}) / \sum_t w^{(t)}$. That these estimators are well justified, follows directly from Neal's argumentation [2001].

3.2 Sampling DAGs and Correcting the Bias

Similar to order-MCMC, sampling DAGs from the sampled node orderings enables posterior inference for nonmodular features and a way to correct the bias. Specifically, if $A^{(t)}$ is a sample from the posterior distribution of DAGs compatible with a node ordering $L^{(t)}$, then we see that the corrected importance sampling estimate for the posterior expectation of f takes the form

$$\sum_t \frac{w^{(t)} f(A^{(t)})}{\ell(A^{(t)})} \Big/ \sum_t \frac{w^{(t)}}{\ell(A^{(t)})} .$$

Likewise, the corrected estimate for the ratio Z/Z_0 becomes

$$\frac{1}{T} \sum_t \frac{w^{(t)}}{\ell(A^{(t)})} .$$

Since sampling a single order $L^{(t)}$ is relatively expensive, it is often advisable to draw several DAGs per sampled order, to reduce the variance of the estimator. If two DAGs, $A^{(t_1)}$ and $A^{(t_1)}$ are generated from the same order $L^{(t)}$, the associated terms in the above sums share the weight $w^{(t)}$ are no longer independent random variables. However, while this will change the variance of the respective estimators, their expectations remain unchanged. What is a reasonable number of DAGs per order, depends on the complexity of evaluating the term $\ell(A^{(t)})$; we consider this issue next.

3.3 Counting Linear Extensions

To count the linear extensions of a DAG $A \subseteq N \times N$, we turn to the corresponding partial order $P(A)$ on N, obtained from A by adding the loops vv for each $v \in N$ (making the relation reflexive) and then taking the transitive closure (making the relation transitive). Clearly, A and $P(A)$ have the same linear extensions. Since the transitive closure $P(A)$ of a given A can be computed efficiently (e.g., in $O(n^3)$ time using the Floyd–Warshall algorithm), it suffices to consider the problem of counting the linear extensions of a given partial order.

It is easy to see that the number of linear extensions of a partial order P on set S satisfies the recurrence

$$\ell(P) = \sum_{v \text{ is maximal in } P} \ell(P - v) ,$$

with $\ell(\emptyset) = 1$, where $P - v$ is the partial order on $S \setminus \{v\}$ obtained from P by deleting all pairs that mention v. This recurrence immediately suggests a dynamic programming algorithm that tabulates the number of linear extensions for all subsets that are downward closed with respect to the partial order, known as the *downsets* or *ideals* of the partial order.

A straightforward implementation runs in time that scales as $n^2 I(P)$, where $I(P)$ is the number of ideals of P. We have further reduced the time requirement to $O(nI(P))$ by implementing an efficient way to maintain the set of maximal elements while traversing through the ideals. Since $I(P)$ is typically much less than its worst-case bound 2^n, our implementation scales well up to around 40-element instances even if the DAG underlying the partial order is relatively sparse; see Figure 1. The bottleneck of the implentation is in fact the space requirement which grows roughly as $I(P)$.

3.4 Lower Bounding

We will use the following elementary but useful consequence of Markov's inequality; for a proof and variations, see the works of Gomes *et al.* [2007] and Gogate and Dechter [2011].

Theorem 1. *Let $X_1, X_2, \ldots, X_s$ be independent nonnegative random variables with mean μ. Let $0 < \delta < 1$. Then, with probability at least $1 - \delta$,*

$$\delta^{1/s} \min\{X_1, X_2, \ldots, X_s\} \leq \mu .$$

For example, at confidence $0.95 = 1 - \delta$, a lower bound is obtained by taking the minimum over $s = 5$ samples and multiplying it by $0.05^{1/5} \approx 0.55$, or by taking the minimum over $s = 30$ samples and multiplying it by $0.05^{1/30} \approx 0.90$.

