Graphical Models: Structure Learning

David Heckerman
Microsoft Research
One Microsoft Way, Redmond, WA 98052
heckerma@microsoft.com

INTRODUCTION

The article GRAPHICAL MODELS: PARAMETER LEARNING discussed the learning of parameters for a fixed graphical model. In this article, we discuss the simultaneous learning of parameters and structure. Real-world applications of such learning abound and can be found in (e.g.) the Proceedings of the Conference on Uncertainty in Artificial Intelligence (1991 and after). An index to software for parameter and structure learning can be found at http://www.cs.berkeley.edu/ murphyk/Bayes/bnsoft.html.

For simplicity, we concentrate on directed-acyclic graphical models (DAG models), but the basic principles described here can be applied more generally. We describe the Bayesian approach in detail and mention several common non-Bayesian approaches.

We use the same notation as the article on parameter learning. In particular, we use $\mathbf{X} = (X_1, \ldots, X_n)$ to denote the n variables that we are modeling, $\mathbf{x}$ to denote a configuration or observation of $\mathbf{X}$, $d = (\mathbf{x}^1, \ldots, \mathbf{x}^N)$ to denote a random sample of N observations of $\mathbf{X}$. In addition, we use $\mathbf{Pa}_i$ to denote the variables corresponding to the parents of X_i in a DAG model and $\mathbf{pa}_i$ to denote a configuration of those variables. Finally, we shall use the terms "model" and "structure" interchangeably. In particular, a DAG model (and hence its structure) is described by (1) its nodes and arcs, and (2) the distribution class of each of its local distributions $p(x_i|\mathbf{pa}_i)$.

THE BAYESIAN APPROACH

When we learn a model and its parameters, we presumably are uncertain about their identity. When following the Bayesian approach—in which all uncertainty is encoded as (subjective) probability—we encode this uncertainty as prior distributions over random variables corresponding to structure and parameters. In particular, let $\mathbf{m}$ be a random variable having states $\mathbf{m}^1, \ldots, \mathbf{m}^M$ corresponding to the possible models. (Note that we are assuming the models are mutually exclusive). In addition, let $\boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^M$ be random variables corresponding the unknown parameters of each of the M possible models. Then we express our uncertainty prior to learning as the prior distributions $p(\mathbf{m})$, and $p(\boldsymbol{\theta}^1), \ldots, p(\boldsymbol{\theta}^M)$.

Given data $\mathbf{d}$, a random sample from the true but unknown joint distribution for $\mathbf{d}$, we compute the posterior distributions for each $\mathbf{m}$ and $\boldsymbol{\theta}^m$ using Bayes' rule:

$$p(\mathbf{m}|\mathbf{d}) = \frac{p(\mathbf{m})\ p(\mathbf{d}|\mathbf{m})}{\sum_{m'} p(\mathbf{m'})\ p(\mathbf{d}|\mathbf{m'})} \tag{1}$$

$$p(\boldsymbol{\theta}^m|\mathbf{d},\mathbf{m}) = \frac{p(\boldsymbol{\theta}^m|\mathbf{m})\ p(\mathbf{d}|\boldsymbol{\theta}^m,\mathbf{m})}{p(\mathbf{d}|\mathbf{m})} \tag{2}$$

where

$$p(\mathbf{d}|\mathbf{m}) = \int p(\mathbf{d}|\boldsymbol{\theta}^m,\mathbf{m})\ p(\boldsymbol{\theta}^m|\mathbf{m})\ d\boldsymbol{\theta}^m \tag{3}$$

is called the *marginal likelihood.* Given some hypothesis of interest, h, we determine the probability that h is true given data $\mathbf{d}$ by averaging over all possible models and their parameters:

$$p(h|\mathbf{d}) = \sum_m p(\mathbf{m}|\mathbf{d})\ p(h|\mathbf{d},\mathbf{m}) \tag{4}$$

$$p(h|\mathbf{d},\mathbf{m}) = \int p(h|\boldsymbol{\theta}^m,\mathbf{m})\ p(\boldsymbol{\theta}^m|\mathbf{d},\mathbf{m})\ d\boldsymbol{\theta}^m \tag{5}$$

For example, h may be the event that the next case $\mathbf{X}^{N+1}$ is observed in configuration $\mathbf{x}^{N+1}$. In this situation, we obtain

$$p(\mathbf{x}^{N+1}|\mathbf{d}) = \sum_m p(\mathbf{m}|\mathbf{d}) \int p(\mathbf{x}^{N+1}|\boldsymbol{\theta}^m,\mathbf{m})\ p(\boldsymbol{\theta}^m|\mathbf{d},\mathbf{m})\ d\boldsymbol{\theta}^m \tag{6}$$

where $p(\mathbf{x}^{N+1}|\boldsymbol{\theta}^m,\mathbf{m})$ is the likelihood for the model. It is important to note that, in the Bayesian approach, no single model is learned. Instead, data is used to update the probability that each possible model is the correct one.

