A Tutorial on Learning With Bayesian Networks

David Heckerman
heckerma@hotmail.com

November 1996 (Revised January 2020)

Abstract

A Bayesian network is a graphical model that encodes probabilistic relationships among variables of interest. When used in conjunction with statistical techniques, the graphical model has several advantages for data analysis. One, because the model encodes dependencies among all variables, it readily handles situations where some data entries are missing. Two, a Bayesian network can be used to learn causal relationships, and hence can be used to gain understanding about a problem domain and to predict the consequences of intervention. Three, because the model has both a causal and probabilistic semantics, it is an ideal representation for combining prior knowledge (which often comes in causal form) and data. Four, Bayesian statistical methods in conjunction with Bayesian networks offer an efficient and principled approach for avoiding the overfitting of data. In this paper, we discuss methods for constructing Bayesian networks from prior knowledge and summarize Bayesian statistical methods for using data to improve these models. With regard to the latter task, we describe methods for learning both the parameters and structure of a Bayesian network, including techniques for learning with incomplete data. In addition, we relate Bayesian-network methods for learning to techniques for supervised and unsupervised learning. We illustrate the graphical-modeling approach using a real-world case study.

1 Introduction

A Bayesian network is a graphical model for probabilistic relationships among a set of variables. Over the last decade, the Bayesian network has become a popular representation for encoding uncertain expert knowledge in expert systems (Heckerman *et al.*, 1995a). More recently, researchers have developed methods for learning Bayesian networks from data. The techniques that have been developed are new and still evolving, but they have been shown to be remarkably effective for some data-analysis problems.

In this paper, we provide a tutorial on Bayesian networks and associated Bayesian techniques for extracting and encoding knowledge from data. There are numerous representations available for data analysis, including rule bases, decision trees, and artificial neural networks; and there are many techniques for data analysis such as density estimation, classification, regression, and clustering. So what do Bayesian networks and Bayesian methods have to offer? There are at least four answers.

One, Bayesian networks can readily handle incomplete data sets. For example, consider a classification or regression problem where two of the explanatory or input variables are strongly anti-correlated. This correlation is not a problem for standard supervised learning techniques, provided all inputs are measured in every case. When one of the inputs is not observed, however, most models will produce an inaccurate prediction, because they do not encode the correlation between the input variables. Bayesian networks offer a natural way to encode such dependencies.

Two, Bayesian networks allow one to learn about causal relationships. Learning about causal relationships are important for at least two reasons. The process is useful when we are trying to gain understanding about a problem domain, for example, during exploratory data analysis. In addition, knowledge of causal relationships allows us to make predictions in the presence of interventions. For example, a marketing analyst may want to know whether or not it is worthwhile to increase exposure of a particular advertisement in order to increase the sales of a product. To answer this question, the analyst can determine whether or not the advertisement is a cause for increased sales, and to what degree. The use of Bayesian networks helps to answer such questions even when no experiment about the effects of increased exposure is available.

Three, Bayesian networks in conjunction with Bayesian statistical techniques facilitate the combination of domain knowledge and data. Anyone who has performed a real-world analysis knows the importance of prior or domain knowledge, especially when data is scarce or expensive. The fact that some commercial systems (i.e., expert systems) can be built from prior knowledge alone is a testament to the power of prior knowledge. Bayesian networks have a causal semantics that makes the encoding of causal prior knowledge particularly straightforward. In addition, Bayesian networks encode the strength of causal relationships with probabilities. Consequently, prior knowledge and data can be combined with well-studied techniques from Bayesian statistics.

Four, Bayesian methods in conjunction with Bayesian networks and other types of models offers an efficient and principled approach for avoiding the over fitting of data. As we shall see, there is no need to hold out some of the available data for testing. Using the Bayesian approach, models can be "smoothed" in such a way that all available data can be used for training.

This tutorial is organized as follows. In Section 2, we discuss the Bayesian interpretation of probability and review methods from Bayesian statistics for combining prior knowledge with data. In Section 3, we describe Bayesian networks and discuss how they can be constructed from prior knowledge alone. In Section 4, we discuss algorithms for probabilistic inference in a Bayesian network. In Sections 5 and 6, we show how to learn the probabilities in a fixed Bayesian-network structure, and describe techniques for handling incomplete data including Monte-Carlo methods and the Gaussian approximation. In Sections 7 through 12, we show how to learn both the probabilities and structure of a Bayesian network. Topics discussed include methods for assessing priors for Bayesian-network structure and parameters, and methods for avoiding the overfitting of data including Monte-Carlo, Laplace, BIC, and MDL approximations. In Sections 13 and 14, we describe the relationships between Bayesian-network techniques and methods for supervised and unsupervised learning. In Section 15, we show how Bayesian networks facilitate the learning of causal relationships. In Section 16, we illustrate techniques discussed in the tutorial using a real-world case study. In Section 17, we give pointers to software and additional literature.

2 The Bayesian Approach to Probability and Statistics

To understand Bayesian networks and associated learning techniques, it is important to understand the Bayesian approach to probability and statistics. In this section, we provide an introduction to the Bayesian approach for those readers familiar only with the classical view.

In a nutshell, the Bayesian probability of an event x is a person's *degree of belief* in that event. Whereas a classical probability is a physical property of the world (e.g., the probability that a coin will land heads), a Bayesian probability is a property of the person who assigns the probability (e.g., your degree of belief that the coin will land heads). To keep these two concepts of probability distinct, we refer to the classical probability of an event as the true or physical probability of that event, and refer to a degree of belief in an event as a Bayesian or personal probability. Alternatively, when the meaning is clear, we refer to a Bayesian probability simply as a probability.

One important difference between physical probability and personal probability is that, to measure the latter, we do not need repeated trials. For example, imagine the repeated tosses of a sugar cube onto a wet surface. Every time the cube is tossed, its dimensions will change slightly. Thus, although the classical statistician has a hard time measuring the probability that the cube will land with a particular face up, the Bayesian simply restricts his or her attention to the next toss, and assigns a probability. As another example, consider the question: What is the probability that the Chicago Bulls will win the championship in 2001? Here, the classical statistician must remain silent, whereas the Bayesian can assign a probability (and perhaps make a bit of money in the process).

One common criticism of the Bayesian definition of probability is that probabilities seem arbitrary. Why should degrees of belief satisfy the rules of probability? On what scale should probabilities be measured? In particular, it makes sense to assign a probability of one (zero) to an event that will (not) occur, but what probabilities do we assign to beliefs that are not at the extremes? Not surprisingly, these questions have been studied intensely.

With regards to the first question, many researchers have suggested different sets of properties that should be satisfied by degrees of belief (e.g., Ramsey 1931, Cox 1946, Good 1950, Savage 1954, DeFinetti 1970). It turns out that each set of properties leads to the same rules: the rules of probability. Although each set of properties is in itself compelling, the fact that different sets all lead to the rules of probability provides a particularly strong argument for using probability to measure beliefs.

The answer to the question of scale follows from a simple observation: people find it fairly easy to say that two events are equally likely. For example, imagine a simplified wheel of fortune having only two regions (shaded and not shaded), such as the one illustrated in Figure 1. Assuming everything about the wheel as symmetric (except for shading), you should conclude that it is equally likely for the wheel to stop in any one position. From this judgment and the sum rule of probability (probabilities of mutually exclusive and collectively exhaustive sum to one), it follows that your probability that the wheel will stop in the shaded region is the percent area of the wheel that is shaded (in this case, 0.3).

This *probability wheel* now provides a reference for measuring your probabilities of other events. For example, what is your probability that Al Gore will run on the Democratic ticket in 2000? First,

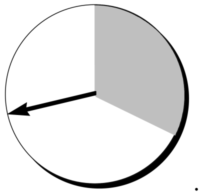


Figure 1: The probability wheel: a tool for assessing probabilities.

ask yourself the question: Is it more likely that Gore will run or that the wheel when spun will stop in the shaded region? If you think that it is more likely that Gore will run, then imagine another wheel where the shaded region is larger. If you think that it is more likely that the wheel will stop in the shaded region, then imagine another wheel where the shaded region is smaller. Now, repeat this process until you think that Gore running and the wheel stopping in the shaded region are equally likely. At this point, your probability that Gore will run is just the percent surface area of the shaded area on the wheel.

In general, the process of measuring a degree of belief is commonly referred to as a *probability assessment*. The technique for assessment that we have just described is one of many available techniques discussed in the Management Science, Operations Research, and Psychology literature. One problem with probability assessment that is addressed in this literature is that of precision. Can one really say that his or her probability for event x is 0.601 and not 0.599? In most cases, no. Nonetheless, in most cases, probabilities are used to make decisions, and these decisions are not sensitive to small variations in probabilities. Well-established practices of *sensitivity analysis* help one to know when additional precision is unnecessary (e.g., Howard and Matheson, 1983). Another problem with probability assessment is that of accuracy. For example, recent experiences or the way a question is phrased can lead to assessments that do not reflect a person's true beliefs (Tversky and Kahneman, 1974). Methods for improving accuracy can be found in the decision-analysis literature (e.g, Spetzler *et al.* (1975)).

Now let us turn to the issue of learning with data. To illustrate the Bayesian approach, consider a common thumbtack—one with a round, flat head that can be found in most supermarkets. If we throw the thumbtack up in the air, it will come to rest either on its point (*heads*) or on its head (*tails*).[1] Suppose we flip the thumbtack $N+1$ times, making sure that the physical properties of the thumbtack and the conditions under which it is flipped remain stable over time. From the first N observations, we want to determine the probability of heads on the $N+1$th toss.

In the classical analysis of this problem, we assert that there is some physical probability of heads, which is unknown. We *estimate* this physical probability from the N observations using criteria such as low bias and low variance. We then use this estimate as our probability for heads on the $N+1$th toss. In the Bayesian approach, we also assert that there is some physical probability of heads, but we encode our uncertainty about this physical probability using (Bayesian) probabilities, and use the rules of probability to compute our probability of heads on the $N+1$th toss.[2]

[1] This example is taken from Howard (1970).

[2] Strictly speaking, a probability belongs to a single person, not a collection of people. Nonetheless, in parts of this

To examine the Bayesian analysis of this problem, we need some notation. We denote a variable by an upper-case letter (e.g., X, Y, X_i, Θ), and the state or value of a corresponding variable by that same letter in lower case (e.g., x, y, x_i, θ). We denote a set of variables by a bold-face upper-case letter (e.g., $\mathbf{X}, \mathbf{Y}, \mathbf{X}_i$). We use a corresponding bold-face lower-case letter (e.g., $\mathbf{x}, \mathbf{y}, \mathbf{x}_i$) to denote an assignment of state or value to each variable in a given set. We say that variable set $\mathbf{X}$ is in *configuration* $\mathbf{x}$. We use $p(X = x|\xi)$ (or $p(x|\xi)$ as a shorthand) to denote the probability that $X = x$ of a person with state of information ξ. We also use $p(x|\xi)$ to denote the probability distribution for X (both mass functions and density functions). Whether $p(x|\xi)$ refers to a probability, a probability density, or a probability distribution will be clear from context. We use this notation for probability throughout the paper. A summary of all notation is given at the end of the chapter.

