Bayesian structure learning using dynamic programming and MCMC

Daniel Eaton
Computer Science Dept.
University of British Columbia
deaton@cs.ubc.ca

Kevin Murphy
Computer Science Dept.
University of British Columbia
murphyk@cs.ubc.ca

Abstract

We show how to significantly speed up MCMC sampling of DAG structures by using a powerful non-local proposal based on Koivisto's dynamic programming (DP) algorithm (11; 10), which computes the exact marginal posterior edge probabilities by analytically summing over orders. Furthermore, we show how sampling in DAG space can avoid subtle biases that are introduced by approaches that work only with orders, such as Koivisto's DP algorithm and MCMC order samplers (6; 5).

1 Introduction

Directed graphical models have proved to be a very useful tool for causal modeling (13; 16). One of the key challenges is to learn the structure of these models from data. Often (e.g., in molecular biology) the sample size is quite small relative to the complexity of the model. In such cases, the posterior over graph structures given data, $p(G|D)$, gives support to many possible models, and using a point estimate (such as MAP) could lead to unwarranted conclusions.

Since there are $O(d!2^{\binom{d}{2}})$ DAGs (directed acyclic graphs) on d nodes (14), even just storing the full posterior is intractable for reasonable d. (For example, the number of DAGs on 5 nodes is about 30,000, and on 6 nodes is over 3.5 million.) Instead, it has become common to return samples from the posterior, or to summarize the posterior in terms of marginal edge probabilities, $p(G_{ij} = 1|D)$ (6; 11; 10; 5).

The standard approach to compute these probabilities is to use MCMC, either in the space of DAGs (12; 8), or in the space of node orderings (6; 5). The advantage of sampling over orders is that the space is much smaller and "smoother". Recently a dynamic programming (DP) algorithm has been devised by Koivisto and Sood (11; 10) which can sum over all possible orderings analytically, and thus exactly compute $p(G_{ij} = 1|D)$ for all edges in $O(d2^d)$ time.

Although the DP method is much faster than the MCMC order sampler, it only returns marginal edge probabilities, which contain less information than samples of full DAGs or orders. Furthermore, both the DP order method and MCMC order samplers require that the prior have a special form (which we explain below) that can lead to some undesirable artefacts. In this paper, we propose to overcome both of these drawbacks by using the DP algorithm as a proposal distribution for Metropolis Hastings in the space of DAGs. We show empirically that this represents good "value for money", in the sense that its posterior estimates converge to the true estimates much faster than the other approaches mentioned above.

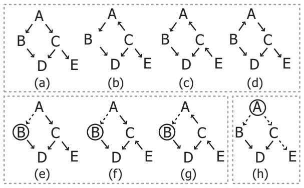

Figure 1: Top left: the "cancer network", from (7). (a-d) are Markov equivalent. (c-g) are equivalent under an intervention on B. (h) is the unique member under an intervention on A. Based on (17).

2 Previous work

In this section, we review related previous work, and point out various flaws in it, in order to motivate our extensions. The MCMC order sampler technique of (6; 5) and the DP technique of (11; 10) represent the state of the art in methods for computing posteriors over DAGs. Unfortunately, we do not have space to explain these methods in detail. For the purposes of this paper, it suffices to know that the input to these algorithms is a local marginal likelihood function for every node and every possible parent set, $p(X_i|X_{G_i})$, a prior over node orderings $q_i(U_i)$, and a prior over possible parent sets, $\rho_i(G_i)$. We now discuss each of these in turn.

For the local marginal likelihoods $p(X_i|X_{G_i})$ we use the standard BDeu score in the case of discrete data and the BGeu score in the case of Gaussian data (9). This can be modified to handle interventional (experimental) data using the simple trick in (3). Such interventional data is crucial for disambiguating between Markov equivalent structures, and hence for inferring causality (see Figure 1).

We now discuss the structural prior. Rather than being able to define an arbitrary prior on graph structures $p(G)$, methods that work with orderings define a joint prior over graphs G and orders $\prec$ as follows:

$$p(\prec, G) \quad = \quad \frac{1}{Z} \prod_{i=1}^{d} q_i(U_i^{\prec}) \rho_i(G_i) \times \text{consistent}(\prec, G)$$

Here $q_i(U_i^{\prec})$ is the prior probability that U_i preceeds i. Throughout this paper, we assume a uniform prior over orderings, $q_i(U_i) = 1$, so $p(\prec) = 1/(d!)$, since typically we do not have prior knowledge on the order. The $\rho_i(G_i)$ term is proportional to the prior that G_i is the parent set of i. One option would be to take $\rho_i(G_i) = 1$; we will call this the "unit rho" prior. In this case, $p(G_i|U_i) = 2^{-|U_i|}$, i.e., the probability of picking a parent set decreases with the parent size. A more popular alternative (used in (11; 10; 6; 5)) is to take $\rho_i(G_i) \propto \binom{d-1}{|G_i|}^{-1}$; we will call this the "nonunit rho" prior.