Unfortunately, this approach—sometimes called *Bayesian model averaging* or the *full Bayesian approach*—is often impractical. For example, the number of different DAG models for a domain containing n variables grows super exponentially with n. Thus, the approach can only be applied in those few settings where one has strong prior knowledge that can eliminate almost all possible models.

Statisticians, who have been confronted by this problem for decades in the context of other types of models, use two approximations to address this problem: *Bayesian model selection* and *selective Bayesian model averaging.* The former approach is to select a likely model from among all possible models, and use it as if it were the correct model. For example, to predict the next case, we use

$$p(\mathbf{x}^{N+1}|\mathbf{d}) \cong p(\mathbf{x}^{N+1}|\mathbf{m},\mathbf{d}) = \int p(\mathbf{x}^{N+1}|\boldsymbol{\theta}^m,\mathbf{m})\ p(\boldsymbol{\theta}^m|\mathbf{d},\mathbf{m})\ d\boldsymbol{\theta}^m \tag{7}$$

where $\mathbf{m}$ is the selected model. The latter approach is to select a manageable number of good models from among all possible models and pretend that these models are exhaustive. In either approach, we need only the *relative* model posterior—$p(\mathbf{m})\ p(\mathbf{d}|\mathbf{m})$—to select likely models.

Both approaches can be characterized as *search-and-score* techniques. That is, in these approaches, we search among a large set of models looking for those with good scores. The use of these approximate methods raise several important questions. Do they yield accurate results when applied to graphical-model learning? If so, can we compute the model posteriors and perform search efficiently?

The question of accuracy is difficult to answer in theory. Nonetheless, several researchers have shown experimentally that the selection of a single good hypothesis often yields accurate predictions (e.g., Cooper and Herskovits 1992; Heckerman, Geiger, and Chickering, 1995) and that selective model averaging using Monte-Carlo methods can sometimes be efficient and yield even better predictions (Madigan et al., 1996). These results are somewhat surprising, and are largely responsible for the great deal of interest in learning graphical models.

In the remainder of this section, we address computational efficiency. In particular, we consider situations in which (relative) model posteriors can be computed efficiently as well as efficient search procedures.

We note that model averaging, model selection, and selective model averaging all help avoid overfitting—situations where models perform well on training data and poorly on new data. In particular, the marginal likelihood balances the fit of the model structure to data with the complexity of the model. One way to understand this fact is to note that, when the number of cases N is large and other conditions hold, the marginal likelihood can be approximated as follows:

$$p(\mathbf{d}|\mathbf{m}) \cong p(\mathbf{d}|\hat{\boldsymbol{\theta}}, \mathbf{m}) - \frac{|\boldsymbol{\theta}|}{2} log N$$

where $\hat{\boldsymbol{\theta}}$ is the maximum-likelihood estimator of the data (e.g., Kass and Raftery, 1995). The first quantity in this expression represents the degree to which the model fits the data, which increases as the model complexity increases. The second quantity, in contrast, penalizes model complexity.

Computation of the Marginal Likelihood

Under certain conditions, the marginal likelihood of a graphical model—and hence its relative posterior—can be computed efficiently. In this section, we examine a particular set of these conditions for structure learning of DAG models. We note that a similar set of conditions hold for the learning of decomposable UG models. For details, see Lauritzen

(1996).

Given any DAG model $\mathbf{m}$, we can factor the likelihood of a single sample as follows:

$$p(\mathbf{x}|\boldsymbol{\theta}_m, \mathbf{m}) = \prod_{i=1}^{n} p(x_i|\mathbf{pa}_i, \boldsymbol{\theta}_i, \mathbf{m}) \tag{8}$$

We shall refer to each term $p(x_i|\mathbf{pa}_i, \boldsymbol{\theta}_i, \mathbf{m})$ in this equation as the *local likelihood for X_i*. Also, in this equation, $\boldsymbol{\theta}_i$ denotes the set of parameters associated with the local likelihood for variable X_i.