Returning to the thumbtack problem, we define Θ to be a variable[3] whose values θ correspond to the possible true values of the physical probability. We sometimes refer to θ as a *parameter*. We express the uncertainty about Θ using the probability density function $p(\theta|\xi)$. In addition, we use X_l to denote the variable representing the outcome of the lth flip, $l = 1, \ldots, N+1$, and $D = \{X_1 = x_1, \ldots, X_N = x_N\}$ to denote the set of our observations. Thus, in Bayesian terms, the thumbtack problem reduces to computing $p(x_{N+1}|D, \xi)$ from $p(\theta|\xi)$.

To do so, we first use Bayes' rule to obtain the probability distribution for Θ given D and background knowledge ξ:

$$p(\theta|D, \xi) = \frac{p(\theta|\xi)\; p(D|\theta, \xi)}{p(D|\xi)} \tag{1}$$

where

$$p(D|\xi) = \int p(D|\theta, \xi)\; p(\theta|\xi)\; d\theta \tag{2}$$

Next, we expand the term $p(D|\theta, \xi)$. Both Bayesians and classical statisticians agree on this term: it is the likelihood function for binomial sampling. In particular, given the value of Θ, the observations in D are mutually independent, and the probability of heads (tails) on any one observation is θ $(1-\theta)$. Consequently, Equation 1 becomes

$$p(\theta|D, \xi) = \frac{p(\theta|\xi)\; \theta^h\; (1-\theta)^t}{p(D|\xi)} \tag{3}$$

where h and t are the number of heads and tails observed in D, respectively. The probability distributions $p(\theta|\xi)$ and $p(\theta|D, \xi)$ are commonly referred to as the *prior* and *posterior* for Θ, respectively. The quantities h and t are said to be *sufficient statistics* for binomial sampling, because they provide a summarization of the data that is sufficient to compute the posterior from the prior. Finally, we average over the possible values of Θ (using the expansion rule of probability) to determine the probability that the $N+1$th toss of the thumbtack will come up heads:

$$\begin{aligned} p(X_{N+1} = heads|D, \xi) &= \int p(X_{N+1} = heads|\theta, \xi)\; p(\theta|D, \xi)\; d\theta \\ &= \int \theta\; p(\theta|D, \xi)\; d\theta \equiv \mathrm{E}_{p(\theta|D,\xi)}(\theta) \end{aligned} \tag{4}$$

discussion, we refer to "our" probability to avoid awkward English.

[3] Bayesians typically refer to Θ as an *uncertain variable*, because the value of Θ is uncertain. In contrast, classical statisticians often refer to Θ as a *random variable*. In this text, we refer to Θ and all uncertain/random variables simply as variables.

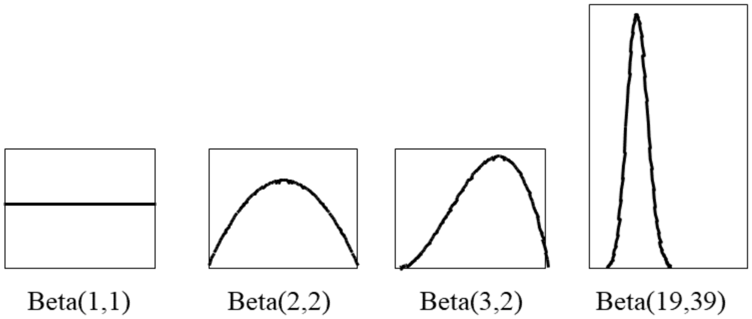


Figure 2: Several beta distributions.

where $\mathrm{E}_{p(\theta|D,\xi)}(\theta)$ denotes the expectation of θ with respect to the distribution $p(\theta|D,\xi)$.

To complete the Bayesian story for this example, we need a method to assess the prior distribution for Θ. A common approach, usually adopted for convenience, is to assume that this distribution is a *beta* distribution:

$$p(\theta|\xi) = \mathrm{Beta}(\theta|\alpha_h, \alpha_t) \equiv \frac{\Gamma(\alpha)}{\Gamma(\alpha_h)\Gamma(\alpha_t)} \theta^{\alpha_h - 1}(1-\theta)^{\alpha_t - 1} \tag{5}$$

where $\alpha_h > 0$ and $\alpha_t > 0$ are the parameters of the beta distribution, $\alpha = \alpha_h + \alpha_t$, and $\Gamma(\cdot)$ is the *Gamma* function which satisfies $\Gamma(x+1) = x\Gamma(x)$ and $\Gamma(1) = 1$. The quantities α_h and α_t are often referred to as *hyperparameters* to distinguish them from the parameter θ. The hyperparameters α_h and α_t must be greater than zero so that the distribution can be normalized. Examples of beta distributions are shown in Figure 2.

The beta prior is convenient for several reasons. By Equation 3, the posterior distribution will also be a beta distribution:

$$p(\theta|D,\xi) = \frac{\Gamma(\alpha + N)}{\Gamma(\alpha_h + h)\Gamma(\alpha_t + t)} \theta^{\alpha_h + h - 1}(1-\theta)^{\alpha_t + t - 1} = \mathrm{Beta}(\theta|\alpha_h + h, \alpha_t + t) \tag{6}$$

We say that the set of beta distributions is a *conjugate family of distributions* for binomial sampling. Also, the expectation of θ with respect to this distribution has a simple form:

$$\int \theta \ \mathrm{Beta}(\theta|\alpha_h, \alpha_t) \ d\theta = \frac{\alpha_h}{\alpha} \tag{7}$$

Hence, given a beta prior, we have a simple expression for the probability of heads in the $N+1$th toss:

$$p(X_{N+1} = heads|D,\xi) = \frac{\alpha_h + h}{\alpha + N} \tag{8}$$

Assuming $p(\theta|\xi)$ is a beta distribution, it can be assessed in a number of ways. For example, we can assess our probability for heads in the first toss of the thumbtack (e.g., using a probability wheel). Next, we can imagine having seen the outcomes of k flips, and reassess our probability for heads in the next toss. From Equation 8, we have (for $k = 1$)

$$p(X_1 = heads|\xi) = \frac{\alpha_h}{\alpha_h + \alpha_t} \qquad p(X_2 = heads|X_1 = heads, \xi) = \frac{\alpha_h + 1}{\alpha_h + \alpha_t + 1}$$

Given these probabilities, we can solve for α_h and α_t. This assessment technique is known as the method of *imagined future data*.

Another assessment method is based on Equation 6. This equation says that, if we start with a Beta(0, 0) prior[4] and observe α_h heads and α_t tails, then our posterior (i.e., new prior) will be a Beta(α_h, α_t) distribution. Recognizing that a Beta(0, 0) prior encodes a state of minimum information, we can assess α_h and α_t by determining the (possibly fractional) number of observations of heads and tails that is equivalent to our actual knowledge about flipping thumbtacks. Alternatively, we can assess $p(X_1 = \mathit{heads}|\xi)$ and α, which can be regarded as an *equivalent sample size* for our current knowledge. This technique is known as the method of *equivalent samples*. Other techniques for assessing beta distributions are discussed by Winkler (1967) and Chaloner and Duncan (1983).

Although the beta prior is convenient, it is not accurate for some problems. For example, suppose we think that the thumbtack may have been purchased at a magic shop. In this case, a more appropriate prior may be a mixture of beta distributions—for example,

$$p(\theta|\xi) = 0.4 \text{ Beta}(20, 1) + 0.4 \text{ Beta}(1, 20) + 0.2 \text{ Beta}(2, 2)$$

where 0.4 is our probability that the thumbtack is heavily weighted toward heads (tails). In effect, we have introduced an additional *hidden* or unobserved variable H, whose states correspond to the three possibilities: (1) thumbtack is biased toward heads, (2) thumbtack is biased toward tails, and (3) thumbtack is normal; and we have asserted that θ conditioned on each state of H is a beta distribution. In general, there are simple methods (e.g., the method of imagined future data) for determining whether or not a beta prior is an accurate reflection of one's beliefs. In those cases where the beta prior is inaccurate, an accurate prior can often be assessed by introducing additional hidden variables, as in this example.

So far, we have only considered observations drawn from a binomial distribution. In general, observations may be drawn from any physical probability distribution:

$$p(x|\boldsymbol{\theta}, \xi) = f(x, \boldsymbol{\theta})$$

where $f(x, \boldsymbol{\theta})$ is the likelihood function with parameters $\boldsymbol{\theta}$. For purposes of this discussion, we assume that the number of parameters is finite. As an example, X may be a continuous variable and have a Gaussian physical probability distribution with mean μ and variance v:

$$p(x|\boldsymbol{\theta}, \xi) = (2\pi v)^{-1/2} \; e^{-(x-\mu)^2/2v}$$

where $\boldsymbol{\theta} = \{\mu, v\}$.

Regardless of the functional form, we can learn about the parameters given data using the Bayesian approach. As we have done in the binomial case, we define variables corresponding to the unknown parameters, assign priors to these variables, and use Bayes' rule to update our beliefs about these parameters given data:

$$p(\boldsymbol{\theta}|D, \xi) = \frac{p(D|\boldsymbol{\theta}, \xi) \; p(\boldsymbol{\theta}|\xi)}{p(D|\xi)} \tag{9}$$

[4] Technically, the hyperparameters of this prior should be small positive numbers so that $p(\theta|\xi)$ can be normalized.

We then average over the possible values of Θ to make predictions. For example,

$$p(x_{N+1}|D,\xi) = \int p(x_{N+1}|\boldsymbol{\theta},\xi)\ p(\boldsymbol{\theta}|D,\xi)\ d\boldsymbol{\theta} \tag{10}$$

For a class of distributions known as the *exponential family*, these computations can be done efficiently and in closed form.[5] Members of this class include the binomial, multinomial, normal, Gamma, Poisson, and multivariate-normal distributions. Each member of this family has sufficient statistics that are of fixed dimension for any random sample, and a simple conjugate prior.[6] Bernardo and Smith (pp. 436–442, 1994) have compiled the important quantities and Bayesian computations for commonly used members of the exponential family. Here, we summarize these items for multinomial sampling, which we use to illustrate many of the ideas in this paper.