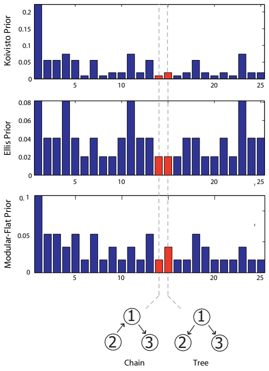

Figure 2: Illustration of some priors on all 25 DAGs on 3 nodes, illustrating the violation of Markov equivalence. Top: the non unit rho prior,. The most probable graph is the empty graph. Middle: the reweighted version of this prior (as proposed by Ellis), that is uniform within Markov equivalence classes but not across classes. The most probable graphs are the empty graphs and all 3 v-structures. Bottom: the unit rho prior. If we reweight this prior, we get a globally uniform prior, but the reweighting is in general #P-hard.

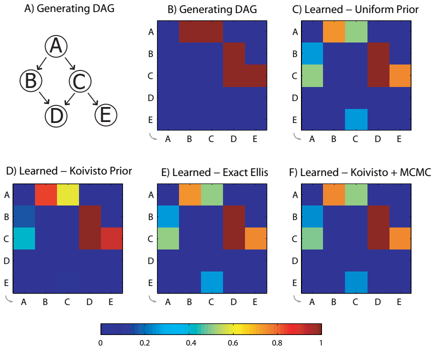

Figure 3: (a-b) The cancer network. (c) Marginal posterior edge probabilities using uniform prior $p(G)$ and a large observational sample. (d) Results using non-unit rho prior. (e) Results using reweighted non-unit rho prior. In this case the results are correct, but this is not always the case. (f) Results using our hybrid method. These are correct, and cheap to compute. This figure is best viewed in colour.

This prior says that different cardinalities of parents are considered to be equally likely a priori.

Of course, G_i and U_i are not independent, since we require $G_i \subseteq U_i$. Hence $q_i(U_i)$ and $\rho_i(G_i)$ should not be thought of as probabilities, but rather as potential functions or factors. The last term in the prior is a binary function that checks that G is consistent with $\prec$, and that $\prec$ is a total order (and hence that G is acyclic). Z is a normalization constant which will cancel out when computing posterior features. By marginalizing over $\prec$, we induce a prior over graphs $p(G)$. We call such an induced prior a modular prior, since it is defined as a product of local terms; this modularity is essential to the efficiency of the algorithms. See (11; 6; 5) for a more detailed discussion of the relationship between priors on orders and graphs.

2.1 Disadvantages of modular priors

Unfortunately the modular prior $p(G)$ is highly non uniform, and in particular, it is not Markov equivalent (9; 6; 11). This can result in incorrect inferences about structure. For example, consider the "cancer network" in Figure 1. Given just observational data, it is only possible to identify the graph up to Markov equivalence. Given a sufficiently large sample, and a uniform graph prior, the posterior should concentrate all its mass on graphs a-d. Hence $p(A \rightarrow B|\text{data}) = 0.75$, since 3 out of the 4 graphs have the edge oriented in this direction. But using the nonunit rho prior, we find that $p(A \rightarrow B|\text{data}) = 0.82$. Many of the other edge probabilities are also wrong (except for the $B \rightarrow D$ and $C \rightarrow D$ edges, whose direction is compelled by the v-structure): see Figure 3.

The root of the problem is that by defining $p(G) = \sum_{\prec} p(G, \prec)$, we favor graphs that are consistent with more orderings. For example, the fully disconnected graph is the most probable under a modular prior. See Figure 2.

