Learning Bayes net structure from sparse data sets

Kevin P. Murphy

9 February 2001

1 Introduction

There are essentially two kinds of approaches for learning the structure of Bayesian Networks (BNs) from data. The first approach tries to find a graph which satisfies all the constraints implied by the empirical conditional independencies measured in the data [PV91, SGS00a, Shi00].[1] The second approach searches through the space of models (either DAGs or PDAGs), and uses some scoring metric (typically Bayesian or some approximation, such as BIC/MDL) to evaluate the models [CH92, Hec95, Hec98, Kra98], typically returning the highest scoring model found.

Our main interest is in learning BN structure from gene expression data [FLNP00, HGJY01, MM99, SGS00b]. In domains such as this, where the ratio of the number of observations to the number of variables is low (i.e., when we have sparse data), selecting a threshold for the conditional independence (CI) tests can be tricky, and repeated use of such tests can lead to inconsistencies [DD99]. Bayesian scoring methods avoid this problem, and can also exploit prior knowledge. However, the most common approach — called Bayesian model selection — only returns a single "best" model, often computed using a greedy local search algorithm. When there is little data, there probably won't be a single "best" model. To reflect our true uncertainty, we need to compute a posterior distribution over models. (We also need such a distribution if we are to design experiments to minimize our structural uncertainty.) This approach is called Bayesian model averaging.

In Bayesian model averaging, we estimate the probability of a feature f (such as the presence of a particular edge) given the data D as follows:

$$P(f|D) = \sum_{G \in \mathcal{G}} P(G|D) f(G)$$

where $\mathcal{G}$ is the set of all the DAGs of a fixed size that we wish to consider. $f(G)$ is 1 if G contains feature f, and is 0 otherwise. For example, $f(G)$ might detect the presence of a certain edge, In this way, we can compute the posterior probability of all possible edges, and display the results by shading the arcs of the fully connected graph to reflect our confidence in each edge.

[1] The output of these constraint-based algorithms is a partially directed acyclic graph (PDAG), and is called a pattern or essential graph. (A PDAG is a chain graph, i.e., a graph which contains directed and undirected edges, but no directed cycles.) The essential graph represents a whole class of Markov equivalent DAGs. Two graphs are Markov equivalent if they imply the same set of (conditional) independencies. For example, $X \to Y \to Z$, $X \leftarrow Y \to Z$ and $X \leftarrow Y \leftarrow Z$ are Markov equivalent, since they all represent $X \perp Z|Y$. In general, two graphs are Markov equivalent iff they have the same structure ignoring arc directions, and the same v-structures [VP90]. (A v-structure consists of converging directed edges into the same node, such as $X \to Y \leftarrow Z$.) We can only distinguish members of the same equivalence class if we have interventional (experimental) data [Pea00, CY99].

$P(G|D)$ is the posterior probability of the graph G, which, by Bayes' rule, is given by

$$P(G|D) = \frac{P(D|G)P(G)}{\sum_{G'} P(D|G')P(G')} \tag{1}$$

$P(G)$ is the prior probability of this graph structure and $P(D|G)$ is the marginal likelihood (also called the evidence):

$$P(D|G) = \int P(D|G,\theta)P(\theta|G)d\theta \tag{2}$$

where θ are the parameters of the model. If there is missing data and/or hidden nodes, we also need to sum over all possible completions of the unobserved data, Z (assuming discrete data for simplicity):

$$P(D|G) = \int \sum_z P(D,z|G,\theta)P(\theta|G)d\theta$$

There are several difficulties with implementing these equations:

- How do we define the priors $P(G)$ and $P(\theta|G)$, and the likelihood $P(x|G,\theta)$?
- How do we integrate over all the parameters?
- How do we marginalize out the hidden nodes?
- How do we sum over all the graphs?

The first three problems also arise in Bayesian model selection; although the [Hec98] tutorial discusses these issues, it does not focus on methods which are suitable for small data sets. The last problem is unique to Bayesian model averaging. Although there have been many general tutorials on BMA (e.g., [HMRV99]), few discuss BNs in any detail. This tutorial aims to fill in these gaps.

2 Priors

2.1 Parameter priors

Following common practice, we will assume global parameter independence:

$$P(\theta|G) = \prod_{i=1}^{n} P(\theta_i|\mathrm{Pa}_G(X_i))$$

where θ_i are the parameters for the Conditional Probability Distribution (CPD) for node i. Hence the likelihood becomes

$$P(x|G,\theta) = \prod_i P(X_i|\mathrm{Pa}_G(X_i),\theta_i)$$

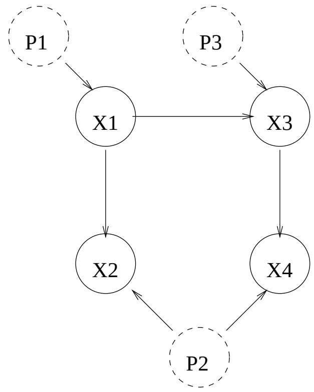

Figure 1: Nodes X_2 and X_4 share the same parameters. (Parameter nodes are shown inside dotted circles.)

If all the data is observed, and the parameter priors are independent, then the parameter posteriors will also be independent [SL90]. This means the marginal likelihood decomposes into a product of terms, one for each node:

$$P(D|G) = \prod_i score(X_i, \mathrm{Pa}_G(X_i)|D)$$

where

$$score(X_i, U|D) = \int \prod_{m=1}^{N} P(x_i[m]|u_i[m], D)P(\theta_i)d\theta_i \tag{3}$$

$x_i[m]$ represents the value of X_i in case m, and $u_i[m]$ the value of its parents; the product over cases follows if we assume i.i.d. training data. Whether this integral is tractable to compute depends on the form the of the CPD and the prior, which we will discuss below.

The most widely used CPD assumes that all nodes are discrete and have unconstrained (multinomial) CPDs, which can therefore be represented as tables. If we use a conjugate (Dirichlet) prior, and all the nodes are observed, and we have global and local parameter independence, we can compute Equation 3 (and hence the marginal likelihood) in closed form: see Appendix A. This still leaves the problem of how to set the hyperparameters (i.e., the parameters of the prior itself) for all possible graph structures in a consistent and robust way. By "consistent" we mean that equivalent models should have equal priors, and by "robust" we mean that tweaking the priors should not radically affect the outcome of model selection. We discuss this both of these issues at length, for the case of Dirichlet distributions, in Appendix A.2.