In multinomial sampling, the observed variable X is discrete, having r possible states $x^1,\ldots,x^r$. The likelihood function is given by

$$p(X = x^k|\boldsymbol{\theta},\xi) = \theta_k, \qquad k = 1,\ldots,r$$

where $\boldsymbol{\theta} = \{\theta_2,\ldots,\theta_r\}$ are the parameters. (The parameter θ_1 is given by $1 - \sum_{k=2}^r \theta_k$.) In this case, as in the case of binomial sampling, the parameters correspond to physical probabilities. The sufficient statistics for data set $D = \{X_1 = x_1,\ldots,X_N = x_N\}$ are $\{N_1,\ldots,N_r\}$, where N_i is the number of times $X = x^k$ in D. The simple conjugate prior used with multinomial sampling is the Dirichlet distribution:

$$p(\boldsymbol{\theta}|\xi) = \mathrm{Dir}(\boldsymbol{\theta}|\alpha_1,\ldots,\alpha_r) \equiv \frac{\Gamma(\alpha)}{\prod_{k=1}^r \Gamma(\alpha_k)} \prod_{k=1}^r \theta_k^{\alpha_k - 1} \tag{11}$$

where $\alpha = \sum_{i=1}^r \alpha_k$, and $\alpha_k > 0, k = 1,\ldots,r$. The posterior distribution $p(\boldsymbol{\theta}|D,\xi) = \mathrm{Dir}(\boldsymbol{\theta}|\alpha_1 + N_1,\ldots,\alpha_r + N_r)$. Techniques for assessing the beta distribution, including the methods of imagined future data and equivalent samples, can also be used to assess Dirichlet distributions. Given this conjugate prior and data set D, the probability distribution for the next observation is given by

$$p(X_{N+1} = x^k|D,\xi) = \int \theta_k\ \mathrm{Dir}(\boldsymbol{\theta}|\alpha_1 + N_1,\ldots,\alpha_r + N_r)\ d\boldsymbol{\theta} = \frac{\alpha_k + N_k}{\alpha + N} \tag{12}$$

As we shall see, another important quantity in Bayesian analysis is the *marginal likelihood* or *evidence* $p(D|\xi)$. In this case, we have

$$p(D|\xi) = \frac{\Gamma(\alpha)}{\Gamma(\alpha + N)} \cdot \prod_{k=1}^r \frac{\Gamma(\alpha_k + N_k)}{\Gamma(\alpha_k)} \tag{13}$$

We note that the explicit mention of the state of knowledge ξ is useful, because it reinforces the notion that probabilities are subjective. Nonetheless, once this concept is firmly in place, the notation simply adds clutter. In the remainder of this tutorial, we shall not mention ξ explicitly.

[5] Recent advances in Monte-Carlo methods have made it possible to work efficiently with many distributions outside the exponential family. See, for example, Gilks et al. (1996).

[6] In fact, except for a few, well-characterized exceptions, the exponential family is the only class of distributions that have sufficient statistics of fixed dimension (Koopman, 1936; Pitman, 1936).

In closing this section, we emphasize that, although the Bayesian and classical approaches may sometimes yield the same prediction, they are fundamentally different methods for learning from data. As an illustration, let us revisit the thumbtack problem. Here, the Bayesian "estimate" for the physical probability of heads is obtained in a manner that is essentially the opposite of the classical approach.

Namely, in the classical approach, θ is fixed (albeit unknown), and we imagine all data sets of size N that *may be* generated by sampling from the binomial distribution determined by θ. Each data set D will occur with some probability $p(D|\theta)$ and will produce an estimate $\theta^*(D)$. To evaluate an estimator, we compute the expectation and variance of the estimate with respect to all such data sets:

$$\begin{aligned} \mathrm{E}_{p(D|\theta)}(\theta^*) &= \sum_D p(D|\theta)\ \theta^*(D) \\ \mathrm{Var}_{p(D|\theta)}(\theta^*) &= \sum_D p(D|\theta)\ (\theta^*(D) - \mathrm{E}_{p(D|\theta)}(\theta^*))^2 \end{aligned} \qquad (14)$$

We then choose an estimator that somehow balances the bias $(\theta - \mathrm{E}_{p(D|\theta)}(\theta^*))$ and variance of these estimates over the possible values for θ.[7] Finally, we apply this estimator to the data set that we actually observe. A commonly-used estimator is the maximum-likelihood (ML) estimator, which selects the value of θ that maximizes the likelihood $p(D|\theta)$. For binomial sampling, we have

$$\theta^*_{\mathrm{ML}}(D) = \frac{N_k}{\sum_{k=1}^r N_k}$$

For this (and other types) of sampling, the ML estimator is *unbiased.* That is, for all values of θ, the ML estimator has zero bias. In addition, for all values of θ, the variance of the ML estimator is no greater than that of any other unbiased estimator (see, e.g., Schervish, 1995).

In contrast, in the Bayesian approach, D is fixed, and we imagine all possible values of θ from which this data set *could have been* generated. Given θ, the "estimate" of the physical probability of heads is just θ itself. Nonetheless, we are uncertain about θ, and so our final estimate is the expectation of θ with respect to our posterior beliefs about its value:

$$\mathrm{E}_{p(\theta|D,\xi)}(\theta) = \int \theta\ p(\theta|D,\xi)\ d\theta \qquad (15)$$

The expectations in Equations 14 and 15 are different and, in many cases, lead to different "estimates". One way to frame this difference is to say that the classical and Bayesian approaches have different definitions for what it means to be a good estimator. Both solutions are "correct" in that they are self consistent. Unfortunately, both methods have their drawbacks, which has lead to endless debates about the merit of each approach. For example, Bayesians argue that it does not make sense to consider the expectations in Equation 14, because we only see a single data set. If we saw more than one data set, we should combine them into one larger data set. In contrast, classical statisticians argue that sufficiently accurate priors can not be assessed in many situations. The common view that seems to be emerging is that one should use whatever method that is most

[7] Low bias and variance are not the only desirable properties of an estimator. Other desirable properties include consistency and robustness.

sensible for the task at hand. We share this view, although we also believe that the Bayesian approach has been under used, especially in light of its advantages mentioned in the introduction (points three and four). Consequently, in this paper, we concentrate on the Bayesian approach.

3 Bayesian Networks

So far, we have considered only simple problems with one or a few variables. In real learning problems, however, we are typically interested in looking for relationships among a large number of variables. The Bayesian network is a representation suited to this task. It is a graphical model that efficiently encodes the joint probability distribution (physical or Bayesian) for a large set of variables. In this section, we define a Bayesian network and show how one can be constructed from prior knowledge.

A Bayesian network for a set of variables $\mathbf{X} = \{X_1, \ldots, X_n\}$ consists of (1) a network structure S that encodes a set of conditional independence assertions about variables in $\mathbf{X}$, and (2) a set P of local probability distributions associated with each variable. Together, these components define the joint probability distribution for $\mathbf{X}$. The network structure S is a directed acyclic graph. The nodes in S are in one-to-one correspondence with the variables $\mathbf{X}$. We use X_i to denote both the variable and its corresponding node, and $\mathbf{Pa}_i$ to denote the parents of node X_i in S as well as the variables corresponding to those parents. The *lack* of possible arcs in S encode conditional independencies. In particular, given structure S, the joint probability distribution for $\mathbf{X}$ is given by

$$p(\mathbf{x}) = \prod_{i=1}^{n} p(x_i|\mathbf{pa}_i) \tag{16}$$

The local probability distributions P are the distributions corresponding to the terms in the product of Equation 16. Consequently, the pair (S, P) encodes the joint distribution $p(\mathbf{x})$.

The probabilities encoded by a Bayesian network may be Bayesian or physical. When building Bayesian networks from prior knowledge alone, the probabilities will be Bayesian. When learning these networks from data, the probabilities will be physical (and their values may be uncertain). In subsequent sections, we describe how we can learn the structure and probabilities of a Bayesian network from data. In the remainder of this section, we explore the construction of Bayesian networks from prior knowledge. As we shall see in Section 10, this procedure can be useful in learning Bayesian networks as well.

To illustrate the process of building a Bayesian network, consider the problem of detecting credit-card fraud. We begin by determining the variables to model. One possible choice of variables for our problem is *Fraud* (F), *Gas* (G), *Jewelry* (J), *Age* (A), and *Sex* (S), representing whether or not the current purchase is fraudulent, whether or not there was a gas purchase in the last 24 hours, whether or not there was a jewelry purchase in the last 24 hours, and the age and sex of the card holder, respectively. The states of these variables are shown in Figure 3. Of course, in a realistic problem, we would include many more variables. Also, we could model the states of one or more of these variables at a finer level of detail. For example, we could let *Age* be a continuous variable.

This initial task is not always straightforward. As part of this task we must (1) correctly identify the goals of modeling (e.g., prediction versus explanation versus exploration), (2) identify many possible observations that may be relevant to the problem, (3) determine what subset of those observations is worthwhile to model, and (4) organize the observations into variables having mutually exclusive and collectively exhaustive states. Difficulties here are not unique to modeling with Bayesian networks, but rather are common to most approaches. Although there are no clean solutions, some guidance is offered by decision analysts (e.g., Howard and Matheson, 1983) and (when data are available) statisticians (e.g., Tukey, 1977).

In the next phase of Bayesian-network construction, we build a directed acyclic graph that encodes assertions of conditional independence. One approach for doing so is based on the following observations. From the chain rule of probability, we have

$$p(\mathbf{x}) = \prod_{i=1}^{n} p(x_i | x_1, \ldots, x_{i-1}) \tag{17}$$

Now, for every X_i, there will be some subset $\Pi_i \subseteq \{X_1, \ldots, X_{i-1}\}$ such that X_i and $\{X_1, \ldots, X_{i-1}\} \setminus \Pi_i$ are conditionally independent given Π_i. That is, for any $\mathbf{x}$,

$$p(x_i | x_1, \ldots, x_{i-1}) = p(x_i | \pi_i) \tag{18}$$

Combining Equations 17 and 18, we obtain

$$p(\mathbf{x}) = \prod_{i=1}^{n} p(x_i | \pi_i) \tag{19}$$

Comparing Equations 16 and 19, we see that the variables sets $(\Pi_1, \ldots, \Pi_n)$ correspond to the Bayesian-network parents $(\mathbf{Pa}_1, \ldots, \mathbf{Pa}_n)$, which in turn fully specify the arcs in the network structure S.

Consequently, to determine the structure of a Bayesian network we (1) order the variables somehow, and (2) determine the variables sets that satisfy Equation 18 for $i = 1, \ldots, n$. In our example, using the ordering (F, A, S, G, J), we have the conditional independencies

$$\begin{aligned} p(a|f) &= p(a) \\ p(s|f,a) &= p(s) \\ p(g|f,a,s) &= p(g|f) \\ p(j|f,a,s,g) &= p(j|f,a,s) \end{aligned} \tag{20}$$

Thus, we obtain the structure shown in Figure 3.