The first condition in our set of sufficient conditions yielding efficient computation is that each local likelihood is in the exponential family. One example of such a factorization occurs when each variable $X_i \in \mathbf{X}$ is finite, having r_i possible values $x_i^1, \ldots, x_i^{r_i}$, and each local likelihood is a collection of multinomial distributions, one distribution for each configuration of $\mathbf{Pa}_i$—that is,

$$p(x_i^k|\mathbf{pa}_i^j, \boldsymbol{\theta}_i, \mathbf{m}) = \theta_{ijk} > 0 \tag{9}$$

where $\mathbf{pa}_i^1, \ldots, \mathbf{pa}_i^{q_i}$ ($q_i = \prod_{X_i \in \mathbf{Pa}_i} r_i$) denote the configurations of $\mathbf{Pa}_i$, and $\boldsymbol{\theta}_i = ((\theta_{ijk})_{k=2}^{r_i})_{j=1}^{q_i}$ are the parameters. The parameter θ_{ij1} is given by $1 - \sum_{k=2}^{r_i} \theta_{ijk}$. We shall use this example to illustrate many of the concepts in this article. For convenience, we define the vector of parameters

$$\boldsymbol{\theta}_{ij} = (\theta_{ij2}, \ldots, \theta_{ijr_i})$$

for all i and j. Examples of other exponential families can be found in Bernardo and Smith (1994).

The second assumption for efficient computation is one of parameter independence. In our multinomial example, we assume that the parameter vectors $\boldsymbol{\theta}_{ij}$ are mutually independent. Note that, when this independence holds and we are given a random sample $\mathbf{d}$ that contains no missing observations, the parameters remain independent:

$$p(\boldsymbol{\theta}_m|\mathbf{d}, \mathbf{m}) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} p(\boldsymbol{\theta}_{ij}|\mathbf{d}, \mathbf{m}) \tag{10}$$

Thus, we can update each vector of parameters $\boldsymbol{\theta}_{ij}$ independently.

The third assumption is that each independent parameter set has a conjugate prior (e.g., Bernardo and Smith, 1994). In our multinomial example, we assume that each $\boldsymbol{\theta}_{ij}$ has a Dirichlet prior $\text{Dir}(\boldsymbol{\theta}_{ij}|\alpha_{ij1}, \ldots, \alpha_{ijr_i})$. In this case, we obtain

$$p(\boldsymbol{\theta}_{ij}|\mathbf{d}, \mathbf{m}) = \text{Dir}(\boldsymbol{\theta}_{ij}|\alpha_{ij1} + N_{ij1}, \ldots, \alpha_{ijr_i} + N_{ijr_i}) \tag{11}$$

where N_{ijk} is the number of cases in $\mathbf{d}$ in which $X_i = x_i^k$ and $\mathbf{Pa}_i = \mathbf{pa}_i^j$. Note that the collection of counts N_{ijk} are sufficient statistics of the data for the model $\mathbf{m}$.

Under these conditions, we can compute the marginal likelihood efficiently and in closed form. For our multinomial example (as first derived in Cooper and Herskovits, 1992), we obtain

$$p(\mathbf{d}|\mathbf{m}) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \cdot \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \tag{12}$$

where $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$ and $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$.

Under these same conditions, the integral in Equation 7 also can be computed efficiently. In our example, suppose that, for a given outcome $\mathbf{x}_{N+1}$ of $\mathbf{X}_{N+1}$, the value of X_i is x_i^k and the configuration of $\mathbf{Pa}_i$ is $\mathbf{pa}_i^j$, where k and j depend on i. Using Equations 4, 8, and 9, we obtain

$$p(\mathbf{x}_{N+1}|\mathbf{d}, \mathbf{m}) = \int \left(\prod_{i=1}^{n} \theta_{ijk} \right) p(\boldsymbol{\theta}_m|\mathbf{d}, \mathbf{m}) \; d\boldsymbol{\theta}_m$$

Because parameters remain independent given $\mathbf{d}$, we get

$$p(\mathbf{x}_{N+1}|\mathbf{d}, \mathbf{m}) = \prod_{i=1}^{n} \int \theta_{ijk} \; p(\boldsymbol{\theta}_{ij}|\mathbf{d}, \mathbf{m}) \; d\boldsymbol{\theta}_{ij}$$

Finally, because each integral in this product is the expectation of a Dirichlet distribution, we have

$$p(\mathbf{x}_{N+1}|\mathbf{d}, \mathbf{m}) = \prod_{i=1}^{n} \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}} \tag{13}$$

To compute the relative posterior probability of a model, we must assess the structure prior $p(\mathbf{m})$ and the parameter priors $p(\boldsymbol{\theta}^m|\mathbf{m})$. Unfortunately, when many models are possible, the assessment process will be intractable. Nonetheless, under certain assumptions, we can derive the structure and parameter priors for many models from a manageable number of direct assessments. Several authors have discussed such assumptions and corresponding methods for deriving priors (e.g., Buntine, 1991; Cooper and Herskovits, 1992; Heckerman, Geiger, and Chickering, 1995; Cowell et al., 1999). In the following two sections, we examine some of these approaches.