Although the multinomial is a simple and flexible distribution, and can model combinatorial interactions between parents, it requires a lot of data to estimate: a binary node with k binary parents needs 2^k parameters to specify its CPD (one number for each possible combination of its parents). Genetic networks are believed to have a fan-in of about $k \sim 5$, but even estimating just $2^5 = 32$ parameters is too much if we only have $N \sim 10$ observations per gene.

One possible solution is to use parameter tying (see Figure 1). This encodes the fact that we expect certain kinds of regulatory interactions (e.g., AND gates) to repeat in many different parts of the network. If we are uncertain about which parameters to tie, we can assume they are all drawn from a mixture distribution, c.f., soft weight sharing [NH92, Bis95] in neural networks. Unfortunately, choosing the hyperparameters for such a complex prior may prove a formidable task. In addition, hierarchical priors are hard to integrate out analytically.

A simpler solution is to use more restrictive CPDs, with $O(k)$ parameters. A simple example is logistic regression, which is applicable when all nodes are binary. We define $P(X_i = 1|U_i = u) = \sigma(\sum_j w_{ij} u_j)$, where $\sigma(z) = 1/(1 + e^{-z})$ is the sigmoid (logistic) function, w_{ij} is the weight on the arc from X_j to X_i, and u is a bit-vector representing the parents' values. A closely related model is the noisy-or function [Pea88]. In this case, the child is "on" if any of its parents are on, provided not all the "links" from the on parents are "broken". Define q_{ij} as the probability that the link from X_j to X_i fails. (Failures are assumed to be independent: see [MH97] for ways to lift this restriction.) Then the noisy-or is defined as $P(X_i = 1|U_i = u) = 1 - \prod_j q_{ij}^{u_j}$. If we set $w_{ij} = -\ln q_{ij}$ and $\rho(z) = 1 - e^{-z}$, we can rewrite this as $P(X_i = 1|U_i = u) = \rho(\sum_j w_{ij} u_j)$, which has the same form as the logistic CPD. We can add a dummy parent node that is always on (called a leak node), to represent "all other causes"; this corresponds to adding an offset or bias term, $\rho(b_i + \sum_j w_{ij} u_j)$.

In logistic regression and related models, each parent contributes independently to the effect on the child. Although this cannot model combinatorial interaction, it is a simple model to parameterize. For genetic networks, we often have prior knowledge about the "sign" of a connection, i.e., whether it is excitatory or inhibitory. (This is a special case of a qualitative probabilistic network (QPN) [Wel90].) We can encode this prior knowledge using a diagonal Gaussian prior on the weight vector w_i, with a mean at $+1$ for excitatory links, -1 for inhibitory links, and at 0 for links of unknown sign. [HGJY01] suggest modeling prior knowledge of signs with constrained Dirichlet distributions, but this requires numerical integration. See also [DvdG95, WJ00] for ways of imposing constraints on parameter priors.

A major disadvantage of the logistic (and related) CPD is the inability to compute the marginal likelihood (Equation 2) exactly. One possible (variational) approximation is discussed in [JJ00]; however, its accuracy on small samples has yet to be determined.

A representation for CPDs for discrete nodes which is of variable complexity, ranging from $O(1)$ to $O(2^k)$ parameters, is the tree. [FG96b] show how allowing such "local structure" can enable the learning of denser global graph structures, without overfitting.

For continuous-valued nodes, the most widely used CPD is linear-Gaussian: $P(X_i = x|U_i = u) = N(x; b_i + w_i^T u, \sigma)$. If we use the corresponding conjugate prior (Normal-Wishart), we can compute the marginal likelihood in closed form [HG95]. Unfortunately, assessing this prior is difficult. For non-linear interactions, we can consider generalized linear models (GLIMs) or neural networks.

Even though the data from gene expression arrays is real-valued, we do not anticipate having enough of it to fit (conditional) density functions very accurately, especially non-linear ones. Therefore we propose discretizing (binning) the data, and then using CPDs for discrete nodes. To reduce artifacts, we might be able to perform the discretization process simultaneously with the structure learning c.f. [FG96a].

2.2 Structure priors

For domains in which we have little prior knowledge, it is common to use a uniform prior over possible models. Alternatively, we can impose penalties based on the number of arcs, or the size of families, to discourage networks that are globally or locally "dense"[Bun91].[2] For domains in which we have a lot of prior knowledge, we can construct a prior network, and can penalize deviations from this initial model by

[2]Such a structural penalty is not strictly necessary, since sparse networks will have fewer free parameters, and hence a larger marginal likelihood or Ockham factor [Gul88]. This matches our intuition that we trust a constrained (but correct!) model more than one that can predict anything.

counting the number of mismatched edges [HGC95]. (For example, in [ITR$^+$01], they constructed a network based on databases listing known protein-protein and protein-DNA interactions.) However, most of the time, we have some "fuzzy" prior knowledge. So we now propose a novel approach that lets us model the fact that we may have more confidence in some arcs than others.

Suppose, for simplicity, that our prior beliefs about each edge are independent. Let W_{ij} be the prior probability that there is an edge from X_i to X_j, W_{ji} be the prior that there is an edge from X_j to X_i, and $1-(W_{ij}+W_{ji})$ be the prior that there is no edge between X_i and X_j. If both $W_{ij}>0$ and $W_{ji}>0$, it means we are unsure whether X_i causes X_j or vice versa. (It might be that we believe that if $X_i \rightarrow X_j$ then $X_j \rightarrow X_k$, but if $X_i \leftarrow X_j$ then $X_j \leftarrow X_k$. Unfortunately, we cannot represent such joint constraints with such a simple, factored prior.)

Given an arbitrary DAG G, let $E_{ij}(G)=1$ if G contains the edge $X_i \rightarrow X_j$, $E_{ij}(G)=2$ if G contains the edge $X_i \leftarrow X_j$, and $E_{ij}(G)=3$ if G contains no edge between X_i and X_j. Thus E_{ij} is a multinomial random variable with prior probabilities $\rho_{ij1}=W_{ij}$, $\rho_{ij2}=W_{ji}$, and $\rho_{ij3}=1-(W_{ij}+W_{ji})$. Let the weight of a graph be

$$w(G)=\sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{k=1}^{3}\rho_{ijk}\delta(E_{ij}(G),k)$$

where $\delta(x,y)=1$ if $x=y$ and is 0 otherwise. We define the prior to be the normalized weight: $P(G)\propto w(G)$.