This approach has a serious drawback. If we choose the variable order carelessly, the resulting network structure may fail to reveal many conditional independencies among the variables. For example, if we construct a Bayesian network for the fraud problem using the ordering (J, G, S, A, F), we obtain a fully connected network structure. Thus, in the worst case, we have to explore $n!$ variable orderings to find the best one. Fortunately, there is another technique for constructing Bayesian networks that does not require an ordering. The approach is based on two observations: (1) people can often readily assert causal relationships among variables, and (2) causal relationships typically

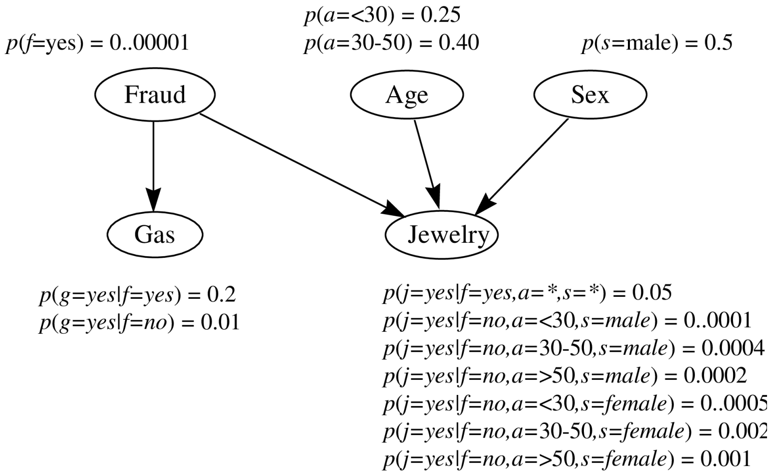

p(f=yes) = 0..00001
p(a=<30) = 0.25
p(a=30-50) = 0.40
p(s=male) = 0.5
Fraud
Age
Sex
Gas
Jewelry
p(g=yes|f=yes) = 0.2
p(g=yes|f=no) = 0.01
p(j=yes|f=yes,a=*,s=*) = 0.05
p(j=yes|f=no,a=<30,s=male) = 0..0001
p(j=yes|f=no,a=30-50,s=male) = 0.0004
p(j=yes|f=no,a=>50,s=male) = 0.0002
p(j=yes|f=no,a=<30,s=female) = 0..0005
p(j=yes|f=no,a=30-50,s=female) = 0.002
p(j=yes|f=no,a=>50,s=female) = 0.001

Figure 3: A Bayesian-network for detecting credit-card fraud. Arcs are drawn from cause to effect. The local probability distribution(s) associated with a node are shown adjacent to the node. An asterisk is a shorthand for "any state."

correspond to assertions of conditional dependence. In particular, to construct a Bayesian network for a given set of variables, we simply draw arcs from cause variables to their immediate effects. In almost all cases, doing so results in a network structure that satisfies the definition *Equation* 16. For example, given the assertions that *Fraud* is a direct cause of *Gas*, and *Fraud*, *Age*, and *Sex* are direct causes of *Jewelry*, we obtain the network structure in Figure 3. The causal semantics of Bayesian networks are in large part responsible for the success of Bayesian networks as a representation for expert systems (Heckerman *et al.*, 1995a). In Section 15, we will see how to learn causal relationships from data using these causal semantics.

In the final step of constructing a Bayesian network, we assess the local probability distribution(s) $p(x_i|\mathbf{pa}_i)$. In our fraud example, where all variables are discrete, we assess one distribution for X_i for every configuration of $\mathbf{Pa}_i$. Example distributions are shown in Figure 3.

Note that, although we have described these construction steps as a simple sequence, they are often intermingled in practice. For example, judgments of conditional independence and/or cause and effect can influence problem formulation. Also, assessments of probability can lead to changes in the network structure. Exercises that help one gain familiarity with the practice of building Bayesian networks can be found in Jensen (1996).

4 Inference in a Bayesian Network

Once we have constructed a Bayesian network (from prior knowledge, data, or a combination), we usually need to determine various probabilities of interest from the model. For example, in our problem concerning fraud detection, we want to know the probability of fraud given observations of the other variables. This probability is not stored directly in the model, and hence needs to be computed. In general, the computation of a probability of interest given a model is known as *probabilistic inference*. In this section we describe probabilistic inference in Bayesian networks.

Because a Bayesian network for $\mathbf{X}$ determines a joint probability distribution for $\mathbf{X}$, we can—in principle—use the Bayesian network to compute any probability of interest. For example, from the Bayesian network in Figure 3, the probability of fraud given observations of the other variables can be computed as follows:

$$p(f|a,s,g,j) = \frac{p(f,a,s,g,j)}{p(a,s,g,j)} = \frac{p(f,a,s,g,j)}{\sum_{f'} p(f',a,s,g,j)} \tag{21}$$

For problems with many variables, however, this direct approach is not practical. Fortunately, at least when all variables are discrete, we can exploit the conditional independencies encoded in a Bayesian network to make this computation more efficient. In our example, given the conditional independencies in Equation 20, Equation 21 becomes

$$\begin{aligned} p(f|a,s,g,j) &= \frac{p(f)p(a)p(s)p(g|f)p(j|f,a,s)}{\sum_{f'} p(f')p(a)p(s)p(g|f')p(j|f',a,s)} \\ &= \frac{p(f)p(g|f)p(j|f,a,s)}{\sum_{f'} p(f')p(g|f')p(j|f',a,s)} \end{aligned} \tag{22}$$

Several researchers have developed probabilistic inference algorithms for Bayesian networks with discrete variables that exploit conditional independence roughly as we have described, although with different twists. For example, Howard and Matheson (1981), Olmsted (1983), and Shachter (1988) developed an algorithm that reverses arcs in the network structure until the answer to the given probabilistic query can be read directly from the graph. In this algorithm, each arc reversal corresponds to an application of Bayes' theorem. Pearl (1986) developed a message-passing scheme that updates the probability distributions for each node in a Bayesian network in response to observations of one or more variables. Lauritzen and Spiegelhalter (1988), Jensen et al. (1990), and Dawid (1992) created an algorithm that first transforms the Bayesian network into a tree where each node in the tree corresponds to a subset of variables in $\mathbf{X}$. The algorithm then exploits several mathematical properties of this tree to perform probabilistic inference. Most recently, D'Ambrosio (1991) developed an inference algorithm that simplifies sums and products symbolically, as in the transformation from Equation 21 to 22. The most commonly used algorithm for discrete variables is that of Lauritzen and Spiegelhalter (1988), Jensen et al (1990), and Dawid (1992).

Methods for exact inference in Bayesian networks that encode multivariate-Gaussian or Gaussian-mixture distributions have been developed by Shachter and Kenley (1989) and Lauritzen (1992), respectively. These methods also use assertions of conditional independence to simplify inference. Approximate methods for inference in Bayesian networks with other distributions, such as the generalized linear-regression model, have also been developed (Saul *et al.*, 1996; Jaakkola and Jordan, 1996).

Although we use conditional independence to simplify probabilistic inference, exact inference in an arbitrary Bayesian network for discrete variables is NP-hard (Cooper, 1990). Even approximate inference (for example, Monte-Carlo methods) is NP-hard (Dagum and Luby, 1993). The source of the difficulty lies in undirected cycles in the Bayesian-network structure—cycles in the structure where we ignore the directionality of the arcs. (If we add an arc from *Age* to *Gas* in the network structure of Figure 3, then we obtain a structure with one undirected cycle: $F - G - A - J - F$.) When a Bayesian-network structure contains many undirected cycles, inference is intractable. For many applications, however, structures are simple enough (or can be simplified sufficiently without sacrificing much accuracy) so that inference is efficient. For those applications where generic inference methods are impractical, researchers are developing techniques that are custom tailored to particular network topologies (Heckerman 1989; Suermondt and Cooper, 1991; Saul *et al.*, 1996; Jaakkola and Jordan, 1996) or to particular inference queries (Ramamurthi and Agogino, 1988; Shachter et al., 1990; Jensen and Andersen, 1990; Darwiche and Provan, 1996).

5 Learning Probabilities in a Bayesian Network

In the next several sections, we show how to refine the structure and local probability distributions of a Bayesian network given data. The result is set of techniques for data analysis that combines prior knowledge with data to produce improved knowledge. In this section, we consider the simplest version of this problem: using data to update the probabilities of a given Bayesian network structure.

Recall that, in the thumbtack problem, we do not learn the probability of heads. Instead, we update our posterior distribution for the variable that represents the physical probability of heads. We follow the same approach for probabilities in a Bayesian network. In particular, we assume—perhaps from causal knowledge about the problem—that the physical joint probability distribution for $\mathbf{X}$ can be encoded in some network structure S. We write

$$p(\mathbf{x}|\boldsymbol{\theta}_s, S^h) = \prod_{i=1}^{n} p(x_i|\mathbf{pa}_i, \boldsymbol{\theta}_i, S^h) \tag{23}$$

where $\boldsymbol{\theta}_i$ is the vector of parameters for the distribution $p(x_i|\mathbf{pa}_i, \boldsymbol{\theta}_i, S^h)$, $\boldsymbol{\theta}_s$ is the vector of parameters $(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n)$, and S^h denotes the event (or "hypothesis" in statistics nomenclature) that the physical joint probability distribution can be factored according to S.[8] In addition, we assume that we have a random sample $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ from the physical joint probability distribution of $\mathbf{X}$. We refer to an element $\mathbf{x}_l$ of D as a *case*. As in Section 2, we encode our uncertainty about the parameters $\boldsymbol{\theta}_s$ by defining a (vector-valued) variable Θ_s, and assessing a prior probability density function $p(\boldsymbol{\theta}_s|S^h)$. The problem of learning probabilities in a Bayesian network can now be stated simply: Given a random sample D, compute the posterior distribution $p(\boldsymbol{\theta}_s|D, S^h)$.

We refer to the distribution $p(x_i|\mathbf{pa}_i, \boldsymbol{\theta}_i, S^h)$, viewed as a function of $\boldsymbol{\theta}_i$, as a *local distribution function*. Readers familiar with methods for supervised learning will recognize that a local distribution function is nothing more than a probabilistic classification or regression function. Thus, a Bayesian network can be viewed as a collection of probabilistic classification/regression models, organized by conditional-independence relationships. Examples of classification/regression models

[8] As defined here, network-structure hypotheses overlap. For example, given $\mathbf{X} = \{X_1, X_2\}$, any joint distribution for $\mathbf{X}$ that can be factored according the network structure containing no arc, can also be factored according to the network structure $X_1 \longrightarrow X_2$. Such overlap presents problems for model averaging, described in Section 7. Therefore, we should add conditions to the definition to insure no overlap. Heckerman and Geiger (1996) describe one such set of conditions.

that produce probabilistic outputs include linear regression, generalized linear regression, probabilistic neural networks (e.g., MacKay, 1992a, 1992b), probabilistic decision trees (e.g., Buntine, 1993; Friedman and Goldszmidt, 1996), kernel density estimation methods (Book, 1994), and dictionary methods (Friedman, 1995). In principle, any of these forms can be used to learn probabilities in a Bayesian network; and, in most cases, Bayesian techniques for learning are available. Nonetheless, the most studied models include the unrestricted multinomial distribution (e.g., Cooper and Herskovits, 1992), linear regression with Gaussian noise (e.g., Buntine, 1994; Heckerman and Geiger, 1996), and generalized linear regression (e.g., MacKay, 1992a and 1992b; Neal, 1993; and Saul *et al.*, 1996).