In our experiments we have applied this idea to lower bound the marginal likelihood $p(D)$ as follows: The T samples generated by AIS are partitioned into $s = 9$ bins. For the

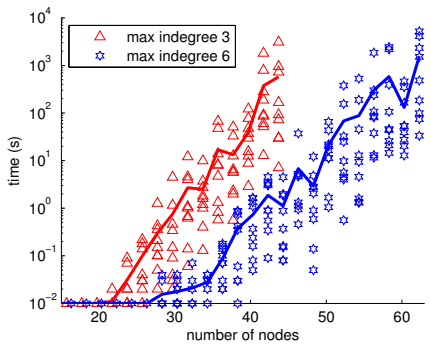

Figure 1: Speed of exact counting of linear extensions. For each number of nodes, the runtime (in seconds) is shown for 9 random DAGs, with the maximum indegree set to 3 or 6. Runtimes less than 0.01 seconds are rounded up to 0.01. The mean of the runtimes is shown by a solid line. No runtime estimates are shown when the memory requirement exceeded 16GB for at least one of the 9 DAGs.

jth bin, let X_j be the corresponding average of $T/9$ weights $w^{(t)}$. By Theorem 1, the minimum over the X_j multiplied by $0.05^{1/9} \approx 0.7168$ is a lower bound of $p(D)/Z_0$ with probability at least 0.95. (We will ignore the term Z_0 as it depends on neither the data nor the prior, except for the maximum indegree, and it could be evaluated, e.g., using Robinson's [1973] recurrences.)

3.5 Experimental Results

We have experimented on three data sets. Our ALARM data set contains 100 data records generated from the 37-variable Alarm network. Mushroom is frequently used a 22-variable data set of 8124 records, of which we included a random subset of size 1000 to our MUSHROOM data set. PROMOTERS is another benchmark data set, consisting of 58 variables and 106 records. See the references [Beinlich *et al.*, 1989; Blake and Merz, 1998] for more about these data sources.

We specified the Bayesian model in a usual manner. For $p(D|A)$ we used the BDeu scoring. We set $q_v(A_v) \equiv 1$ for the modular part, and $\rho(L) \equiv 1$ for the term concerning node orderings. The maximum indegree parameter k was set to 4 for MUSHROOM and to 3 for ALARM and PROMOTERS. For MUSHROOM we used exact algorithms [Koivisto, 2006; Tian and He, 2009] to compute the correct arc posterior probabilities, and for the biased prior, also the marginal likelihood.

For a fair comparison of AIS and MC^3, we fixed or varied the free parameters as follows: For both methods, we used linear stepping in the annealing scheme. (We experimented also with other schemes, but the linear one achieved the most robust performance across the data sets and repeated runs.) However, the number of steps was varied: for MC^3 we put $r = 4, 16, 64$; for AIS we put $r = nm, 4nm, 16nm$, thus depending on the data size. These choices reflect the fact that in MC^3 it is crucial to simulate each parallel chain a large number of steps to get sufficient mixing and convergence, whereas in AIS it is crucial to simulate the annealed chain a large number of steps. While we gave each of the six samplers roughly 200 hours of running time, we measured the performance of the corresponding estimators throughout the process. Consequently, at any given time budget samplers with larger r yielded respectively fewer samples. For each run of MC^3, we and included every 100th (MUSHROOM and ALARM) or 10th (PROMOTERS) simulated states in the sample. The first half of the samples were discarded as burn-in. From each sampled node ordering a single DAG was generated for bias correction, except for AIS on MUSHROOM we generated 10 DAGs per node ordering to balance the time requirements of sampling and counting linear extensions.

We first studied the performance of AIS and MC^3 in estimating the marginal likelihood (Figure 2). We observed that all the six samplers typically produced a good estimate, the larger number of steps paying off when given sufficiently long running time. An exception is the MC^3 sampler with $r = 4$ on MUSHROOM. We also observed that, for MUSHROOM, AIS (e.g., with $r = 4nm$ steps) finally yields a lower bound that is within a factor of about 2 of the exact value. For ALARM and PROMOTERS the lower bounds are even better (compared to the median estimate); in fact, they are nearly as good as they can get by taking the minimum over 9 estimates, which necessarily implies an error of at least $\ln 0.7168 \approx -0.33$ in the logarithm of the marginal likelihood.

We then investigated the performance of the proposed bias-corrected estimators (Figure 3, left). (Here we do not have results for PROMOTERS, as the number of nodes, 58, turned out to be too large for counting the linear extensions, given the small maximum indegree of 3.) We found moderate increase in the variance of the estimates, compared to the case of not correcting the bias. The lower bounds were now within a factor of about 2 and 3 of the median estimate over 9 runs, for MUSHROOM and ALARM, respectively.