To understand the behavior of this prior, let us first consider the case where all entries in W are either 0 or 1, which means we know the true model. In this case, the weight of a graph is equal to number of edges it gets correct, where "correct" means that it should be absent if the prior says it is absent, and should agree with the prior about the orientation of the edge if it is present. For example, suppose we know the true structure is $X_1 \rightarrow X_2 \rightarrow X_3$, which is reflected in our prior

$$W=\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

There are 25 possible DAGs on 3 variables. The graph $X_1 \rightarrow X_2 \rightarrow X_3$ has the highest weight, which is 3: 1 for the correct $X_1 \rightarrow X_2$, 1 for the correct missing $X_1 \ X_3$, and 1 for the correct $X_2 \rightarrow X_3$.

Now suppose we are uncertain of the orientation of the edge between X_1 and X_2. In addition, we are not sure if there is an edge between X_2 and X_3, and if there is, we are uncertain of its orientation. This can be encoded in the following weight matrix:

$$W=\begin{pmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{3} \\ 0 & \frac{1}{3} & 0 \end{pmatrix}$$

In this case, there are 6 graphs which achieve the maximal weight of $1+\frac{1}{2}+\frac{1}{3}=1.83$; these graphs correspond to the choice of setting $X_1 \rightarrow X_2$ or $X_1 \leftarrow X_2$, combined with the choice of setting $X_2 \rightarrow X_3$, $X_2 \leftarrow X_3$ or $X_2 \ \ X_3$.

3 MCMC methods

Recall that the goal is to compute

$$P(G|D) = \frac{P(D|G)P(G)}{\sum_{G'} P(D|G')P(G')}$$

The normalizing constant $P(D) = \sum_{G'} P(D|G')P(G')$ is intractable to compute, because there a super-exponential number of graphs. To avoid this intractability, we plan to use MCMC (Markov Chain Monte Carlo) techniques to search the very large space of possible models (see e.g., [GRS96] for an introduction to MCMC). Specifically, we plan to use the Metropolis-Hastings (MH) algorithm, which only requires that we be able to compute the posterior odds between the current candidate model, G_1, and the proposed new model, G_2:

$$\frac{P(G_2|D)}{P(G_1|D)} = \frac{P(G_2)}{P(G_1)} \times \frac{P(D|G_2)}{P(D|G_1)} \tag{4}$$

The ratio of the evidences, $\frac{P(D|G_2)}{P(D|G_1)}$, is called the Bayes factor, and is the Bayesian equivalent of the likelihood ratio test.

The idea of applying the MH algorithm to graphical models was first proposed in [MY95], who called the technique MC^3, for MCMC Model Composition. The basic idea is to construct a Markov Chain whose state space is the set of all DAGs and whose stationary distribution is $P(G|D)$. We achieve as follows. Define a transition matrix or kernel, $q(G'|G)$. (The only constraints on q are that the resulting chain should be irreducible and aperiodic.) We sample a new state G' from this proposal distribution, $G' \sim q(\cdot|G)$, and accept this new state with probability $\min\{1, R\}$, where R is the acceptance rate:

$$R = \frac{P(G'|D)}{P(G|D)} \times \frac{q(G|G')}{q(G'|G)}$$

(If the kernel is symmetric, so $q(G|G') = q(G'|G)$, the last term cancels, and the algorithm is called the Metropolis algorithm.) The idea is to sample from this chain for "long enough" to ensure it has reached its stationary distribution (this is called the burn-in time) and throw these samples away; any further samples are then (non-independent) samples from the true posterior, $P(G|D)$, and can be used to estimate many quantities of interest, such as $P(f|D)$. This algorithm is given in pseudo-code in Figure 2. For comparison, we show the pseudo-code for greedy hill-climbing in Figure 3, a popular local search algorithm. In both cases, the initial graph could be chosen using CI-based techniques [SM95, DD99].

The issue of diagnosing convergence of MCMC algorithms is discussed in [GRS96]. The number of samples needed after reaching convergence depends on how rapidly the chain "mixes" (i.e., moves around the posterior distribution). To get a ballbark figure, [GC01] use MC^3 to find a distribution over the 3,781,503 DAGs with 6 binary nodes (of course, many of these are Markov equivalent), using a fully observed dataset with 1,841 cases. They used $T = 100,000$ iterations with no burn-in, but it seemed to converge after "only" 10,000 iterations.

3.1 The proposal distribution

[MY95] suggested the following kernel. Define the neighborhood of the current state, $nbd(G)$, to be the set of DAGs which differ by 1 edge from G, i.e., we can generate $nbd(G)$ by considering all single edge additions,

```
Choose G somehow
While not converged
        Pick a G' u.a.r. from nbd(G)
        Compute R = P(G'|D)q(G|G') / P(G|D)q(G'|G)
        Sample u ~ Unif(0,1)
        If u < min{1, R}
           then G := G'
```

Figure 2: Pseudo-code for the MC^3 algorithm.

```
Choose G somehow
While not converged
        For each G' in nbd(G)
             Compute score(G') = P(G')P(D|G')
        G* := arg max_{G'} score(G')
        If score(G*) > score(G)
             then G := G*
             else converged := true
```

Figure 3: Pseudo-code for hill-climbing.

deletions and reversals, subject to the acyclicity constraint. (A way of quickly checking that the proposed DAG is acyclic (based on the ancestor matrix) was derived in [GC01].) Then let $q(G'|G) = 1/|nbd(G)|$, for $G' \in nbd(G)$, and $q(G'|G) = 0$ for $G' \notin ndb(G)$, so

$$R = \frac{|nbd(G)|P(G')P(D|G')}{|nbd(G')|P(G)P(D|G))}$$

The main advantage of this proposal distribution is that it is efficient to compute R when G and G' only differ by a single edge (assuming complete data): see Section 3.2. This advantage is even more important for greedy search (Figure 3): although we need to know the score of $O(n^2)$ neighbors at each step, only $O(n)$ of these scores change if the steps are one edge at a time.

The single-edge-change proposal can lead to high acceptance rates, but slow mixing, because it takes such small steps through the space. An alternative would be to propose large changes, such as swapping whole substructures, as in genetic programming. If one is not sure which proposal to use, one can always create a mixture distribution. The weights of this mixture are parameters that have to be tuned by hand. [GRG96] suggest that the kernel should be designed so that the average acceptance rate is 0.25.