In this tutorial, we illustrate the basic ideas for learning probabilities (and structure) using the unrestricted multinomial distribution. In this case, each variable $X_i \in \mathbf{X}$ is discrete, having r_i possible values $x_i^1, \ldots, x_i^{r_i}$, and each local distribution function is collection of multinomial distributions, one distribution for each configuration of $\mathbf{Pa}_i$. Namely, we assume

$$p(x_i^k | \mathbf{pa}_i^j, \boldsymbol{\theta}_i, S^h) = \theta_{ijk} > 0 \tag{24}$$

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 $1 - \sum_{k=2}^{r_i} \theta_{ijk}$.) For convenience, we define the vector of parameters

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

for all i and j. We use the term "unrestricted" to contrast this distribution with multinomial distributions that are low-dimensional functions of $\mathbf{Pa}_i$—for example, the generalized linear-regression model.

Given this class of local distribution functions, we can compute the posterior distribution $p(\boldsymbol{\theta}_s|D, S^h)$ efficiently and in closed form under two assumptions. The first assumption is that there are no missing data in the random sample D. We say that the random sample D is *complete.* The second assumption is that the parameter vectors $\boldsymbol{\theta}_{ij}$ are mutually independent.[9] That is,

$$p(\boldsymbol{\theta}_s|S^h) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} p(\boldsymbol{\theta}_{ij}|S^h)$$

We refer to this assumption, which was introduced by Spiegelhalter and Lauritzen (1990), as *parameter independence.*

Given that the joint physical probability distribution factors according to some network structure S, the assumption of parameter independence can itself be represented by a larger Bayesian-network structure. For example, the network structure in Figure 4 represents the assumption of parameter independence for $\mathbf{X} = \{X, Y\}$ (X, Y binary) and the hypothesis that the network structure $X \to Y$ encodes the physical joint probability distribution for $\mathbf{X}$.

Under the assumptions of complete data and parameter independence, the parameters remain independent given a random sample:

$$p(\boldsymbol{\theta}_s|D, S^h) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} p(\boldsymbol{\theta}_{ij}|D, S^h) \tag{25}$$

[9] The computation is also straightforward if two or more parameters are equal. For details, see Thiesson (1995).

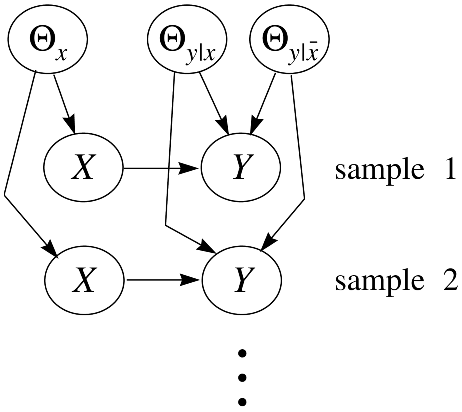


Figure 4: A Bayesian-network structure depicting the assumption of parameter independence for learning the parameters of the network structure $X \rightarrow Y$. Both variables X and Y are binary. We use x and $\bar{x}$ to denote the two states of X, and y and $\bar{y}$ to denote the two states of Y.

Thus, we can update each vector of parameters $\boldsymbol{\theta}_{ij}$ independently, just as in the one-variable case. Assuming each vector $\boldsymbol{\theta}_{ij}$ has the prior distribution $\text{Dir}(\boldsymbol{\theta}_{ij}|\alpha_{ij1}, \ldots, \alpha_{ijr_i})$, we obtain the posterior distribution

$$p(\boldsymbol{\theta}_{ij}|D, S^h) = \text{Dir}(\boldsymbol{\theta}_{ij}|\alpha_{ij1} + N_{ij1}, \ldots, \alpha_{ijr_i} + N_{ijr_i}) \tag{26}$$

where N_{ijk} is the number of cases in D in which $X_i = x_i^k$ and $\mathbf{Pa}_i = \mathbf{pa}_i^j$.

As in the thumbtack example, we can average over the possible configurations of $\boldsymbol{\theta}_s$ to obtain predictions of interest. For example, let us compute $p(\mathbf{x}_{N+1}|D, S^h)$, where $\mathbf{x}_{N+1}$ is the next case to be seen after D. Suppose that, in case $\mathbf{x}_{N+1}$, $X_i = x_i^k$ and $\mathbf{Pa}_i = \mathbf{pa}_i^j$, where k and j depend on i. Thus,

$$p(\mathbf{x}_{N+1}|D, S^h) = \text{E}_{p(\boldsymbol{\theta}_s|D,S^h)}\left(\prod_{i=1}^{n} \theta_{ijk}\right)$$

To compute this expectation, we first use the fact that the parameters remain independent given D:

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

Then, we use Equation 12 to obtain

$$p(\mathbf{x}_{N+1}|D, S^h) = \prod_{i=1}^{n} \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}} \tag{27}$$

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

These computations are simple because the unrestricted multinomial distributions are in the exponential family. Computations for linear regression with Gaussian noise are equally straightforward (Buntine, 1994; Heckerman and Geiger, 1996).

6 Methods for Incomplete Data

Let us now discuss methods for learning about parameters when the random sample is incomplete (i.e., some variables in some cases are not observed). An important distinction concerning missing data is whether or not the absence of an observation is dependent on the actual states of the variables. For example, a missing datum in a drug study may indicate that a patient became too sick—perhaps due to the side effects of the drug—to continue in the study. In contrast, if a variable is hidden (i.e., never observed in any case), then the absence of this data is independent of state. Although Bayesian methods and graphical models are suited to the analysis of both situations, methods for handling missing data where absence is independent of state are simpler than those where absence and state are dependent. In this tutorial, we concentrate on the simpler situation only. Readers interested in the more complicated case should see Rubin (1978), Robins (1986), and Pearl (1995).

Continuing with our example using unrestricted multinomial distributions, suppose we observe a single incomplete case. Let $\mathbf{Y} \subset \mathbf{X}$ and $\mathbf{Z} \subset \mathbf{X}$ denote the observed and unobserved variables in the case, respectively. Under the assumption of parameter independence, we can compute the posterior distribution of $\boldsymbol{\theta}_{ij}$ for network structure S as follows:

$$\begin{aligned} p(\boldsymbol{\theta}_{ij}|\mathbf{y}, S^h) &= \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{y}, S^h)\ p(\boldsymbol{\theta}_{ij}|\mathbf{y}, \mathbf{z}, S^h) \qquad (28) \\ &= (1 - p(\mathbf{pa}_i^j|\mathbf{y}, S^h))\ \{p(\boldsymbol{\theta}_{ij}|S^h)\} + \sum_{k=1}^{r_i} p(x_i^k, \mathbf{pa}_i^j|\mathbf{y}, S^h)\ \left\{p(\boldsymbol{\theta}_{ij}|x_i^k, \mathbf{pa}_i^j, S^h)\right\} \end{aligned}$$

(See Spiegelhalter and Lauritzen (1990) for a derivation.) Each term in curly brackets in Equation 28 is a Dirichlet distribution. Thus, unless both X_i and all the variables in $\mathbf{Pa}_i$ are observed in case $\mathbf{y}$, the posterior distribution of $\boldsymbol{\theta}_{ij}$ will be a linear combination of Dirichlet distributions—that is, a Dirichlet mixture with mixing coefficients $(1 - p(\mathbf{pa}_i^j|\mathbf{y}, S^h))$ and $p(x_i^k, \mathbf{pa}_i^j|\mathbf{y}, S^h), k = 1, \ldots, r_i$.

When we observe a second incomplete case, some or all of the Dirichlet components in Equation 28 will again split into Dirichlet mixtures. That is, the posterior distribution for $\boldsymbol{\theta}_{ij}$ we become a mixture of Dirichlet mixtures. As we continue to observe incomplete cases, each missing values for $\mathbf{Z}$, the posterior distribution for $\boldsymbol{\theta}_{ij}$ will contain a number of components that is exponential in the number of cases. In general, for any interesting set of local likelihoods and priors, the exact computation of the posterior distribution for $\boldsymbol{\theta}_s$ will be intractable. Thus, we require an approximation for incomplete data.

6.1 Monte-Carlo Methods

One class of approximations is based on Monte-Carlo or sampling methods. These approximations can be extremely accurate, provided one is willing to wait long enough for the computations to converge.

In this section, we discuss one of many Monte-Carlo methods known as *Gibbs sampling*, introduced by Geman and Geman (1984). 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 $p(\mathbf{x})$ 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*: 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. Introductions to Gibbs sampling and other Monte-Carlo methods—including methods for initialization and a discussion of convergence—are given by Neal (1993) and Madigan and York (1995).

To illustrate Gibbs sampling, let us approximate the probability density $p(\boldsymbol{\theta}_s|D, S^h)$ for some particular configuration of $\boldsymbol{\theta}_s$, given an incomplete data set $D = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ and a Bayesian network for discrete variables with independent Dirichlet priors. To approximate $p(\boldsymbol{\theta}_s|D, S^h)$, we first initialize the states of the unobserved variables in each case somehow. As a result, we have a complete random sample 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

$$p(x'_{il}|D_c \setminus x_{il}, S^h) = \frac{p(x'_{il}, D_c \setminus x_{il}|S^h)}{\sum_{x''_{il}} p(x''_{il}, D_c \setminus x_{il}|S^h)}$$

where $D_c \setminus x_{il}$ denotes the data set D_c with observation x_{il} removed, and the sum in the denominator runs over all states of variable X_{il}. As we shall see in Section 7, the terms in the numerator and denominator can be computed efficiently (see Equation 35). Third, we repeat this reassignment for all unobserved variables in D, producing a new complete random sample D'_c. Fourth, we compute the posterior density $p(\boldsymbol{\theta}_s|D'_c, S^h)$ as described in Equations 25 and 26. Finally, we iterate the previous three steps, and use the average of $p(\boldsymbol{\theta}_s|D'_c, S^h)$ as our approximation.

6.2 The Gaussian Approximation

Monte-Carlo methods yield accurate results, but they are often intractable—for example, when the sample size is large. Another approximation that is more efficient than Monte-Carlo methods and often accurate for relatively large samples is the *Gaussian approximation* (e.g., Kass *et al.*, 1988; Kass and Raftery, 1995).