Finally, we compared the performance of AIS and MC^3 with and without bias correction in estimating the arc posterior probabilities (Figure 3, right). We include results only for MUSHROOM, for which the exact values are available, and only for the best choice of the number of annealing steps r. We measured the performance of a single run by the largest absolute error over all possible arcs. We observed that bias correction does not significantly worsen the accuracy of the estimates. We also observed that while MC^3 is somewhat more efficient than AIS, the MUSHROOM data set presents a difficult challenge for all the methods studied here.

4 Conclusions and Future Work

We have contributed two new ideas to structure learning in Bayesian networks: (1) the use of Neal's [2001] annealed importance sampling (AIS) method to obtain independent samples with known expected value, and (2) the use of moderately-exponential-time dynamic programming algorithms to correct the bias due to sampling node orderings. By comparing to an advanced MCMC technique, MC^3, we have shown that the efficiency of AIS in exploring the posterior is competitive—AIS does not sacrifice accuracy for its other advantages. We next discuss to what extent our present implementation satisfies the three desiderata that motivated this work, and what are the future prospects in this regard.

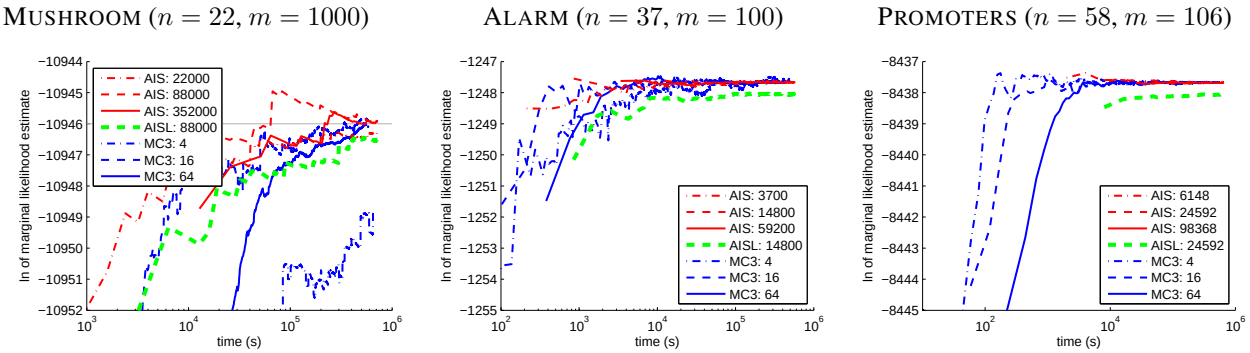

Figure 2: Performance of AIS and MC^3 in estimating the marginal likelihood, without bias correction. For each of the six samplers, the median (of 9 runs) of the estimates is shown as a function of running time. In addition, a 0.95-confidence lower bound based on AIS with the medium number of steps is shown ("AISL"). For MUSHROOM the correct value is shown by a horizontal line. The legend shows the number of annealing steps r.