Priors for Model Parameters

First, let us consider the assessment of priors for the parameters of DAG models. We consider the approach of Heckerman, Geiger, and Chickering (1995)—herein, HGC—who address the case for $\mathbf{X}$ where the local likelihoods are multinomial distributions. A similar approach exists for situations where the local likelihoods are linear regressions (Heckerman and Geiger, 1995)..

Their approach is based on two key concepts: Markov equivalence and distribution equivalence. We say that two models for $\mathbf{X}$ are *Markov equivalent* if they represent the same set of conditional-independence assertions for $\mathbf{X}$. For example, given $\mathbf{X} = \{X, Y, Z\}$, the models $X \rightarrow Y \rightarrow Z$, $X \leftarrow Y \rightarrow Z$, and $X \leftarrow Y \leftarrow Z$ represent only the independence assertion that X and Z are conditionally independent given Y. Consequently, these models are equivalent. Another example of Markov equivalence is the set of *complete models* on $\mathbf{X}$. A complete model is one that has no missing edge and which encodes no assertion of conditional independence. When $\mathbf{d}$ contains n variables, there are $n!$ possible complete models, one model structure for every possible ordering of the variables. All complete models for $p(\mathbf{d})$ are Markov equivalent. In general, two models are Markov equivalent if and only if they have the same structure ignoring arc directions and the same v-structures. A *v-structure* is an ordered tuple (X, Y, Z) such that there is an arc from X to Y and from Z to Y, but no arc between X and Z.

The concept of distribution equivalence is closely related to that of Markov equivalence. Suppose that all models for $\mathbf{X}$ under consideration have local likelihoods in the family $\mathcal{F}$. This is not a restriction, per se, because $\mathcal{F}$ can be a large family. We say that two model structures $\mathbf{m}_1$ and $\mathbf{m}_2$ for $\mathbf{X}$ are *distribution equivalent with respect to (wrt)* $\mathcal{F}$ if they can represent the same joint probability distributions for $\mathbf{X}$—that is, if, for every $\boldsymbol{\theta}_{m1}$, there exists a $\boldsymbol{\theta}_{m2}$ such that $p(\mathbf{x}|\boldsymbol{\theta}_{m1}, \mathbf{m}_1) = p(\mathbf{x}|\boldsymbol{\theta}_{m2}, \mathbf{m}_2)$, and vice versa.

Distribution equivalence wrt some $\mathcal{F}$ implies Markov equivalence, but the converse does not hold. For example, when $\mathcal{F}$ is the family of generalized linear-regression models, the complete model structures for $n \geq 3$ variables do not represent the same sets of distributions. Nonetheless, there are families $\mathcal{F}$—for example, multinomial distributions and linear-regression models with Gaussian noise—where Markov equivalence implies distribution equivalence wrt $\mathcal{F}$ (see HGC). The notion of distribution equivalence is important, because if two model structures $\mathbf{m}_1$ and $\mathbf{m}_2$ are distribution equivalent with respect to a given $\mathcal{F}$, then it is often reasonable to expect that data can not help to discriminate them. That is, we expect $p(\mathbf{d}|\mathbf{m}_1) = p(\mathbf{d}|\mathbf{m}_2)$ for any data set $\mathbf{d}$. HGC call this property *likelihood equivalence.*

Now let us return to the main issue of this section: the derivation of parameter priors from a manageable number of assessments. HGC show that the assumption of likelihood equivalence combined with the assumption that the $\boldsymbol{\theta}_{ij}$ are mutually independent imply that the parameters for any *complete* model $\mathbf{m}_c$ must have a Dirichlet distribution with constraints on the hyperparameters given by

$$\alpha_{ijk} = \alpha \; p(x_i^k, \mathbf{pa}_i^j|\mathbf{m}_c) \tag{14}$$

where α is the user's equivalent sample size[1], and $p(x_i^k, \mathbf{pa}_i^j|\mathbf{m}_c)$ is computed from the user's

[1] Discussions of equivalent sample size can be found in—for example—Heckerman, Geiger, and Chickering (1995).

joint probability distribution $p(\mathbf{d}|\mathbf{m}_c)$. Note that this result is rather surprising, as the two assumptions leading to the constrained Dirichlet solution are qualitative.

To determine the priors for parameters of *incomplete* models HGC use the assumption of *parameter modularity,* which says that if X_i has the same parents in models $\mathbf{m}_1$ and $\mathbf{m}_2$, then

$$p(\boldsymbol{\theta}_{ij}|\mathbf{m}_1) = p(\boldsymbol{\theta}_{ij}|\mathbf{m}_2)$$

for $j = 1, \ldots, q_i$. They call this property parameter modularity, because it says that the distributions for parameters $\boldsymbol{\theta}_{ij}$ depend only on a portion of the graph structure—namely, X_i and its parents.