The idea behind this approximation is that, for large amounts of data, $p(\boldsymbol{\theta}_s|D, S^h) \propto p(D|\boldsymbol{\theta}_s, S^h) \cdot p(\boldsymbol{\theta}_s|S^h)$ can often be approximated as a multivariate-Gaussian distribution. In particular, let

$$g(\boldsymbol{\theta}_s) \equiv \log(p(D|\boldsymbol{\theta}_s, S^h) \cdot p(\boldsymbol{\theta}_s|S^h)) \tag{29}$$

Also, define $\tilde{\boldsymbol{\theta}_s}$ to be the configuration of $\boldsymbol{\theta}_s$ that maximizes $g(\boldsymbol{\theta}_s)$. This configuration also maximizes $p(\boldsymbol{\theta}_s|D, S^h)$, and is known as the *maximum a posteriori* (MAP) configuration of $\boldsymbol{\theta}_s$. Using a second degree Taylor polynomial of $g(\boldsymbol{\theta}_s)$ about the $\tilde{\boldsymbol{\theta}}_s$ to approximate $g(\boldsymbol{\theta}_s)$, we obtain

$$g(\boldsymbol{\theta}_s) \approx g(\tilde{\boldsymbol{\theta}}_s) - \frac{1}{2}(\boldsymbol{\theta}_s - \tilde{\boldsymbol{\theta}}_s)A(\boldsymbol{\theta}_s - \tilde{\boldsymbol{\theta}}_s)^t \tag{30}$$

where $(\boldsymbol{\theta}_s - \tilde{\boldsymbol{\theta}}_s)^t$ is the transpose of row vector $(\boldsymbol{\theta}_s - \tilde{\boldsymbol{\theta}}_s)$, and A is the negative Hessian of $g(\boldsymbol{\theta}_s)$ evaluated at $\tilde{\boldsymbol{\theta}}_s$. Raising $g(\boldsymbol{\theta}_s)$ to the power of e and using Equation 29, we obtain

$$\begin{aligned} p(\boldsymbol{\theta}_s|D, S^h) &\propto p(D|\boldsymbol{\theta}_s, S^h)\ p(\boldsymbol{\theta}_s|S^h) \\ &\approx p(D|\tilde{\boldsymbol{\theta}}_s, S^h)\ p(\tilde{\boldsymbol{\theta}}_s|S^h)\ \exp\{-\frac{1}{2}(\boldsymbol{\theta}_s - \tilde{\boldsymbol{\theta}}_s)A(\boldsymbol{\theta}_s - \tilde{\boldsymbol{\theta}}_s)^t\} \end{aligned} \tag{31}$$

Hence, $p(\boldsymbol{\theta}_s|D, S^h)$ is approximately Gaussian.

To compute the Gaussian approximation, we must compute $\tilde{\boldsymbol{\theta}}_s$ as well as the negative Hessian of $g(\boldsymbol{\theta}_s)$ evaluated at $\tilde{\boldsymbol{\theta}}_s$. In the following section, we discuss methods for finding $\tilde{\boldsymbol{\theta}}_s$. Meng and Rubin (1991) describe a numerical technique for computing the second derivatives. Raftery (1995) shows how to approximate the Hessian using likelihood-ratio tests that are available in many statistical packages. Thiesson (1995) demonstrates that, for unrestricted multinomial distributions, the second derivatives can be computed using Bayesian-network inference.

6.3 The MAP and ML Approximations and the EM Algorithm

As the sample size of the data increases, the Gaussian peak will become sharper, tending to a delta function at the MAP configuration $\tilde{\boldsymbol{\theta}}_s$. In this limit, we do not need to compute averages or expectations. Instead, we simply make predictions based on the MAP configuration.

A further approximation is based on the observation that, as the sample size increases, the effect of the prior $p(\boldsymbol{\theta}_s|S^h)$ diminishes. Thus, we can approximate $\tilde{\boldsymbol{\theta}}_s$ by the maximum *maximum likelihood* (ML) configuration of $\boldsymbol{\theta}_s$:

$$\hat{\boldsymbol{\theta}}_s = \arg\max_{\boldsymbol{\theta}_s} \{p(D|\boldsymbol{\theta}_s, S^h)\}$$

One class of techniques for finding a ML or MAP is gradient-based optimization. For example, we can use gradient ascent, where we follow the derivatives of $g(\boldsymbol{\theta}_s)$ or the likelihood $p(D|\boldsymbol{\theta}_s, S^h)$ to a local maximum. Russell *et al.* (1995) and Thiesson (1995) show how to compute the derivatives of the likelihood for a Bayesian network with unrestricted multinomial distributions. Buntine (1994) discusses the more general case where the likelihood function comes from the exponential family. Of course, these gradient-based methods find only local maxima.

Another technique for finding a local ML or MAP is the expectation–maximization (EM) algorithm (Dempster *et al.*, 1977). To find a local MAP or ML, we begin by assigning a configuration to $\boldsymbol{\theta}_s$ somehow (e.g., at random). Next, we compute the *expected sufficient statistics* for a complete data set, where expectation is taken with respect to the joint distribution for $\mathbf{X}$ conditioned on the assigned configuration of $\boldsymbol{\theta}_s$ and the known data D. In our discrete example, we compute

$$\mathrm{E}_{p(\mathbf{x}|D,\boldsymbol{\theta}_s,S^h)}(N_{ijk}) = \sum_{l=1}^{N} p(x_i^k, \mathbf{pa}_i^j|\mathbf{y}_l, \boldsymbol{\theta}_s, S^h) \tag{32}$$

where $\mathbf{y}_l$ is the possibly incomplete lth case in D. When X_i and all the variables in $\mathbf{Pa}_i$ are observed in case $\mathbf{x}_l$, the term for this case requires a trivial computation: it is either zero or one. Otherwise, we can use any Bayesian network inference algorithm to evaluate the term. This computation is called the *expectation step* of the EM algorithm.

Next, we use the expected sufficient statistics as if they were actual sufficient statistics from a complete random sample D_c. If we are doing an ML calculation, then we determine the configuration of $\boldsymbol{\theta}_s$ that maximize $p(D_c|\boldsymbol{\theta}_s, S^h)$. In our discrete example, we have

$$\theta_{ijk} = \frac{\mathrm{E}_{p(\mathbf{x}|D,\boldsymbol{\theta}_s,S^h)}(N_{ijk})}{\sum_{k=1}^{r_i} \mathrm{E}_{p(\mathbf{x}|D,\boldsymbol{\theta}_s,S^h)}(N_{ijk})}$$

If we are doing a MAP calculation, then we determine the configuration of $\boldsymbol{\theta}_s$ that maximizes $p(\boldsymbol{\theta}_s|D_c, S^h)$. In our discrete example, we have[10]

$$\theta_{ijk} = \frac{\alpha_{ijk} + \mathrm{E}_{p(\mathbf{x}|D,\boldsymbol{\theta}_s,S^h)}(N_{ijk})}{\sum_{k=1}^{r_i} (\alpha_{ijk} + \mathrm{E}_{p(\mathbf{x}|D,\boldsymbol{\theta}_s,S^h)}(N_{ijk}))}$$

This assignment is called the *maximization step* of the EM algorithm. Dempster *et al.* (1977) showed that, under certain regularity conditions, iteration of the expectation and maximization steps will converge to a local maximum. The EM algorithm is typically applied when sufficient statistics exist (i.e., when local distribution functions are in the exponential family), although generalizations of the EM algroithm have been used for more complicated local distributions (see, e.g., Saul et al. 1996).

7 Learning Parameters and Structure

Now we consider the problem of learning about both the structure and probabilities of a Bayesian network given data.

Assuming we think structure can be improved, we must be uncertain about the network structure that encodes the physical joint probability distribution for $\mathbf{X}$. Following the Bayesian approach, we encode this uncertainty by defining a (discrete) variable whose states correspond to the possible network-structure hypotheses S^h, and assessing the probabilities $p(S^h)$. Then, given a random sample D from the physical probability distribution for $\mathbf{X}$, we compute the posterior distribution $p(S^h|D)$ and the posterior distributions $p(\boldsymbol{\theta}_s|D, S^h)$, and use these distributions in turn to compute expectations of interest. For example, to predict the next case after seeing D, we compute

$$p(\mathbf{x}_{N+1}|D) = \sum_{S^h} p(S^h|D) \int p(\mathbf{x}_{N+1}|\boldsymbol{\theta}_s, S^h)\ p(\boldsymbol{\theta}_s|D, S^h)\ d\boldsymbol{\theta}_s \tag{33}$$

In performing the sum, we assume that the network-structure hypotheses are mutually exclusive. We return to this point in Section 9.

[10] The MAP configuration $\tilde{\boldsymbol{\theta}}_s$ depends on the coordinate system in which the parameter variables are expressed. The expression for the MAP configuration given here is obtained by the following procedure. First, we transform each variable set $\boldsymbol{\theta}_{ij} = (\theta_{ij2}, \ldots, \theta_{ijr_i})$ to the new coordinate system $\phi_{ij} = (\phi_{ij2}, \ldots, \phi_{ijr_i})$, where $\phi_{ijk} = \log(\theta_{ijk}/\theta_{ij1}), k = 2, \ldots, r_i$. This coordinate system, which we denote by ϕ_s, is sometimes referred to as the *canonical* coordinate system for the multinomial distribution (see, e.g., Bernardo and Smith, 1994, pp. 199–202). Next, we determine the configuration of ϕ_s that maximizes $p(\phi_s|D_c, S^h)$. Finally, we transform this MAP configuration to the original coordinate system. Using the MAP configuration corresponding to the coordinate system ϕ_s has several advantages, which are discussed in Thiesson (1995b) and MacKay (1996).

The computation of $p(\boldsymbol{\theta}_s|D, S^h)$ is as we have described in the previous two sections. The computation of $p(S^h|D)$ is also straightforward, at least in principle. From Bayes' theorem, we have

$$p(S^h|D) = p(S^h)\ p(D|S^h)/p(D) \tag{34}$$

where $p(D)$ is a normalization constant that does not depend upon structure. Thus, to determine the posterior distribution for network structures, we need to compute the marginal likelihood of the data ($p(D|S^h)$) for each possible structure.

We discuss the computation of marginal likelihoods in detail in Section 9. As an introduction, consider our example with unrestricted multinomial distributions, parameter independence, Dirichlet priors, and complete data. As we have discussed, when there are no missing data, each parameter vector $\boldsymbol{\theta}_{ij}$ is updated independently. In effect, we have a separate multi-sided thumbtack problem for every i and j. Consequently, the marginal likelihood of the data is the just the product of the marginal likelihoods for each i–j pair (given by Equation 13):

$$p(D|S^h) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \cdot \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \tag{35}$$

This formula was first derived by Cooper and Herskovits (1992).

Unfortunately, the full Bayesian approach that we have described is often impractical. One important computation bottleneck is produced by the average over models in Equation 33. If we consider Bayesian-network models with n variables, the number of possible structure hypotheses is more than exponential in n. Consequently, in situations where the user can not exclude almost all of these hypotheses, the approach is intractable.

Statisticians, who have been confronted by this problem for decades in the context of other types of models, use two approaches to address this problem: *model selection* and *selective model averaging*. The former approach is to select a "good" model (i.e., structure hypothesis) from among all possible models, and use it as if it were the correct 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. These related approaches raise several important questions. In particular, do these approaches yield accurate results when applied to Bayesian-network structures? If so, how do we search for good models? And how do we decide whether or not a model is "good"?

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 (Cooper and Herskovits 1992; Aliferis and Cooper 1994; Heckerman *et al.*, 1995b) and that 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 recent interest in learning with Bayesian networks. In Sections 8 through 10, we consider different definitions of what is means for a model to be "good", and discuss the computations entailed by some of these definitions. In Section 11, we discuss model search.

We note that model averaging and model selection lead to models that generalize well to *new* data. That is, these techniques help us to avoid the overfitting of data. As is suggested by Equation 33, Bayesian methods for model averaging and model selection are efficient in the sense that all cases in

D can be used to both smooth and train the model. As we shall see in the following two sections, this advantage holds true for the Bayesian approach in general.