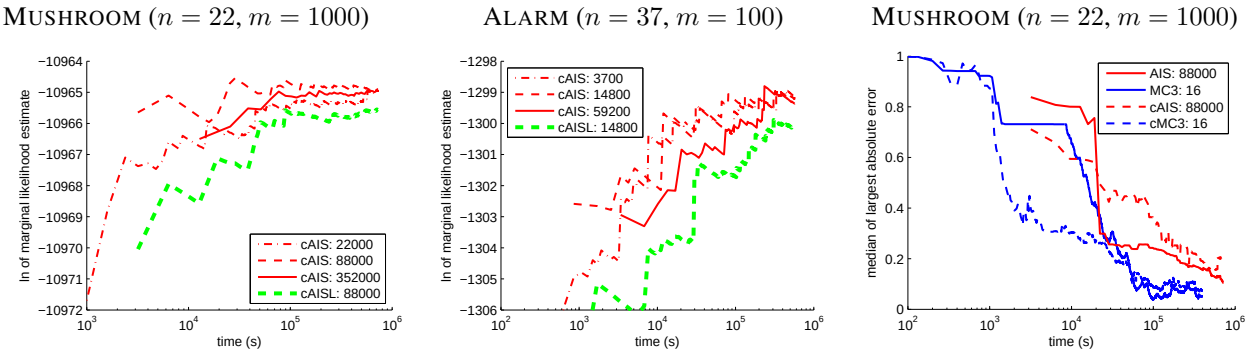

Figure 3: *Left*: Performance of AIS in estimating the marginal likelihood, with bias correction. For each sampler, the median (of 9 runs) of the estimates is shown as a function of running time. In addition, a 0.95-confidence lower bound based on AIS with the medium number of steps is shown ("cAISL"). The legend shows the number of annealing steps r. *Rightmost*: Performance of AIS and MC^3 in estimating the arc posterior probabilities, with and without bias correction. For each sampler, the median (of 9 runs) of the largest absolute error over all possible arcs is shown as a function of running time.

Bias correction. We implemented a direct way to correct the bias by explicitly counting the linear extensions of each sampled DAG. The main advantage of this approach is that it yields a weighting that provably corrects the bias. We showed that the #P-hardness of the problem does not render it intractable in practice, as long as the number of nodes is at most about 40—this is much beyond the scope of the existing dynamic programming algorithms for structure learning [Koivisto, 2006; Eaton and Murphy, 2007; Tian and He, 2009]. A topic of future research is to improve the exact algorithm further and to explore the practical value of the existing polynomial-time approximation schemes [Dyer *et al.*, 1991; Karzanov and Khachiyan, 1991; Huber, 2006].

Parallelization. A main advantage of AIS over MCMC is that AIS enables easy large-scale parallelization. There do exist parallel implementations of MCMC [Altekar *et al.*, 2004; Corander *et al.*, 2008], which rely on frequent synchronized communication. They are thus only suited for architectures with shared memory and some tens of parallel processes. In AIS the bottleneck is the complexity of simulating the annealing process to get a single sample. As it is vital to keep the number of simulation steps large, the hope for scalability to larger instances is in developing significantly faster algorithms for evaluating a given node ordering. Our current implementation can be boosted by an order of magnitude by incorporating some of the tricks of Friedman and Koller [2003].

Guarantees. We showed that AIS supplies an importance sampling distribution that enables lower-bounding of the marginal likelihood of the model with fairly good probabilistic guarantees. In obtaining quality guarantees for sampling-based estimates more generally, our success is, admittedly, only partial so far. Most importantly, we currently do not obtain useful lower (or upper) bounds for the posterior probabilities of individual arcs or larger subgraph, since we do not have useful upper bound for the involved normalization constant, that is, the marginal likelihood. In this light, the key topic of future research is to find efficient methods for upper-bounding the marginal likelihood.

References

[Altekar *et al.*, 2004] G. Altekar, S. Dwarkadas, J. P. Huelsenbeck, and F. Ronquist. Parallel Metropolis coupled Markov chain Monte Carlo for Bayesian phylogenetic inference. *Bioinformatics*, 20:407–415, 2004.

[Battle *et al.*, 2010] A. J. Battle, M. Jonikas, P. Walter, J. Weissman, and D. Koller. Automated identification of pathways from quantitative genetic interaction data. *Molecular Systems Biology*, 6:379–391, June 2010.

[Beinlich *et al.*, 1989] I. Beinlich, G. Suermondt, R. Chavez, and G. F. Cooper. The ALARM monitoring system. In *Proc. of the Second European Conference on Artificial Intelligence and Medicine*, pages 247–256, 1989.

[Blake and Merz, 1998] C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998.

[Brightwell and Winkler, 1991] G. Brightwell and P. Winkler. Counting linear extensions. *Order*, 8:225–242, 1991.

[Buntine, 1991] W. Buntine. Theory refinement on Bayesian networks. In B. D'Ambrosio and P. Smets, editors, *UAI*, pages 52–60. Morgan Kaufmann, 1991.

[Cappé and Robert, 2000] O. Cappé and C. P. Robert. Markov chain Monte Carlo: 10 years and still running! *Journal of the American Statistical Association*, 95:1282–1286, 2000.