Given the assumptions of parameter modularity and parameter independence, it is a simple matter to construct priors for the parameters of an arbitrary model given the priors on complete models. In particular, given parameter independence, we construct the priors for the parameters of each node separately. Furthermore, if node X_i has parents $\mathbf{Pa}_i$ in the given model, then we identify a complete model structure where X_i has these parents, and use Equation 14 and parameter modularity to determine the priors for this node. The result is that all terms α_{ijk} for all model structures are determined by Equation 14. Thus, from the assessments α and $p(\mathbf{d}|\mathbf{m}_c)$, we can derive the parameter priors for all possible model structures. We can assess $p(\mathbf{d}|\mathbf{m}_c)$ by constructing a parameterized model called a *prior network*, that encodes this joint distribution.

Priors for Model Structures

Now, let us consider the assessment of priors on structure. The simplest approach for assigning priors to models is to assume that every model is equally likely. Of course, this assumption is typically inaccurate and used only for the sake of convenience. A simple refinement of this approach is to ask the user to exclude various structures (perhaps based on judgments of cause and effect), and then impose a uniform prior on the remaining structures. We use this approach in an example described later.

Buntine (1991) describes a set of assumptions that leads to a richer yet efficient approach for assigning priors. The first assumption is that the variables can be ordered (e.g., through a knowledge of time precedence). The second assumption is that the presence or absence of possible arcs are mutually independent. Given these assumptions, $n(n-1)/2$ probability assessments (one for each possible arc in an ordering) determines the prior probability of every possible model. One extension to this approach is to allow for multiple possible orderings. One simplification is to assume that the probability that an arc is absent or present is independent of the specific arc in question. In this case, only one probability assessment is required.

An alternative approach, described by Heckerman et al. (1995) uses the prior network described in the previous section. The basic idea is to penalize the prior probability of any structure according to some measure of deviation between that structure and the prior network. Heckerman et al. (1995) suggest one reasonable measure of deviation.

Search Methods

In this section, we examine search methods for identifying DAG models with high scores. Consider the problem of finding the best DAG model from the set of all DAG models in which each node has no more than k parents. Unfortunately, the problem for $k > 1$ is NP-hard even when we use the restrictive prior given by Equation 14 (Chickering, 1996). Thus, researchers have used heuristic search algorithms, including greedy search, greedy search with restarts, best-first search, and Monte-Carlo methods.

One consolation is that these search methods can be made more computationally efficient when the model score is factorable. Given a DAG model for domain $\mathbf{X}$, we say that a score for that model $\mathrm{S}(\mathbf{m}, \mathbf{d})$ is *factorable* if it can be written as a product of variable-specific scores:

$$\mathrm{S}(\mathbf{m}, \mathbf{d}) = \prod_{i=1}^{n} s(X_i, \mathbf{Pa}_i, \mathbf{d}_i) \tag{15}$$

where $\mathbf{d}_i$ is the data restricted to the variables X_i and $\mathbf{Pa}_i$. An example of a factorable score is Equation 12 used in conjunction with any of the structure priors described previously.

Most of the commonly used search methods for DAG models also make successive arc changes to the graph structure, and employ the property of factorability to evaluate the merit of each change. One commonly used set of arc changes is as follows. For any pair of variables, if there is an arc connecting them, then this arc can either be reversed or removed. If there is no arc connecting them, then an arc can be added in either direction. All changes are subject to the constraint that the resulting DAG contains no directed cycles. We use E to denote the set of eligible changes to a graph, and $\Delta(e)$ to denote the change in $\log p(\mathbf{d}|\mathbf{m})p(\mathbf{m})$ resulting from the modification $e \in E$. Given a factorable score, if an arc to X_i is added or deleted, only $c(X_i, \mathbf{Pa}_i, \mathbf{d}_i)$ need be evaluated to determine $\Delta(e)$. If an arc between X_i and X_j is reversed, then only $c(X_i, \mathbf{Pa}_i, \mathbf{d}_i)$ and $c(X_j, \Pi_j, \mathbf{d}_j)$ need be evaluated.

One simple heuristic search algorithm is greedy hill climbing. We begin with some DAG model. Then, we evaluate $\Delta(e)$ for all $e \in E$, and make the change e for which $\Delta(e)$ is a maximum, provided it is positive. We terminate search when there is no e with a positive value for $\Delta(e)$. Candidates for the initial model include the empty graph, a random graph, and the prior network used for the assessment of parameter and structure priors.

A potential problem with any local-search method is getting stuck at a local maximum.