8 Criteria for Model Selection

Most of the literature on learning with Bayesian networks is concerned with model selection. In these approaches, some *criterion* is used to measure the degree to which a network structure (equivalence class) fits the prior knowledge and data. A search algorithm is then used to find an equivalence class that receives a high score by this criterion. Selective model averaging is more complex, because it is often advantageous to identify network structures that are significantly different. In many cases, a single criterion is unlikely to identify such complementary network structures. In this section, we discuss criteria for the simpler problem of model selection. For a discussion of selective model averaging, see Madigan and Raftery (1994).

8.1 Relative Posterior Probability

A criterion that is often used for model selection is the log of the relative posterior probability $\log p(D, S^h) = \log p(S^h) + \log p(D|S^h)$.[11] The logarithm is used for numerical convenience. This criterion has two components: the log prior and the log marginal likelihood. In Section 9, we examine the computation of the log marginal likelihood. In Section 10.2, we discuss the assessment of network-structure priors. Note that our comments about these terms are also relevant to the full Bayesian approach.

The log marginal likelihood has the following interesting interpretation described by Dawid (1984). From the chain rule of probability, we have

$$\log p(D|S^h) = \sum_{l=1}^{N} \log p(\mathbf{x}_l|\mathbf{x}_1, \ldots, \mathbf{x}_{l-1}, S^h) \tag{36}$$

The term $p(\mathbf{x}_l|\mathbf{x}_1, \ldots, \mathbf{x}_{l-1}, S^h)$ is the prediction for $\mathbf{x}_l$ made by model S^h after averaging over its parameters. The log of this term can be thought of as the utility or reward for this prediction under the utility function $\log p(\mathbf{x})$.[12] Thus, a model with the highest log marginal likelihood (or the highest posterior probability, assuming equal priors on structure) is also a model that is the best sequential predictor of the data D under the log utility function.

Dawid (1984) also notes the relationship between this criterion and cross validation. When using one form of cross validation, known as *leave-one-out* cross validation, we first train a model on all but one of the cases in the random sample—say, $V_l = \{\mathbf{x}_1, \ldots, \mathbf{x}_{l-1}, \mathbf{x}_{l+1}, \ldots, \mathbf{x}_N\}$. Then, we predict the omitted case, and reward this prediction under some utility function. Finally, we repeat this procedure for every case in the random sample, and sum the rewards for each prediction. If the

[11] An equivalent criterion that is often used is $\log(p(S^h|D)/p(S^h_0|D)) = \log(p(S^h)/p(S^h_0)) + \log(p(D|S^h)/p(D|S^h_0))$. The ratio $p(D|S^h)/p(D|S^h_0)$ is known as a *Bayes' factor*.

[12] This utility function is known as a *proper scoring rule*, because its use encourages people to assess their true probabilities. For a characterization of proper scoring rules and this rule in particular, see Bernardo (1979).

prediction is probabilistic and the utility function is $\log p(\mathbf{x})$, we obtain the cross-validation criterion

$$\mathrm{CV}(S^h, D) = \sum_{l=1}^{N} \log p(\mathbf{x}_l | V_l, S^h) \tag{37}$$

which is similar to Equation 36. One problem with this criterion is that training and test cases are interchanged. For example, when we compute $p(\mathbf{x}_1 | V_1, S^h)$ in Equation 37, we use $\mathbf{x}_2$ for training and $\mathbf{x}_1$ for testing. Whereas, when we compute $p(\mathbf{x}_2 | V_2, S^h)$, we use $\mathbf{x}_1$ for training and $\mathbf{x}_2$ for testing. Such interchanges can lead to the selection of a model that over fits the data (Dawid, 1984). Various approaches for attenuating this problem have been described, but we see from Equation 36 that the log-marginal-likelihood criterion avoids the problem altogether. Namely, when using this criterion, we never interchange training and test cases.

8.2 Local Criteria

Consider the problem of diagnosing an ailment given the observation of a set of findings. Suppose that the set of ailments under consideration are mutually exclusive and collectively exhaustive, so that we may represent these ailments using a single variable A. A possible Bayesian network for this classification problem is shown in Figure 5.

The posterior-probability criterion is *global* in the sense that it is equally sensitive to all possible dependencies. In the diagnosis problem, the posterior-probability criterion is just as sensitive to dependencies among the finding variables as it is to dependencies between ailment and findings. Assuming that we observe all (or perhaps all but a few) of the findings in D, a more reasonable criterion would be *local* in the sense that it ignores dependencies among findings and is sensitive only to the dependencies among the ailment and findings. This observation applies to all classification and regression problems with complete data.

One such local criterion, suggested by Spiegelhalter et al. (1993), is a variation on the sequential log-marginal-likelihood criterion:

$$\mathrm{LC}(S^h, D) = \sum_{l=1}^{N} \log p(a_l | \mathbf{F}_l, D_l, S^h) \tag{38}$$

where a_l and $\mathbf{F}_l$ denote the observation of the ailment A and findings $\mathbf{F}$ in the lth case, respectively. In other words, to compute the lth term in the product, we train our model S with the first $l-1$ cases, and then determine how well it predicts the ailment given the findings in the lth case. We can view this criterion, like the log-marginal-likelihood, as a form of cross validation where training and test cases are never interchanged.

The log utility function has interesting theoretical properties, but it is sometimes inaccurate for real-world problems. In general, an appropriate reward or utility function will depend on the decision-making problem or problems to which the probabilistic models are applied. Howard and Matheson (1983) have collected a series of articles describing how to construct utility models for specific decision problems. Once we construct such utility models, we can use suitably modified forms of Equation 38 for model selection.

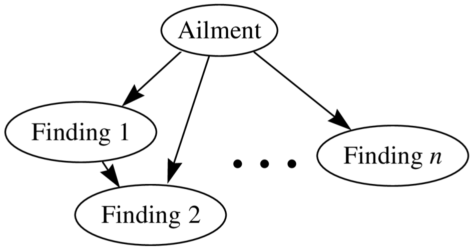


Figure 5: A Bayesian-network structure for medical diagnosis.

9 Computation of the Marginal Likelihood

As mentioned, an often-used criterion for model selection is the log relative posterior probability $\log p(D, S^h) = \log p(S^h) + \log p(D|S^h)$. In this section, we discuss the computation of the second component of this criterion: the log marginal likelihood.

Given (1) local distribution functions in the exponential family, (2) mutual independence of the parameters $\boldsymbol{\theta}_i$, (3) conjugate priors for these parameters, and (4) complete data, the log marginal likelihood can be computed efficiently and in closed form. Equation 35 is an example for unrestricted multinomial distributions. Buntine (1994) and Heckerman and Geiger (1996) discuss the computation for other local distribution functions. Here, we concentrate on approximations for incomplete data.

The Monte-Carlo and Gaussian approximations for learning about parameters that we discussed in Section 6 are also useful for computing the marginal likelihood given incomplete data. One Monte-Carlo approach, described by Chib (1995) and Raftery (1996), uses Bayes' theorem:

$$p(D|S^h) = \frac{p(\boldsymbol{\theta}_s|S^h)\ p(D|\boldsymbol{\theta}_s, S^h)}{p(\boldsymbol{\theta}_s|D, S^h)} \tag{39}$$

For any configuration of $\boldsymbol{\theta}_s$, the prior term in the numerator can be evaluated directly. In addition, the likelihood term in the numerator can be computed using Bayesian-network inference. Finally, the posterior term in the denominator can be computed using Gibbs sampling, as we described in Section 6.1. Other, more sophisticated Monte-Carlo methods are described by DiCiccio et al. (1995).

As we have discussed, Monte-Carlo methods are accurate but computationally inefficient, especially for large data sets. In contrast, methods based on the Gaussian approximation are more efficient, and can be as accurate as Monte-Carlo methods on large data sets.

Recall that, for large amounts of data, $p(D|\boldsymbol{\theta}_s, S^h) \cdot p(\boldsymbol{\theta}_s|S^h)$ can often be approximated as a multivariate-Gaussian distribution. Consequently,

$$p(D|S^h) = \int p(D|\boldsymbol{\theta}_s, S^h)\ p(\boldsymbol{\theta}_s|S^h)\ d\boldsymbol{\theta}_s \tag{40}$$

can be evaluated in closed form. In particular, substituting Equation 31 into Equation 40, integrating, and taking the logarithm of the result, we obtain the approximation:

$$\log p(D|S^h) \approx \log p(D|\tilde{\boldsymbol{\theta}}_s, S^h)\ + \log p(\tilde{\boldsymbol{\theta}}_s|S^h)\ + \frac{d}{2}\log(2\pi) - \frac{1}{2}\log|A| \tag{41}$$

where d is the dimension of $g(\boldsymbol{\theta}_s)$. For a Bayesian network with unrestricted multinomial distributions, this dimension is typically given by $\sum_{i=1}^{n} q_i(r_i - 1)$. Sometimes, when there are hidden variables, this dimension is lower. See Geiger et al. (1996) for a discussion of this point.

This approximation technique for integration is known as *Laplace's method*, and we refer to Equation 41 as the *Laplace approximation*. Kass *et al.* (1988) have shown that, under certain regularity conditions, relative errors in this approximation are $O(1/N)$, where N is the number of cases in D. Thus, the Laplace approximation can be extremely accurate. For more detailed discussions of this approximation, see—for example—Kass et al. (1988) and Kass and Raftery (1995).

Although Laplace's approximation is efficient relative to Monte-Carlo approaches, the computation of $|A|$ is nevertheless intensive for large-dimension models. One simplification is to approximate $|A|$ using only the diagonal elements of the Hessian A. Although in so doing, we incorrectly impose independencies among the parameters, researchers have shown that the approximation can be accurate in some circumstances (see, e.g., Becker and Le Cun, 1989, and Chickering and Heckerman, 1996). Another efficient variant of Laplace's approximation is described by Cheeseman and Stutz (1995), who use the approximation in the AutoClass program for data clustering (see also Chickering and Heckerman, 1996.)

We obtain a very efficient (but less accurate) approximation by retaining only those terms in Equation 41 that increase with N: $\log p(D|\tilde{\boldsymbol{\theta}}_s, S^h)$, which increases linearly with N, and $\log |A|$, which increases as $d \log N$. Also, for large N, $\tilde{\boldsymbol{\theta}}_s$ can be approximated by the ML configuration of $\boldsymbol{\theta}_s$. Thus, we obtain

$$\log p(D|S^h) \approx \log p(D|\hat{\boldsymbol{\theta}}_s, S^h) \ - \frac{d}{2} \log N \tag{42}$$

This approximation is called the *Bayesian information criterion* (BIC), and was first derived by Schwarz (1978).