A natural way to speed up mixing is to reduce the size of the search space. Suppose that, for each node, we restrict the maximum number of parents to be k, instead of n. (This is reasonable, since we expect the fan-in to be small). This reduces the number of parent sets we need to evaluate from $O(2^n)$ to $\binom{n}{k} \leq n^k$. Some heuristics for choosing the set of k potential parents are given in [FNP99]. As long as we give non-zero probability to all possible edge changes in our proposal, we are guaranteed to get the correct answer (since we can get from any graph to any other by single edge changes, and hence the chain is irreducible); heuristics merely help us reach the right answer faster. The heuristics in [FNP99] change with time, since they look for observed dependencies that cannot be explained by the current model. Such adaptive proposal distributions cause no theoretical problems for convergence.

3.2 Computing the marginal likelihood

We now discuss how to efficiently compute the posterior odds (Equation 4) used to decide whether to accept or reject a proposal. Typically, evaluating the ratio of two priors, $P(G_2)/P(G_1)$ is efficient, so the problem reduces to evaluating Bayes factors efficiently. As mentioned in Section 2.1, if the data is complete and we have global parameter independence, the marginal likelihood becomes a product of terms, one per node. Graphs that only differ by a single link have marginal likelihoods that differ by at most two terms, since all the others cancel. For example, let G_1 be the chain $X_1 \to X_2 \to X_3 \to X_4$, and G_2 be the same but with the middle arc reversed: $X_1 \to X_2 \leftarrow X_3 \to X_4$. Then

$$\begin{aligned} \frac{P(D|G_2)}{P(D|G_1)} &= \frac{score(X_1)score(X_1, X_2, X_3)score(X_3)score(X_3, X_4)}{score(X_1)score(X_1, X_2)score(X_2, X_3)score(X_3, X_4)} \\ &= \frac{score(X_1, X_2, X_3)score(X_3)}{score(X_1, X_2)score(X_2, X_3)} \end{aligned}$$

where score was defined in Equation 3. In general, if an arc to X_i is added or deleted, only $score(X_i|\Pi_i)$ needs to be evaluated; if an arc between X_i and X_j is reversed, only $score(X_i|\Pi_i)$ and $score(X_j|\Pi_j)$ need to be evaluated.

3.3 Partial observability

3.3.1 Approximating the marginal likeihood

When we have partial observability, we cannot compute the marginal likelihood in closed form, even for discrete variables with conjugate priors, because the parameter posteriors are no longer independent. (Marginalizing over the hidden variables induces a mixture distribution.) The most straightforward approach is to approximate the marginal likelihoods directly, and then take ratios.

One accurate way of approximating the marginal likelihood, known as the Candidate's method [Chi95, Raf96], is as follows. We pick an arbitrary value θ_G^* (e.g., the MAP value), and compute

$$P(D|G) = \frac{P(D|\theta_G^*, G)P(\theta_G^*|G)}{P(\theta_G^*|D, G)}$$

Computing $P(\theta_G^*|G)$ is trivial, and computing $P(D|\theta_G^*, G)$ can be done using any BN inference algorithm. The denominator can be approximated using Gibbs sampling: see [CH97] for details. (See also [SNR00] for a related approach, based on the harmonic mean estimator.)

Various large sample (e.g., Laplace) approximations to the marginal likelihood, which are computationally cheaper, can also be used. These are compared in [CH97]. The conclusion is that the Cheeseman-Stutz approximation, which is a variation of BIC, is the most accurate, at least in the case of naive-Bayes (mixture) models. (It often even beats the computationally more expensive Laplace approximation.) The cost of this approximation is equal to the cost of running EM to find the MAP value (c.f., the variational Bayes technique [Att00, Min00b]). However, the accuracy of these techniques for small samples is suspect. Even for large samples, these approximations are inaccurate at estimating $P(D|G_1)/P(D|G_2)$, where G_1 is the most probable model, and G_2 is the second most probable. That is, these approximations can be useful for model selection, but less so for model averaging.

3.3.2 Data augmentation

An alternative approach is to extend the state-space of the Markov chain, so that it not only searches over models, but also over the values of the unobserved nodes [YMHL95]. (To simplify the computation, we will also add the parameters to the state space, although they do not need to be sampled.) The idea is to use Gibbs sampling, where we alternate between sampling a new model given the current completed data set, and sampling a completed data set given the current model. (This is basically an extension of the IP algorithm for data augmentation [TW87]).

At a high level, the algorithm cycles through the following steps (where Y is the observed data and Z is the hidden data):

1. Sample $G_{t+1} \sim P(G|Y, Z_t) \propto P(G)P(Y, Z_t|G)$
2. Compute $P(\theta_{t+1}|Y, Z_t, G_{t+1})$
3. Sample $Z_{t+1} \sim P(Z|Y, G_{t+1}, \overline{\theta}_{t+1})$

To avoid computing the normalizing constant implicit in the first step, we use the MH algorithm, and approximate the Bayes factor by using the current completed dataset. This is an approximation to the expected complete-data Bayes factor [Bun94, Raf96]:

$$\frac{P(Y|G')}{P(Y|G_t)} = E\left[\frac{P(Y, Z|G')}{P(Y, Z|G_t)} \mid Y, G_t\right] \approx \frac{P(Y, Z_t|G')}{P(Y, Z_t|G_t)} \tag{5}$$

The second step can be performed in closed form for conjugate distributions (see Appendix A). Finally, we sample from the predictive distribution, $P(Z|Y, G_{t+1})$, by using the mean parameter values, $\overline{\theta}_{t+1}$ (see Appendix A), with Gibbs sampling. The overall algorithm is sketched in Figure 4.

In principle, if the chain is ergodic, the starting point, (G_1, Z_1), does not matter, but in practice, good initialisation is important for fast convergence. We can choose the initial Z_1 by sampling from $P(Z|G_1, Y)$ using Gibbs sampling, or by computing the MAP estimate, $\hat{\theta}_{MAP} = \arg\max_\theta P(\theta|G_1)P(Y|\theta, G_1)$, and then using Gibbs sampling on $P(Z|G_1, Y, \hat{\theta}_{MAP})$. It is difficult to choose the initial graph structure, G_1. We cannot use CI-based methods [SGS00a] because we have latent variables. On the other hand, we cannot choose, say, the empty graph, because our initial estimate of Z_1 will be poor. We therefore assume we have some good initial guess based on domain knowledge.