One method for escaping local maxima is greedy search with random restarts. In this approach, we apply greedy search until we hit a local maximum. Then, we randomly perturb the structure, and repeat the process for some manageable number of iterations. Another method for escaping local maxima is simulated annealing. In this approach, we initialize the system at some temperature T_0. Then, we pick some eligible change e at random, and evaluate the expression $p = \exp(\Delta(e)/T_0)$. If $p > 1$, then we make the change e; otherwise, we make the change with probability p. We repeat this selection and evaluation process α times or until we make β changes. If we make no changes in α repetitions, then we stop searching. Otherwise, we lower the temperature by multiplying the current temperature T_0 by a decay factor $0 < \gamma < 1$, and continue the search process. We stop searching if we have lowered the temperature more than δ times. Thus, this algorithm is controlled by five parameters: $T_0, \alpha, \beta, \gamma$ and δ. To initialize this algorithm, we can start with the empty graph, and make T_0 large enough so that almost every eligible change is made, thus creating a random graph. Alternatively, we may start with a lower temperature, and use one of the initialization methods described for local search.

Another method for escaping local maxima is best-first search. In this approach, the space of all models is searched systematically using a heuristic measure that determines the next best structure to examine. Experiments (e.g., Heckerman, Geiger, and Chickering, 1995) have shown that, for a fixed amount of computation time, greedy search with random restarts produces better models than does best-first search.

One important consideration for any search algorithm is the search space. The methods that we have described search through the space of DAG models. Nonetheless, when likelihood equivalence is assumed, one can search through the space of model equivalence classes. One benefit of the latter approach is that the search space is smaller. One drawback of the latter approach is that it takes longer to move from one element in the search space to another. Experiments have shown that the two effects roughly cancel.

Example: College Plans

In this section, we consider an analysis of data, obtain by Sewell and Shah (1968), regarding factors that influence the intention of high school students to attend college. This analysis was given previously by Heckerman in Jordan (1999).

Sewell and Shah (1968) measured the following variables for 10,318 Wisconsin high school seniors: *Sex* (SEX): male, female; *Socioeconomic Status* (SES): low, lower middle, upper middle, high; *Intelligence Quotient* (IQ): low, lower middle, upper middle, high; *Parental Encouragement* (PE): low, high; and *College Plans* (CP): yes, no. Our goal in this analysis is to understand the relationships among these variables.

The data are (completely) described by the counts in Table . Each entry denotes the number of cases in which the five variables take on some particular configuration. The first entry corresponds to the configuration SEX=male, SES=low, IQ=low, PE=low, and CP=yes. The remaining entries correspond to configurations obtained by cycling through the states of each variable such that the last variable (CP) varies most quickly. Thus, for example, the upper (lower) half of the table corresponds to male (female) students.

To generate priors for model parameters, we used the method described earlier in this section with an equivalent sample size of five and a prior network describing a uniform distribution over $\mathbf{X}$. (The results we report remain the same for equivalent sample sizes ranging from 3 to 40.) For structure priors, we assumed that all models were equally likely, except we excluded structures (based on causal considerations) where SEX and/or SES had parents, and/or CP had children. We used Equation 12 to compute the marginal likelihoods of the models. The two most likely models that we found after an exhaustive search over all structures are shown in Figure 1. Note that the most likely model has a posterior probability that is extremely close to one. Both models show a reasonable result: that CP and SEX are independent, given the remaining variables.

Methods for Incomplete Data

Among the assumptions that yield a efficient method for computing the marginal likelihood, the one that is most often violated is the assumption that all variables are observed in every case. In many situations, some variables will be hidden (i.e., never observed) or will be observed for only a subset of the data samples. There are a variety of methods for handling such situations—albeit at greater computational cost—including Monte-Carlo (MC) approaches (e.g., DiCiccio, Kass, Raftery, and Wasserman, 1995), large-sample approximations (e.g., Kass and Raftery, 1995), and variational approximations (e.g., Jordan et al. in Jordan, 1999).

In this section, we examine a simple MC approach called *Gibbs sampling* (e.g., MacKay in Jordan, 1999). In general, given variables $\mathbf{X} = \{X_1, \ldots, X_n\}$ with some joint distribution $p(\mathbf{x})$, we can use a Gibbs sampler to approximate the expectation of a function $f(\mathbf{x})$ with respect to $p(\mathbf{x})$. This approximation is made as follows. First, we choose an initial state for each of the variables in $\mathbf{X}$ somehow (e.g., at random). Next, we pick some variable X_i, unassign its current state, and compute its probability distribution given the states of the other $n-1$ variables. Then, we sample a state for X_i based on this probability distribution, and compute $f(\mathbf{x})$. Finally, we iterate the previous two steps, keeping track of the average value of $f(\mathbf{x})$. In the limit, as the number of cases approach infinity, this average is equal to $\mathrm{E}_{p(\mathbf{x})}(f(\mathbf{x}))$ provided two conditions are met. First, the Gibbs sampler must be *irreducible*. That is, the probability distribution $p(\mathbf{x})$ must be such that we can eventually sample any possible configuration of $\mathbf{X}$ given any possible initial configuration of $\mathbf{X}$. For example, if $p(\mathbf{x})$ contains no zero probabilities, then the Gibbs sampler will be irreducible. Second, each X_i must be chosen infinitely often. In practice, an algorithm for deterministically rotating through the variables is typically used. An Introduction to Gibbs sampling and other Monte-Carlo methods—including methods for initialization and a discussion of convergence—is given by Neal (1993).