Ellis and Wong (5) recognized this problem, and tried to fix it as follows. Let $p^*(G) = \frac{1}{Z}\prod_i \rho_i(G)$ be the desired (modular!) prior, and let $q(G)$ be an approximation gotten by running the MCMC sampler on orders and then sampling a DAG given an order. We can correct for the bias by using an importance sampling weight given by

$$\begin{aligned} w(G) &= \frac{p^*(G)}{q(G)} = \frac{\frac{1}{Z}\prod_i \rho_i(G_i)}{\sum_{\prec} \frac{1}{Z}\prod_i \rho_i(G_i)\text{consistent}(\prec, G)} \\ &= \frac{1}{\#\text{consistent}(G)} \end{aligned}$$

Since computing the number of orders consistent with a graph is #P-complete (1), they approximated this sum using the sampled orders, $w(G) \approx \frac{1}{\sum_{s=1}^{S} \text{consistent}(\prec^s, G)}$. (Note that their samples $\prec^s$ are drawn from the posterior $p(\prec|D)$, which will not result in an unbiased estimate of $\#\text{consistent}(G)$.)

For small problems we can compute this modified prior exactly. We show the result for 3 nodes in Figure 2 for the nonunit rho case, which is the prior used in all previous papers. We see that, although it is uniform amongst members of a Markov equivalence class, it is not uniform across different equivalence classes. If we set $\rho_i(G_i) = 1$, and use the reweighting trick, we can get a globally uniform prior $p(G)$. However, computing the reweighting terms is #P-complete.

2.2 Disadvantages of the DP algorithm

The DP algorithm suffers from the modular prior requirement discussed above. However, it also has additional problems. The DP algorithm returns exact marginal edge probabilities, $p(G_{ij} = 1|D)$, which is useful for visualization, but this output cannot be used for prediction. In particular, it is not possible to assess the quality of the estimated model using predictive likelihood, $p(D'|D) = \sum_G p(D'|G)p(G|D)$. This makes it hard to compare methods in an objective fashion (since the "true" structure is often unknown). In addition, the algorithm takes time and space exponential in d.

2.3 Disadvantages of the MCMC order sampler

The MCMC order sampler suffers from the modular prior requirement discussed above. However, it also has additional problems. In particular, each proposal is expensive to evaluate, because of the need to integrate out over all graphs consistent with an ordering. Also, the output is a sample of orders, from which we must sample graphs, which introduces an extra layer of Monte Carlo variance.

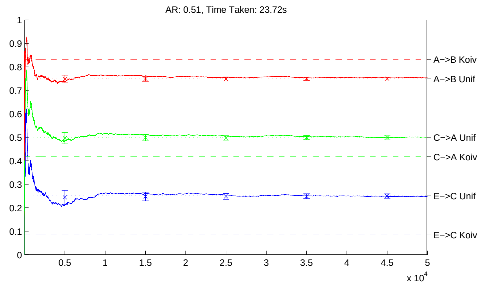

Figure 4: Estimated marginal edge probabilities vs number of MCMC iterations on the cancer network. Dotted lines are the correct answers. Dashed lines are the answers with DP method.

3 Our method

Our method is to perform MCMC in DAG space, but to use the DP algorithm as a proposal distribution (we give the details below). This gets the best of both worlds: it uses dynamic programming to produce a proposal that is close to the target distribution, and that can make large moves through the state space; and it uses Metropolis Hastings to correct for the bias introduced by using modular priors. In addition, since the sampler returns DAGs, it is easy to compute predictive likelihoods and cheaply compute any other feature of interest.

In more detail, our proposal distribution is as follows. It is a mixture of the standard local proposal, that adds, deletes or reverses an edge at random, and a more global proposal that uses the output of the DP algorithm:

$$q(G'|G) = \alpha q_{local}(G'|G) + (1-\alpha) q_{global}(G'|G)$$

Specifically, the global proposal includes an edge between i and j with probability $p_{ij} + p_{ji} \leq 1$, where $p_{ij} = p(G_{ij}|D)$ are the exact marginal posteriors computed using DP. If this edge is included, it is oriented as $i \rightarrow j$ w.p. $p_{ij}/(p_{ij} + p_{ji})$, otherwise it is oriented as $i \leftarrow j$. After sampling each edge pair, we check if the resulting graph is acyclic. (The acyclicity check can be done in amortized constant time using the ancestor matrix trick (8).) If the sampled graph is cyclic, we find all the edges involved in loops, and then, for each loop, we flip the orientation of one edge chosen at random, before proposing the resulting graph. This results in an acceptance rate of about 50% on the 5 node cancer network.