3.3.3 Structural EM

A deterministic approximation to the data augmentation scheme, called Structural EM, was proposed in [Fri97]. The basic idea is to compute the expected complete-data marginal likelihood, $P(Y|G') \approx E_Z\left[P(Y, Z|G')|G, Y, \hat{\theta}_G\right]$, using a BN inference algorithm applied to the current model G and its current

Choose G_1, Z_1 somehow
for $t = 1, 2, \ldots$
    Pick a G' u.a.r. from $nbd(G_t)$
    Compute $\hat{B} = \frac{P(Y,Z_t|G')}{P(Y,Z_t|G_t)}$
    Compute $R = \frac{|nbd(G_t)|P(G')}{|nbd(G')|P(G_t)}\hat{B}$
    Sample $u \sim \text{Unif}(0,1)$
    If $u < \min\{1, R\}$
        then $G_{t+1} = G'$
            Compute $P(\theta_{t+1}|Y, Z_t, G_{t+1})$ using Bayesian updating
            Compute $\overline{\theta}_{t+1}$ from $P(\theta_{t+1}|Y, Z_t, G_{t+1})$
            Sample $Z_{t+1} \sim P(Z|Y, G_{t+1}, \overline{\theta}_{t+1})$ using Gibbs sampling

Figure 4: Pseudo-code for the MC^3 algorithm modified to handle missing data.

Choose G somehow
While not converged
    For each G' in $nbd(G)$
        Compute $\overline{N}^{G'}$ in Equation 6 using an inference algorithm [E step]
        Compute $score(G') = P(G')P(\overline{N}^{G'}|G', \hat{\theta}(\overline{N}^{G'}))$
    $G^* := \arg\max_{G'} score(G')$
    If $score(G^*) > score(G)$
        then $G := G^*$ [structural M step]
            $\hat{\theta}_G := \arg\max_\theta P(\overline{N}^G|G, \theta)P(\theta|G)$ [parametric M step]
        else converged := true

Figure 5: Pseudo-code for Structural EM.

MAP parameters, $\hat{\theta}_G$. In the case of multinomials (see Appendix A), we can compute the expected sufficient statistics for each family in the candidate model G' as follows:[3]

$$\overline{N}^{G'}_{ijk} = \sum_{m=1}^{N} P(X_i = k, \text{Pa}_{G'}(X_i) = j|y(m), G, \hat{\theta}_G) \tag{6}$$

SEM was designed for model selection, as opposed to model averaging, and hence the above formula was embedded inside of the hill-climbing algorithm in Figure 3; the resulting algorithm is shown in Figure 5. A "Bayesian" version of this was proposed in [Fri98]. In this case, instead of using the MAP value $\hat{\theta}_G$, he approximately integrates over all θ.

[3]Note that G' might have families that are not in G. If we are using the junction tree algorithm [Jen96] for inference, this means we may have to compute joint probability distributions on sets of nodes that are not in any clique, which can be slow.

3.4 Searching over orderings

[FK00] claim that MCMC over structures does not mix well for large models (more than 10 variables, say). Instead they use MCMC to search over variable *orderings*. Given a total ordering $\prec$, the likelihood decomposes into a product of terms, one per family, since the parents for each node can be chosen independently (there is no global acyclicity constraint). The following equation was first noted in [Bun91]:

$$\begin{aligned} P(D|\prec) &= \sum_{G \in \mathcal{G}_\prec} \prod_{i=1}^{n} score(X_i, Pa_G(X_i)|D) \\ &= \prod_i \sum_{U \in \mathcal{U}_{\prec,i}} score(X_i, U|D) \end{aligned} \tag{7}$$

$\mathcal{G}_\prec$ is the set of graphs consistent with the ordering $\prec$, and $\mathcal{U}_{\prec,i}$ is the set of legal parents for node i consistent with $\prec$. If we bound the fan-in (number of parents) by k, each summation in Equation 7 takes $\binom{n}{k} \leq n^k$ time to compute, so the whole equation takes $O(n^{k+1})$ time. This quantity must then be averaged over orderings produced by the Markov chain.

[FK00] claim that the posterior "landscape" over orderings is "smoother" than that over models, since small changes in an ordering do not affect the score as much as small changes (such as an edge reversal) in a model. While this is probably true (not least because the space of orderings is "only" of size $n!$, whereas the space of DAGs in $O(2^{n^2})$), there are some disadvantages to their approach. Firstly, it is much more natural to specify a prior over model structures than over variable orderings.[4] Secondly, it seems harder to extend their technique to the missing data case. By contrast, searching directly in model space allows us to apply a whole suite of well known approximations.

3.5 Searching over essential graphs

Since we cannot distinguish members of the same Markov equivalence class if we only have observational data, it makes sense to search in the space of essential graphs, which is much smaller than the space of all DAGs. (Constraint-based algorithms [SGS00a] only work with essential graphs.) The Bayesian scoring metric can only be applied to DAGs; hence one has to convert the PDAG to a DAG to evaluate it [Chi96]. MCMC methods for finding high scoring PDAGs are discussed in [MAPV95]. Hybrid methods, that start with constraint-based methods and then switch to greedy search using a Bayesian evaluation metric, are discussed in [SV93, SM95, DD99]. None of these methods are applicable if we have experimental (interventional) as well as observational data.

3.6 Reversible jump MCMC

We mention, just for completeness, the reversible jump MCMC algorithm [Gre98]. This is necessary when the state space has variable dimension, as occurs when estimating parameters as well as model structure (since the number of parameters varies with the structure). For an application of RJMCMC to graphical models, see [GGT00]. By using data augmentation, we are able to integrate out the parameters, and hence avoid the complexities of RJMCMC.

[4] However, interventions give us some hints about the ordering. For example, if we knockout gene X_1, and notice that genes X_2 and X_3 change from their "wildtype" state, but genes X_4 and X_5 do not, it suggests that X_1 is the ancestor of X_2 and X_3. This heuristic, together with the set covering algorithm, was used to learn boolean networks from interventional data [ITK00].

A Multinomial distributions and Dirichlet priors

For discrete nodes, it is very common to assume the local CPDs are multinomial, i.e., represented as a table of the form $\Pr(X_i = k|\Pi_i = j) = \theta_{ijk}$, for $k = 1, \ldots, r_i$ and $j = 1, \ldots, q_i$, where r_i is the number of values node i can take on, and $q_i = \prod_{l \in \Pi_i} r_l$ is the number of values node i's parents, Π_i, can take on. These parameters satisfy the constraints $0 \leq \theta_{ijk} \leq 1$ and $\sum_k \theta_{ijk} = 1$.