To illustrate Gibbs sampling, consider again the case where every variable in $\mathbf{X}$ is finite, the parameters $\boldsymbol{\theta}_{ij}$ for a given DAG model $\mathbf{m}$ are mutually independent, and each $\boldsymbol{\theta}_{ij}$ has a Dirichlet prior. In this situation, let us approximate the probability density $p(\boldsymbol{\theta}_m|\mathbf{d}, \mathbf{m})$ for some particular configuration of $\boldsymbol{\theta}_m$, given an incomplete data set $\mathbf{d}$. First, we initialize the states of the unobserved variables in each case somehow. As a result, we have a complete random sample $\mathbf{d}_c$. Second, we choose some variable X_{il} (variable X_i in case l) that is not observed in the original random sample D, and reassign its state according to the probability distribution

$$p(x'_{il}|\mathbf{d}_c \setminus x_{il}, \mathbf{m}) = \frac{p(x'_{il}, \mathbf{d}_c \setminus x_{il}|\mathbf{m})}{\sum_{x''_{il}} p(x''_{il}, \mathbf{d}_c \setminus x_{il}|\mathbf{m})}$$

where $\mathbf{d}_c \setminus x_{il}$ denotes the data set $\mathbf{d}_c$ with observation x_{il} removed, and the sum in the denominator runs over all states of variable X_{il}. As we have seen, the terms in the numerator and denominator can be computed efficiently (see Equation 12). Third, we repeat this reassignment for all unobserved variables in $\mathbf{d}$, producing a new complete random sample $\mathbf{d}'_c$. Fourth, we compute the posterior density $p(\boldsymbol{\theta}_m|\mathbf{d}'_c, \mathbf{m})$ as described in Equations 10 and 11. Finally, we iterate the previous three steps, and use the average of $p(\boldsymbol{\theta}_m|\mathbf{d}'_c, \mathbf{m})$ as our approximation.

Monte-Carlo approximations are also useful for computing the marginal likelihood given incomplete data. One Monte-Carlo approach uses Bayes' theorem:

$$p(\mathbf{d}|\mathbf{m}) = \frac{p(\boldsymbol{\theta}_m|\mathbf{m})\ p(\mathbf{d}|\boldsymbol{\theta}_m, \mathbf{m})}{p(\boldsymbol{\theta}_m|\mathbf{d}, \mathbf{m})} \tag{16}$$

For any configuration of $\boldsymbol{\theta}_m$, the prior term in the numerator can be evaluated directly. In addition, the likelihood term in the numerator can be computed using DAG-model inference (e.g., Kjaerulff in Jordan, 1999). Finally, the posterior term in the denominator can be computed using Gibbs sampling, as we have just described.

NON-BAYESIAN APPROACHES

In this section, we consider several commonly used alternatives to the Bayesian approach for structure learning.

One such class of algorithms mimic the search-and-score approach of Bayesian model selection but incorporate a non-Bayesian score. Alternative scores include (1) prediction accuracy on new data, (2) prediction accuracy over cross-validated data sets, and (3) non-Bayesian information criteria such as AIC.

Another class of algorithms for structure learning is the *constraint-based* approach described by Pearl (2000) and Spirtes, Glymour, and Scheines (2002). In this set of algorithms, statistical tests are performed on the data to determine independence and dependence relationships among the variables. Then, search methods are used to identify one or more models that are consistent with those relationships.

To illustrate this approach, suppose we seek to learn one or more DAG models given data for three finite variables (X_1, X_2, X_3). Assuming each local likelihood is a collection of multinomial distributions, there are eleven possible DAG models that are distinct: (1) a complete model, (2) $X_1 \rightarrow X_2 \rightarrow X_3$, (3) $X_1 \rightarrow X_3 \rightarrow X_2$, (4) $X_2 \rightarrow X_1 \rightarrow X_3$, (5) $X_1 \rightarrow X_2 \leftarrow X_3$, (6) $X_1 \rightarrow X_3 \leftarrow X_2$, (7) $X_2 \rightarrow X_1 \leftarrow X_3$, (8) $X_1 \rightarrow X_2X_3$, (9) $X_1 \rightarrow X_3X_2$, (10) $X_2 \rightarrow X_3X_1$, and (11) $X_1X_2X_3$, where X_iX_j means there is no arc between X_i and X_j. There are other possible models that are not listed, but each such model represents a set of distributions that is equivalent to one of the other models above. For example, $X_3 \rightarrow X_2 \rightarrow X_1$ and model 2 are distribution equivalent.