If we set $\alpha = 1$, we get the standard local proposal. If we set $\alpha = 0$, we get a purely global proposal. Note that $q_{global}(G'|G)$ is actually independent of G, so this is an independence sampler; consequently the resulting samples are uncorrelated. We tried various other settings of α (including adapting it according to a fixed schedule), which results in performance somewhere in between purely local and purely global. Below we just show results for $\alpha = 0.1$ for brevity.

4 Results

In Figure 4, we show that our method rapidly converges to the correct answer for the cancer network in Figure 1. (See also Figure 3(f).) We find that the acceptance rate of the global proposal is about 50%, and that generating 50,000 samples only takes about 20 seconds.[1] We have checked that our method gives the right answers on other larger networks, too.

One obvious concern about our method is that the additional cost of running the DP algorithm and processing its output is not worth it, compared to the simpler approach of just using the standard local proposal in DAG space. Another concern is that searching through DAG space is inherently a bad idea, no matter what proposal, compared to sampling in order space. Below we show (experimentally) that both of these concerns are unwarranted.

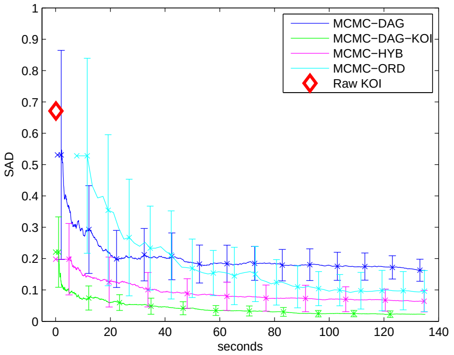

Figure 5: SAD error vs running time on the cancer network for different proposals. The colored lines represent different proposals. From top to bottom (at the end): dark blue is the local proposal, light blue is the order sampler, purple is the $\alpha = 0.1$ hybrid, and green is the global DP sampler. Note that 140 seconds corresponds to about 130,000 samples from the hybrid sampler. The red diamond is the result of the DP algorithm, without using MCMC. Note how fast it is to compute! This figure is best viewed in colour.

In Figure 5, we plot the sum of absolute differences (SAD), $S_t = \sum_{ij} |p(G_{ij} = 1|D) - q_t(G_{ij} = 1|D)|$, versus running time, on the cancer network, where $p(G_{ij} = 1|D)$ are the exact posterior edge marginals (computed using brute force enumeration over the posterior), and $q_t(G_{ij}|D)$ is the approximation based on samples up to time t. We compare 4 MCMC methods: purely local moves through DAG space ($\alpha = 1$), purely global moves through DAG space using the DP proposal ($\alpha = 0$, which is an independence sampler), a mixture of local and global (probability of local move is $\alpha = 0.1$), and an MCMC order sampler (6) with Ellis' importance weighting term.[2] (In our implementation of the order sampler, we took care to implement the various caching schemes described in (6), to ensure a fair comparison.) We also plot the results of the DP algorithm (which takes under one second for this problem). The error bars (representing one standard deviation across 25 chains starting in different conditions) are initially large, because the chains have not burned in.

We see that sampling in order space gives better results than sampling in DAG space (using the standard local proposal), as previously reported in (6). (This is true even after accounting for the fact that order moves are much more expensive than DAG moves.) However, we also see that using a better proposal in DAG space helps even more. Specifically, we see that the best "value for money" (in terms of lowest error per unit time) is to draw 100% of our samples from the global DP-based proposal (which has the added benefit of being an independence sampler).

To see whether the relative orderings would remain the same, we repeated the above experiment on a randomly generated 10 node DAG with binary nodes and random CPTs (with parameters chosen randomly as in (2)). We sampled 4000 cases from this network. With 10 nodes, we cannot compute the exact

[1] Experiments were performed in Matlab on a 3GHz Xeon with 2GB RAM running under linux.

[2] Without the reweighting term, the MCMC order sampler (6) would give the same results (as measured by SAD) as the DP method (11; 10), only much, much slower.

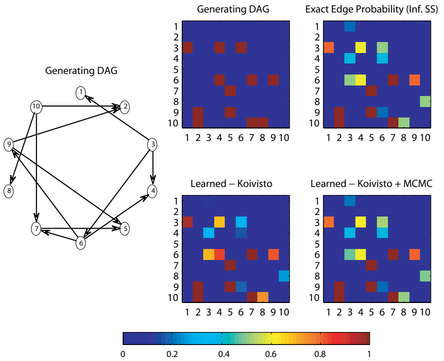

Figure 6: Results for a random 10 node DAG. We show the generating DAG, the asymptotic true posterior marginals, the DP approximation, and the result of our DP+MCMC method.