Following common practice, we will make two assumptions. First, global parameter independence: $P(\theta) = \prod_{i=1}^n P(\theta_i)$, where $\theta_i = \{\theta_{ijk}, j = 1, \ldots, q_i, k = 1, \ldots, r_i\}$ are the parameters for node i. Second, local parameter independence: $P(\theta_i) = \prod_{j=1}^{q_i} \theta_{ij}$, where $\theta_{ij} = \{\theta_{ijk}, k = 1, \ldots, r_i\}$ are the parameters for the j'th row of X_i's table (i.e., parameters for the j'th instantiation of X_i's parents). Given a factored prior and complete data, the posterior over parameters will also be factored [SL90]. We will give the form of this posterior below. (Note that, if we have missing data, the parameter posterior will no longer be factored. Hence assuming parameter independence is equivalent to assuming that one's prior knowledge was derived from a fully obsered "virtual database".)

[GH97, RG01] prove that the assumptions of global and local parameter independence, plus an additional assumption called likelihood equivalence[5], imply that the prior must be Dirichlet. Fortunately, the Dirichlet prior is the conjugate prior for the multinomial [Ber85], which makes analysis easier, as we will see below. (For this reason, the Dirichlet is often used even if the assumption of likelihood equivalence is violated.) Note that, in the case of binary nodes, the multinomial becomes the Bernoulli distribution, and the Dirichlet becomes the Beta.

Given global and local independence, each CPD $P(X_i|U_i = j) = \theta_{ij}$ is a multinomial random variable with r_i possible values. The Dirichlet prior, $\theta_{ij} \sim \mathcal{D}(\alpha_{ij1}, \ldots, \alpha_{i,j,r_i})$, is defined as

$$P(\theta_{ij}|\alpha_{ij}) = \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}-1} \times \frac{1}{B(\alpha_{ij1}, \ldots, \alpha_{i,j,r_i})}$$

The normalizing constant is the r_i-dimensional Beta function

$$B(\alpha_1, \ldots, \alpha_r) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_{k=1}^r \Gamma(\alpha_k)}$$

where $\Gamma(\cdot)$ is the gamma function; for positive integers, $\Gamma(n) = (n-1)!$.

The hyperparameters, $\alpha_{ijk} > 0$, have a simple interpretation as pseudo counts. The quantity $\alpha_{ijk} - 1$ represents the number of imaginary cases in which event $(X_i = k, \Pi_i = j)$ has already occured (in some virtual prior database). Upon seeing a database D in which the event $(X_i = k, \Pi_i = j)$ occurs N_{ijk} times, the parameter posterior becomes

$$\theta_{ij}|D \sim \mathcal{D}(\alpha_{ij1} + N_{ij1}, \ldots, \alpha_{i,j,r_i} + N_{i,j,r_i})$$

The posterior mean is

$$E[\theta_{ijk}|D] = \frac{\alpha_{ijk} + N_{ijk}}{\sum_{l=1}^{r_i} \alpha_{ijl} + N_{ijl}} \tag{8}$$

[5]Two graph structures are likelihood equivalent if they assign the same marginal likelihood to data, i.e., $P(D|G_1) = P(D|G_2)$. This is weaker than the assumption of Markov (hypothesis) equivalence, which says two graphs are equivalent if they encode the same set of conditional independence assumptions. Hypothesis equivalence is clearly violated if we adopt a causal interpretation of BNs. In addition, likelihood equivalence is violated if we have interventional data. See [Hec95] for a discussion.

and the posterior mode (MAP estimate) is

$$\arg\max P[\theta_{ijk}|D] = \frac{\alpha_{ijk} + N_{ijk} - 1}{\sum_{l=1}^{r_i} \alpha_{ijl} + N_{ijl} - r_i} \tag{9}$$

A.1 Computing the marginal likelihood

The predictive distribution is just

$$\begin{aligned} P(x|\theta) &= \prod_{i=1}^{n}\prod_{j=1}^{q_i}\int\prod_{k=1}^{r_i}\theta_{ijk}^{1_{ijk}(x)}P(\theta_{ijk})d\theta_{ijk} \\ &= \prod_{i=1}^{n}\prod_{j=1}^{q_i}\prod_{k=1}^{r_i}E(\theta_{ijk})^{1_{ijk}(x)} \end{aligned} \tag{10}$$

where $1_{ijk}(x)$ is an indicator function that is 1 if the event $(X_i = k, \Pi_i = j)$ occurs in case x, and is 0 otherwise.

To compute the marginal likelihood for a database of N cases, $D = (x^1, \ldots, x^N)$, we can use sequential Bayesian updating [SL90]:

$$\begin{aligned} P(D) &= P(x^1|\theta_0)P(x^2|\theta_0, x^1)\ldots P(x^N|\theta_0, x^{1:N-1}) \\ &= P(x^1|\theta_0)P(x^2|\theta_1)\ldots P(x^N|\theta_{N-1}) \end{aligned}$$

where $\theta_0 = \alpha$ is our prior, and θ_t is the result of updating θ_{t-1} with x^t.

To compute this in batch form, we simply compute the posterior means of the parameters (Equation 8), and plug these expected values into the sample likelihood equation:

$$P(D) = P(D|\overline{\theta}) = \prod_{i=1}^{n}\prod_{j=1}^{q_i}\prod_{k=1}^{r_i}\overline{\theta}_{ijk}^{N_{ijk}} \tag{11}$$

Alternatively, this can be written as follows [CH92]

$$\begin{aligned} P(D) &= \prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{B(\alpha_{ij1} + N_{ij1}, \ldots, \alpha_{i,j,r_i} + N_{i,j,r_i})}{B(\alpha_{ij1}, \ldots, \alpha_{i,j,r_i})} \\ &= \prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})}\cdot\prod_{k=1}^{r_i}\frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \end{aligned} \tag{12}$$

For interventional data, Equation 12 is modified by defining N_{ijk} to be the number of times $X_i = k$ is *passively observed* in the context $\Pi_i = j$, as shown by [CY99]. (The intuition is that *setting* $X_i = k$ does not tell us anything about how likely this event is to occur "by chance", and hence should not be counted). Hence, in addition to D, we need to keep a record of which variables were clamped (if any) in each data case.

All of the above equations are conditioned on a specific graph structure, G. Making this explicit, we can write the (unnormalized) posterior of a graph as $P(G,D) = P(G)P(D|G)$, where $P(D|G)$ is given by Equation 12. In [HGC95], this is called the BD (Bayesian Dirichlet) metric.