Now, suppose that statistical tests applied to the data reveal that the *only* independence relationship is that X_1 and X_3 are independent. Only models 1 and 5 can exhibit only this independence. Furthermore, if we use parameter prior assignments of the form described earlier in this section, then model 1 will exhibit this independence with probably zero. Consequently, we conclude that model 5 is correct (with probability one).

One drawback of the constraint-based approach is that any statistical test will be an approximation for finite data; and errors in the tests may lead the search mechanism to (1) conclude that the found relationships are inconsistent or (2) return erroneous models. One advantage of the approach over most search-and-score methods is that more structures can be considered for a fixed amount computation, because the results of some statistical tests can greatly constrain model search.

REFERNCES

* Bernardo, J. and Smith, A. (1994). Bayesian Theory. John Wiley and Sons, New York.

Buntine, W. (1991). Theory refinement on Bayesian networks. In Proceedings of Seventh Conference on Uncertainty in Artificial Intelligence, Los Angeles, CA, pages 52–60. Morgan Kaufmann.

Chickering, D. (1996). Learning Bayesian networks is NP-complete. In Fisher, D. and Lenz, H., editors, Learning from Data, pages 121–130. Springer-Verlag.

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

* Cowell, R., Dawid, A. P., Lauritzen, S., and Spiegelhalter, D. (1999). Probabilistic Networks and Expert Systems (Statistics for Engineering and Information Science). Springer Verlag.

DiCiccio, T., Kass, R., Raftery, A., and Wasserman, L. (July, 1995). Computing Bayes factors by combining simulation and asymptotic approximations. Technical Report 630, Department of Statistics, Carnegie Mellon University, PA.

Heckerman, D. and Geiger, D. (1995). Learning Bayesian networks: A unification for discrete and Gaussian domains. In Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence, Montreal, QU, pages 274–284. Morgan Kaufmann.

Heckerman, D., Geiger, D., and Chickering, D. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. Machine Learning, 20:197–243.

* Jordan, M., editor (1999). Learning in Graphical Models. MIT Press.

* Kass, R. and Raftery, A. (1995). Bayes factors. Journal of the American Statistical Association, 90:773–795.

* Lauritzen, S. (1996). Graphical Models. Claredon Press.

Madigan, D., Raftery, A., Volinsky, C., and Hoeting, J. (1996). Bayesian model averaging. In Proceedings of the AAAI Workshop on Integrating Multiple Learned Models, Portland, OR.

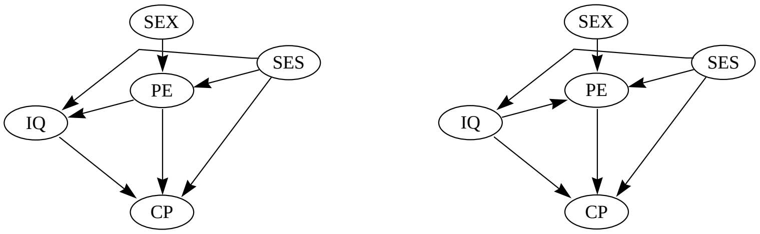
* Pearl, J., editor (2000). Causality: Models, Reasoning, and Inference. Cambridge University Press.

Sewell, W. and Shah, V. (1968). Social class, parental encouragement, and educational aspirations. American Journal of Sociology, 73:559–572.

* Spirtes, P., Glymour, C., and Scheines, R. (2001). Causation, Prediction, and Search, Second Edition. MIT Press, Cambridge, MA.

Table 1: Sufficient statistics for the Sewall and Shah (1968) study.

4	349	13	64	9	207	33	72	12	126	38	54	10	67	49	43
2	232	27	84	7	201	64	95	12	115	93	92	17	79	119	59
8	166	47	91	6	120	74	110	17	92	148	100	6	42	198	73
4	48	39	57	5	47	123	90	9	41	224	65	8	17	414	54
5	454	9	44	5	312	14	47	8	216	20	35	13	96	28	24
11	285	29	61	19	236	47	88	12	164	62	85	15	113	72	50
7	163	36	72	13	193	75	90	12	174	91	100	20	81	142	77
6	50	36	58	5	70	110	76	12	48	230	81	13	49	360	98



$\log p(D|S_1^h) = -45653$

$p(S_1^h|D) = 1.0$

$\log p(D|S_2^h) = -45699$

$p(S_2^h|D) = 1.2 \times 10^{-10}$

Figure 1: The a posteriori most likely models.