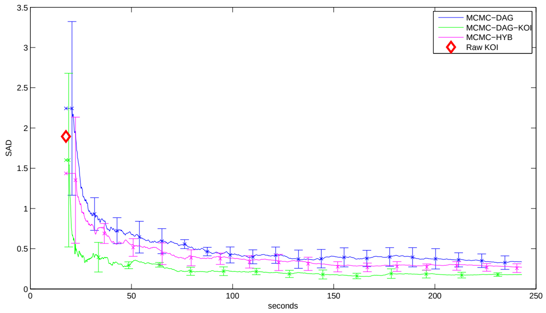

Figure 7: SAD error vs running time on the 10 node network for different proposals.

posterior (since there are $\sim 10^{18}$ DAGs on 10 nodes), so instead we assumed the sample size was large enough that the posterior would concentrate all its mass on the true Markov equivalence class, E. We then assigned $1/|E|$ units of probability mass to all members of this class, and use this to compute $p(G_{ij} = 1|D)$; we call this the asymptotic posterior. (With this definition of $p(G_{ij}|D)$, even the Bayes optimal algorithm may incur non zero error.)

The results are shown in Figure 6 and 7. (We do not yet have results for the order sampler on this 10 node network.) We see that, once again, our DP+MCMC scheme converges on the right answers, and does so more quickly than the other approaches. (The initial 25 second delay, incurred by all algorithms, is the time required to pre-compute all the local marginal likelihoods, $p(X_i|X_{G_i})$.

We have also applied our technique to the biological data set (which includes experimental interventions) in (15), which had previously been analysed using local DAG MCMC in (15), order MCMC in (5), and DP in (4). We found that our DP+MCMC method did not change the results significantly from just using DP (results omitted due to lack of space), but this hybrid method does have the big advantage that it samples full graphs, not marginals, which can then be evaluated in terms of performance in a cross-validation setting (5). (However, we have not yet tried this experiment.)

5 Summary and future work

We pointed out a subtle problem with some of the current (best) approaches for Bayesian structure learning, due to their dependence on modular priors. We then showed how to fix this using Metropolis Hastings. Furthermore, we were able to make this efficient by using a very informative proposal distribution computed using dynamic programming.

In the future we would like to investigate active learning of network structure. Since the posterior may not be factored after having performed an experiment, even if the prior is, it is crucial to be able to handle arbitrary $p(G)$ in the sequential/ active learning setting. The present method will be a suitable enabling technology.

References

[1] G. Brightwell and P. Winkler. Computing linear extensions is #P-complete. In *STOC*. 1991.

[2] D. Chickering and C. Meek. Finding Optimal Bayesian Networks. In *UAI*. 2002.

[3] G. Cooper and C. Yoo. Causal discovery from a mixture of experimental and observational data. In *UAI*. 1999.

[4] D. Eaton and K. Murphy. Exact Bayesian structure learning from uncertain interventions. In *AI/Statistics*. 2007. Submitted.

[5] B. Ellis and W. Wong. Sampling Bayesian Networks quickly. In *Interface*. 2006.

[6] N. Friedman and D. Koller. Being Bayesian about Network Structure: A Bayesian Approach to Structure Discovery in Bayesian Networks. *Machine Learning*, 50:95–126, 2003.

[7] N. Friedman, K. Murphy, et al. Learning the structure of dynamic probabilistic networks. In *UAI*. 1998.

[8] P. Giudici and R. Castelo. Improving Markov chain Monte Carlo model search for data mining. *Machine Learning*, 50(1–2):127 – 158, January 2003.

[9] D. Heckerman, D. Geiger, et al. Learning Bayesian networks: the combination of knowledge and statistical data. *Machine Learning*, 1995.

[10] M. Koivisto. Advances in exact Bayesian structure discovery in Bayesian networks. In *UAI*. 2006.

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

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

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

[14] R. W. Robinson. Counting labeled acyclic digraphs. In F. Harary, ed., *New Directions in the Theory of Graphs*, pp. 239–273. Academic Press, 1973.

[15] K. Sachs, O. Perez, et al. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308, 2005.

[16] P. Spirtes, C. Glymour, et al. *Causation, Prediction, and Search*. MIT Press, 2000. 2nd edition.

[17] J. Tian and J. Pearl. Causal discovery from changes: a Bayesian approach. Tech. rep., UCLA, 2001.