A.2 Assessing Dirichlet priors

It is clearly impossible for the user to specify parametric priors for $O(2^{n^2})$ graph structures. [HGC95] show that, under some assumptions[6], it is possible to derive the Dirichlet parameters for an arbitrary graph from a single prior network, plus a confidence factor, α. Specifically, $\alpha_{ijk} = \alpha P(X_i = k, \Pi_i = j|G_c)$, where G_c is the complete graph. When priors are derived in this manner, the BD score is called BDe (BD with likelihood equivalence). Unfortunately, it might be difficult to parameterize the prior network (especially because of the counterintuitive conditioning on G_c: see [HGC95] for a discussion). In addition, computing the parameter priors for an arbitrary graph structure from such a prior network requires running an inference algorithm, which can be slow. [SDLC93] suggest a similar way of computing Dirichlet priors from a prior network.

A much simpler alternative is to use a non-informative prior. A natural choice is $\alpha_{ijk} = 0$, which corresponds to maximum likelihood. (In the binary case, this is called Haldane's prior.) However, this is an improper prior. More importantly, this will cause the log-likelihood to explode if a future case contains an event that was not seen in the training data.

If we set $0 < \alpha_{ijk} < 1$, we encourage the parameter values θ_{ijk} to be near 0 or 1, thus encoding near-deterministic distributions. This might be desirable in some domains. [Bra99] explicitely encodes this bias using an "entropic prior" of the form

$$P(\theta_{ij}) \propto e^{-H(\theta_{ij})} = \prod_k \theta_{ijk}^{\theta_{ijk}}.$$

Unfortunately, the entropic prior is not a conjugate prior.

[CH92] suggest the uniform prior, $\alpha_{ijk} = 1$. This is a non-informative prior since it does not affect the posterior (although it does affect the marginal likelihood). Unfortunately, it is not entirely uninformative, because it is not transformation invariant. The fully non-informative prior is called a Jeffrey's prior. For the special case of a binary root node (i.e., a node with no parents and a beta distribution), the Jeffrey's prior is just $\alpha_{ijk} = \frac{1}{2}$. Computing a Jeffrey's prior for an arbitrary BN is hard: see [KMS$^+$98].

[Bun91] suggests the prior $\alpha_{ijk} = \alpha/(r_i q_i)$, where α is an equivalent sample size. This induces the following distribution

$$P(X_i = k|\Pi_i = j) = E[\theta_{ijk}] = \frac{\alpha/(r_i q_i)}{\alpha/q_i} = \frac{1}{r_i}$$

This is a special case of the BDe metric where the prior network assigns a uniform distribution to the joint distribution; hence [HGC95] call this the BDeu metric.

A more sophisticated approach is to use a hierarchical prior, where we place a prior on the hyperparameters themselves. For example, let $\alpha_{ijk} = \alpha_{ij0}/r_i$ for each i, j, where α_{ij0} is the prior precision for α_{ij}. α_{ij0} is itself an r.v., and can be given e.g., a gamma distribution. Unfortunately, we can no longer compute

[6] The assumptions are global and local parameter independence, likelihood equivalence, parameter modularity and structural possibility. Parameter modularity says that $P(\theta_{ij})$ is the same for any two graph structures in which X_i has the same set of parents. Structural possibility says that all complete (fully connected) graph structures are possible a priori.

the marginal likelihood in closed form using such hierarchical priors, and must resort to sampling, as in [GGT00, DF99]. A more efficient method, known as empirical Bayes or maximum likelihood type II, is to estimate the hyperparameters from data: see [Min00a] for details.

A.3 Other CPDs in the exponential family

Many CPDs in the exponential family (e.g., multinomial, Gaussian) can be given a conjugate prior (Dirichlet, Normal-Wishart), for which the corresponding posterior and marginal likelihood can be computed in closed-form (provided we have complete data). See [BS94, Bun94] for a discussion of the general case, and [GH94] for a discussion of the Gaussian case.

References

[Att00] H. Attias. A variational Bayesian framework for graphical models. In *NIPS-12*, 2000.

[Ber85] J. Berger. *Statistical Decision Theory and Bayesian Analysis.* Springer-Verlag, 1985.

[Bis95] C. M. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, 1995.

[Bra99] M. Brand. Structure learning in conditional probability models via an entropic prior and parameter extinction. *Neural Computation*, 11:1155–1182, 1999.

[BS94] J. Bernardo and A. Smith. *Bayesian Theory.* John Wiley, 1994.

[Bun91] W. Buntine. Theory refinemement on Bayesian networks. In *UAI*, 1991.

[Bun94] W. L. Buntine. Operations for learning with graphical models. *J. of AI Research*, pages 159–225, 1994.

[CH92] G. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

[CH97] D. Chickering and D. Heckerman. Efficient approximations for the marginal likelihood of incomplete data given a bayesian network. *Machine Learning*, 29:181–212, 1997.

[Chi95] S. Chib. Marginal likelihood from the Gibbs output. *JASA*, 90:1313–1321, 1995.

[Chi96] M. Chickering. Learning equivalence classes of Bayesian network structures. In *UAI*, 1996.

[CY99] G. Cooper and C. Yoo. Causal discovery from a mixture of experimental and observational data. In *UAI*, 1999.

[DD99] D. Dash and M. Druzdel. A hybrid anytime algoruhtm for the construction of causal models from sparse data. In *UAI*, 1999.

[DF99] P. Dellaportas and J. Forster. Markov chain Monte Carlo model determination for hierarchical and graphical log-linear models. *Biometrika*, 1999. To appear.

[DvdG95] M. Druzdel and L. van der Gaag. Elicitation of probabilities for belief networks: Combining qualitative and quantitative information. In *UAI*, 1995.

[FG96a] N. Friedman and M. Goldszmidt. Discretizing continuous attributes while learning Bayesian networks. In *Intl. Conf. on Machine Learning*, 1996.

[FG96b] N. Friedman and M. Goldszmidt. Learning Bayesian networks with local structure. In *UAI*, 1996.

[FK00] N. Friedman and D. Koller. Being Bayesian about network structure. In *UAI*, 2000.

[FLNP00] N. Friedman, M. Linearl, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. *J. Computational Biology*, 2000.

[FNP99] N. Friedman, I. Nachman, and D. Peer. Learning Bayesian network structure from massive datasets: The "sparse candidate" algorithm. In *UAI*, 1999.

[Fri97] N. Friedman. Learning Bayesian networks in the presence of missing values and hidden variables. In *UAI*, 1997.

[Fri98] N. Friedman. The Bayesian structural EM algorithm. In *UAI*, 1998.

[GC01] P. Giudici and R. Castelo. Improving Markov chain Monte Carlo model search for data mining. *Machine Learning*, 2001. To appear.

[GGT00] P. Giudici, P. Green, and C. Tarantola. Efficient model determination for discrete graphical models. *Biometrika*, 2000. To appear.