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

[Corander *et al.*, 2008] J. Corander, M. Ekdahl, and T. Koski. Parallel interacting MCMC for learning of topologies of graphical models. *Data Min. Knowl. Discov.*, 17(3):431–456, 2008.

[Diaconis, 2009] P. Diaconis. The Markov chain Monte Carlo revolution. *Bull. Amer. Soc.*, 46(2):179–205, 2009.

[Dyer *et al.*, 1991] M. E. Dyer, A. M. Frieze, and R. Kannan. A random polynomial time algorithm for approximating the volume of convex bodies. *J. ACM*, 38(1):1–17, 1991.

[Eaton and Murphy, 2007] D. Eaton and K. P. Murphy. Bayesian structure learning using dynamic programming and MCMC. In Parr and van der Gaag [2007], pages 101–108.

[Ellis and Wong, 2008] B. Ellis and W. H. Wong. Learning causal Bayesian network structures from experimental data. *Journal of the American Statistical Association*, 103:778–789, 2008.

[Friedman and Koller, 2003] N. Friedman and D. Koller. Being Bayesian about network structure: A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, 50:95–125, 2003.

[Geyer and Thompson, 1995] C. J. Geyer and E. A. Thompson. Annealing Markov chain Monte Carlo with application to ancestral inference. *Journal of the American Statistical Association*, 90:909–920, 1995.

[Geyer, 1991] C. J. Geyer. Markov chain Monte Carlo maximum likelihood. In E. M. Keramidas, editor, *23rd Symposium on Interface*, pages 156–163, Fairfax Station: Interface Foundation, 1991.

[Gogate and Dechter, 2011] V. Gogate and R. Dechter. Sampling-based lower bounds for counting queries. *Intelligenza Artificiale*, 5(2):171–188, 2011.

[Gogate *et al.*, 2007] V. Gogate, B. Bidyuk, and R. Dechter. Studies in lower bounding probabilities of evidence using the Markov inequality. In Parr and van der Gaag [2007], pages 141–148.

[Gomes *et al.*, 2007] C. P. Gomes, J. Hoffmann, A. Sabharwal, and B. Selman. From sampling to model counting. In M. M. Veloso, editor, *IJCAI*, pages 2293–2299, 2007.

[Grzegorczyk and Husmeier, 2008] M. Grzegorczyk and D. Husmeier. Improving the structure MCMC sampler for Bayesian networks by introducing a new edge reversal move. *Machine Learning*, 71(2-3):265–305, 2008.

[Heckerman *et al.*, 1995] D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.

[Huber, 2006] M. Huber. Fast perfect sampling from linear extensions. *Discrete Mathematics*, 306(4):420–428, 2006.

[Karzanov and Khachiyan, 1991] A. Karzanov and L. Khachiyan. On the conductance of order Markov chains. *Order*, 8:7–15, 1991.

[Koivisto and Sood, 2004] M. Koivisto and K. Sood. Exact Bayesian structure discovery in Bayesian networks. *Journal of Machine Learning Research*, 5:549–573, 2004.

[Koivisto, 2006] M. Koivisto. Advances in exact Bayesian structure discovery in Bayesian networks. In *UAI*, pages 241–248. AUAI, 2006.

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

[Neal, 2001] R. M. Neal. Annealed importance sampling. *Statistics and Computing*, 11:125–139, 2001.

[Niinimaki *et al.*, 2011] T. Niinimaki, P. Parviainen, and M. Koivisto. Partial order MCMC for structure discovery in Bayesian networks. In F. G. Cozman and A. Pfeffer, editors, *UAI*, pages 557–564. AUAI, 2011.

[Parr and van der Gaag, 2007] R. Parr and L.-C. van der Gaag, editors. *UAI 2007, Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence, Vancouver, BC, Canada, July 19-22, 2007*. AUAI Press, 2007.

[Pearl, 1988] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco, 1988.

[Pearl, 2000] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, 2000.

[Robinson, 1973] R. W. Robinson. Counting labeled acyclic digraphs. In F. Harary, editor, *New Directions in Graph Theory*. Academic Press, 1973.

[Tian and He, 2009] J. Tian and R. He. Computing posterior probabilities of structural features in Bayesian networks. In J. Bilmes and A. Y. Ng, editors, *UAI*, pages 538–547. AUAI, 2009.