[GH94] D. Geiger and D. Heckerman. Learning Gaussian networks. In *UAI*, volume 10, pages 235–243, 1994.

[GH97] D. Geiger and D. Heckerman. A characterization of Dirchlet distributions through local and global independence. *Annals of Statistics*, 25:1344–1368, 1997.

[Gre98] P. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732, 1998.

[GRG96] A. Gelman, G. Roberts, and W. Gilks. Efficient Metropolis jumping rules. In J. Bernardo, J. Berger, A. Dawid, and A. Smith, editors, *Bayesian Statistics 5*. Oxford, 1996.

[GRS96] W. Gilks, S. Richardson, and D. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, 1996.

[Gul88] S. Gull. Bayesian inductive inference and maximum entropy. In G. Erickson and C. Smith, editors, *Maximum Entropy and Bayesian Methods in Science and Engineering, Volume 1: Foundations*, pages 53–74. Dordrecht, 1988.

[Hec95] D. Hecerkman. A Bayesian approach to learning causal networks. In *UAI*, 1995.

[Hec98] D. Heckerman. A tutorial on learnign with Bayesian networks. In M. Jordan, editor, *Learning in Graphical Models*. MIT Press, 1998.

[HG95] D. Heckerman and D. Geiger. Learning Bayesian networks: a unification for discrete and Gaussian domains. In *UAI*, volume 11, pages 274–284, 1995.

[HGC95] D. Heckerman, D. Geiger, and M. Chickering. Learning Bayesian networks: the combination of knowledge and statistical data. *Machine Learning*, 1995.

[HGJY01] A. Hartemink, D. Gifford, T. Jaakkola, and R. Young. Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. In *Proc. of the Pacific Symp. on Biocomputing*, 2001.

[HMRV99] J. Hoeting, D. Madigan, A. Raftery, and C. Volinsky. Bayesian model averaging: A tutorial. *Statistical Science*, 4(4), 1999.

[ITK00] T. Ideker, V. Thorsson, and R. Karp. Discovery of regulatory interactions through perturbation: inference and experimental design. In *Proc. of the Pacific Symp. on Biocomputing*, 2000.

[ITR+01] T. Ideker, V. Thorsson, J. Ranish, R. Christmas, J. Buhler, R. Bumgarner, R. Aebersold, and L. Hood. Integrated genomic and proteomic analysis of a systematically perturned metabolic network. *Science*, 2001. Submitted.

[Jen96] F. V. Jensen. *An Introduction to Bayesian Networks*. UCL Press, London, England, 1996.

[JJ00] T. S. Jaakkola and M. I. Jordan. Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10:25–37, 2000.

[KMS+98] P. Kontkanen, P. Mullymäki, T. Silander, H. Tirri, and P. Grünwald. A comparison of non-informative priors for Bayesian networks. In *The Yearbook of the Finnish Statistical Society 1997*, pages 53–62. ?, 1998.

[Kra98] P. Krause. Learning probabilistic networks. Technical report, Philips Research Labs, Redhill, England, 1998.

[MAPV95] D. Madigan, S. Anderson, M. Perlman, and C. Volinsky. Bayesian model averaging and model selection for Markov equivalence classes of acyclic graphs. Technical report, Univ. Washington, 1995.

[MH97] C. Meek and D. Heckerman. Structure and parameter learning for causal independence and causal interaction models. In *UAI*, pages 366–375, 1997.

[Min00a] T. Minka. Estimating a Dirichlet distribution. Technical report, MIT, 2000.

[Min00b] T. Minka. Variational Bayes for mixture models: Reversing EM. Technical report, MIT, 2000.

[MM99] K. Murphy and S. Mian. Modelling gene expression data using dynamic Bayesian networks. Technical report, U.C. Berkeley, Dept. Comp. Sci., 1999.

[MY95] D. Madigan and J. York. Bayesian graphical models for discrete data. *Intl. Statistical Review*, 63:215–232, 1995.

[NH92] S. Nowlan and G. Hinton. Simplifying neural networks by soft weight sharing. *Neural Computation*, 4(4):473–493, 1992.

[Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[Pea00] J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge Univ. Press, 2000.

[PV91] J. Pearl and T. Verma. A theory of inferred causation. In *Knowledge Representation*, pages 441–452, 1991.

[Raf96] A. Raftery. Hypothesis testing and model selection via posterior simulation. In *Markov Chain Monte Carlo in Practice*. Chapman and Hall, 1996.

[RG01] D. Rusakov and D. Geiger. On the parameter priors for discrete DAG models. In *AI/Stats*, 2001.

[SDLC93] David J. A. Spiegelhalter, Philip Dawid, Steffen L. Lauritzen, and Robert G. Cowell. Bayesian analysis in expert systems. *Statistical Science*, 8(3):219–283, 1993.

[SGS00a] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT Press, 2000. 2nd edition.

[SGS00b] P. Spirtes, C. Glymour, and R. Scheines. Constructing Bayesian network models of gene expression networks from microarray data. In *Proc. of the Atlantic Symposium on Computational Biology, Genome Information Systems & Technology*, 2000.

[Shi00] B. Shipley. *Cause and Correlation in Biology: A User's Guide to Path Analysis, Structural Equations and Causal Inference.* Cambridge, 2000.

[SL90] D. J. Spiegelhalter and S. L. Lauritzen. Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20, 1990.

[SM95] P. Spirtes and C. Meek. Learning Baeysian networks with discrete variables from data. In *Proc. of the the Conf. on Knowledge Discovery and Data Mining*, 1995.

[SNR00] J. Satagopan, M. Newton, and A. Raftery. Easy estimation of normalizing constants and bayes factors from posterior simulation: Stabilizing the harmonic mean estimator. Technical report, U. Washington, 2000.

[SV93] M. Singh and M. Valtorta. An algorithm for the construction of Bayesian network structures from data. In *UAI*, 1993.

[TW87] M. Tanner and W. Wong. The calculation of posterior distributions by data augmentation. *JASA*, 82(398):528–540, 1987.

[VP90] T. Verma and J. Pearl. Equivalence and synthesis of causal models. In *UAI*, 1990.

[Wel90] M. P. Wellman. Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, 44(3):257–303, 1990.

[WJ00] F. Wittig and A. Jameson. Exploiting qualitative knowledge in the learning of conditional probabilities of bayesian networks. In *UAI*, 2000.

[YMHL95] J. York, D. Madigan, I. Heuch, and R. Lie. Birth defects registered by double sampling: a Bayesian approach incorporating covariates and model uncertainty. *Applied Statistics*, 44(2):227–242, 1995.