The BIC approximation is interesting in several respects. First, it does not depend on the prior. Consequently, we can use the approximation without assessing a prior.[13] Second, the approximation is quite intuitive. Namely, it contains a term measuring how well the parameterized model predicts the data ($\log p(D|\hat{\boldsymbol{\theta}}_s, S^h)$) and a term that punishes the complexity of the model ($d/2 \ logN$). Third, the BIC approximation is exactly minus the Minimum Description Length (MDL) criterion described by Rissanen (1987). Thus, recalling the discussion in Section 9, we see that the marginal likelihood provides a connection between cross validation and MDL.

10 Priors

To compute the relative posterior probability of a network structure, we must assess the structure prior $p(S^h)$ and the parameter priors $p(\boldsymbol{\theta}_s|S^h)$ (unless we are using large-sample approximations such as BIC/MDL). The parameter priors $p(\boldsymbol{\theta}_s|S^h)$ are also required for the alternative scoring functions discussed in Section 8. Unfortunately, when many network structures are possible, these assessments will be intractable. Nonetheless, under certain assumptions, we can derive the structure

[13] One of the technical assumptions used to derive this approximation is that the prior is non-zero around $\hat{\boldsymbol{\theta}}_s$.

and parameter priors for many network structures from a manageable number of direct assessments. Several authors have discussed such assumptions and corresponding methods for deriving priors (Cooper and Herskovits, 1991, 1992; Buntine, 1991; Spiegelhalter *et al.*, 1993; Heckerman *et al.*, 1995b; Heckerman and Geiger, 1996). In this section, we examine some of these approaches.

10.1 Priors on Network Parameters

First, let us consider the assessment of priors for the model parameters. We consider the approach of Heckerman *et al.* (1995b) who address the case where the local distribution functions are unrestricted multinomial distributions and the assumption of parameter independence holds.

Their approach is based on two key concepts: independence equivalence and distribution equivalence. We say that two Bayesian-network structures for $\mathbf{X}$ are *independence equivalent* if they represent the same set of conditional-independence assertions for $\mathbf{X}$ (Verma and Pearl, 1990). For example, given $\mathbf{X} = \{X, Y, Z\}$, the network structures $X \to Y \to Z$, $X \leftarrow Y \to Z$, and $X \leftarrow Y \leftarrow Z$ represent only the independence assertion that X and Z are conditionally independent given Y. Consequently, these network structures are equivalent. As another example, a *complete network structure* is one that has no missing edge—that is, it encodes no assertion of conditional independence. When $\mathbf{X}$ contains n variables, there are $n!$ possible complete network structures: one network structure for each possible ordering of the variables. All complete network structures for $p(\mathbf{x})$ are independence equivalent. In general, two network structures are independence equivalent if and only if they have the same structure ignoring arc directions and the same v-structures (Verma and Pearl, 1990). 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 independence equivalence. Suppose that all Bayesian networks for $\mathbf{X}$ under consideration have local distribution functions in the family $\mathcal{F}$. This is not a restriction, per se, because $\mathcal{F}$ can be a large family. We say that two Bayesian-network structures S_1 and S_2 for $\mathbf{X}$ are *distribution equivalent with respect to (wrt)* $\mathcal{F}$ if they represent the same joint probability distributions for $\mathbf{X}$—that is, if, for every $\boldsymbol{\theta}_{s1}$, there exists a $\boldsymbol{\theta}_{s2}$ such that $p(\mathbf{x}|\boldsymbol{\theta}_{s1}, S_1^h) = p(\mathbf{x}|\boldsymbol{\theta}_{s2}, S_2^h)$, and vice versa.

Distribution equivalence wrt some $\mathcal{F}$ implies independence equivalence, but the converse does not hold. For example, when $\mathcal{F}$ is the family of generalized linear-regression models, the complete network structures for $n \geq 3$ variables do not represent the same sets of distributions. Nonetheless, there are families $\mathcal{F}$—for example, unrestricted multinomial distributions and linear-regression models with Gaussian noise—where independence equivalence implies distribution equivalence wrt $\mathcal{F}$ (Heckerman and Geiger, 1996).

The notion of distribution equivalence is important, because if two network structures S_1 and S_2 are distribution equivalent wrt to a given $\mathcal{F}$, then the hypotheses associated with these two structures are identical—that is, $S_1^h = S_2^h$. Thus, for example, if S_1 and S_2 are distribution equivalent, then their probabilities must be equal in any state of information. Heckerman *et al.* (1995b) call this property *hypothesis equivalence.*

In light of this property, we should associate each hypothesis with an equivalence class of structures rather than a single network structure, and our methods for learning network structure should actually be interpreted as methods for learning equivalence classes of network structures (although, for the sake of brevity, we often blur this distinction). Thus, for example, the sum over network-structure hypotheses in Equation 33 should be replaced with a sum over equivalence-class hypotheses. An efficient algorithm for identifying the equivalence class of a given network structure can be found in Chickering (1995).

We note that hypothesis equivalence holds provided we interpret Bayesian-network structure simply as a representation of conditional independence. Nonetheless, stronger definitions of Bayesian networks exist where arcs have a causal interpretation (see Section 15). Heckerman *et al.* (1995b) and Heckerman (1995) argue that, although it is unreasonable to assume hypothesis equivalence when working with causal Bayesian networks, it is often reasonable to adopt a weaker assumption of *likelihood equivalence,* which says that the observational data can not help to discriminate two indepence equivalent network structures.

Now let us return to the main issue of this section: the derivation of priors from a manageable number of assessments. Geiger and Heckerman (1995) show that the assumptions of parameter independence and likelihood equivalence imply that the parameters for any *complete* network structure S_c must have a Dirichlet distribution with constraints on the hyperparameters given by

$$\alpha_{ijk} = \alpha \; p(x_i^k, \mathbf{pa}_i^j | S_c^h) \tag{43}$$

where α is the user's equivalent sample size,[14], and $p(x_i^k, \mathbf{pa}_i^j | S_c^h)$ is computed from the user's joint probability distribution $p(\mathbf{x}|S_c^h)$. This result is rather remarkable, as the two assumptions leading to the constrained Dirichlet solution are qualitative.

To determine the priors for parameters of *incomplete* network structures, Heckerman *et al.* (1995b) use the assumption of *parameter modularity,* which says that if X_i has the same parents in network structures S_1 and S_2, then

$$p(\boldsymbol{\theta}_{ij}|S_1^h) = p(\boldsymbol{\theta}_{ij}|S_2^h)$$

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 the structure of the network that is local to variable X_i—namely, X_i and its parents.

Given the assumptions of parameter modularity and parameter independence,[15] it is a simple matter to construct priors for the parameters of an arbitrary network structure given the priors on complete network structures. 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 network structure, we identify a complete network structure where X_i has these parents, and use Equation 43 and parameter modularity to determine the priors for this node. The result is that all terms α_{ijk} for all network structures are determined by Equation 43. Thus, from the assessments α and $p(\mathbf{x}|S_c^h)$, we can derive the parameter priors for all possible network structures. Combining

[14] Recall the method of equivalent samples for assessing beta and Dirichlet distributions discussed in Section 2.

[15] This construction procedure also assumes that every structure has a non-zero prior probability.

Equation 43 with Equation 35, we obtain a model-selection criterion that assigns equal marginal likelihoods to independence equivalent network structures.

We can assess $p(\mathbf{x}|S_c^h)$ by constructing a Bayesian network, called a *prior network*, that encodes this joint distribution. Heckerman *et al.* (1995b) discuss the construction of this network.

10.2 Priors on Structures

Now, let us consider the assessment of priors on network-structure hypotheses. Note that the alternative criteria described in Section 8 can incorporate prior biases on network-structure hypotheses. Methods similar to those discussed in this section can be used to assess such biases.

The simplest approach for assigning priors to network-structure hypotheses is to assume that every hypothesis 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 hypotheses (perhaps based on judgments of of cause and effect), and then impose a uniform prior on the remaining hypotheses. We illustrate this approach in Section 12.

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 network-structure hypothesis. 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.* (1995b) uses a prior network. 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.* (1995b) suggest one reasonable measure of deviation.

Madigan *et al.* (1995) give yet another approach that makes use of imaginary data from a domain expert. In their approach, a computer program helps the user create a hypothetical set of complete data. Then, using techniques such as those in Section 7, they compute the posterior probabilities of network-structure hypotheses given this data, assuming the prior probabilities of hypotheses are uniform. Finally, they use these posterior probabilities as priors for the analysis of the real data.

11 Search Methods

In this section, we examine search methods for identifying network structures with high scores by some criterion. Consider the problem of finding the best network from the set of all networks 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 43 (Chickering *et al.* 1995). 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 efficient when the model-selection criterion is separable. Given a network structure for domain $\mathbf{X}$, we say that a criterion for that structure is *separable* if it can be written as a product of variable-specific criteria:

$$\mathrm{C}(S^h, D) = \prod_{i=1}^{n} c(X_i, \mathbf{Pa}_i, D_i) \tag{44}$$

where D_i is the data restricted to the variables X_i and $\mathbf{Pa}_i$. An example of a separable criterion is the BD criterion (Equations 34 and 35) used in conjunction with any of the methods for assessing structure priors described in Section 10.

Most of the commonly used search methods for Bayesian networks make successive arc changes to the network, and employ the property of separability to evaluate the merit of each change. The possible changes that can be made are easy to identify. 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 network 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 score of the network resulting from the modification $e \in E$. Given a separable criterion, if an arc to X_i is added or deleted, only $c(X_i, \mathbf{Pa}_i, D_i)$ need be evaluated to determine $\Delta(e)$. If an arc between X_i and X_j is reversed, then only $c(X_i, \mathbf{Pa}_i, D_i)$ and $c(X_j, \Pi_j, D_j)$ need be evaluated.

One simple heuristic search algorithm is greedy search. First, we choose a network structure. 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 $\Delta(e)$. When the criterion is separable, we can avoid recomputing all terms $\Delta(e)$ after every change. In particular, if neither X_i, X_j, nor their parents are changed, then $\Delta(e)$ remains unchanged for all changes e involving these nodes as long as the resulting network is acyclic. Candidates for the initial graph include the empty graph, a random graph, and a prior network.

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 network 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 (e.g., Korf, 1993). In this approach, the space of all network structures is searched systematically using a heuristic measure that determines the next best structure to examine. Chickering (1996b) has 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 Bayesian-network structures. Nonetheless, when the assumption of hypothesis equivalence holds, one can search through the space of network-structure 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. Work by Spirtes and Meek (1995) and Chickering (1996) confirm these observations experimentally. Unfortunately, no comparisons are yet available that determine whether the benefits of equivalence-class search outweigh the costs.

12 A Simple Example

Before we move on to other issues, let us step back and look at our overall approach. In a nutshell, we can construct both structure and parameter priors by constructing a Bayesian network (the prior network) along with additional assessments such as an equivalent sample size and causal constraints. We then use either Bayesian model selection, selective model averaging, or full model averaging to obtain one or more networks for prediction and/or explanation. In effect, we have a procedure for using data to improve the structure and probabilities of an initial Bayesian network.

Here, we present two artificial examples to illustrate this process. Consider again the problem of fraud detection from Section 3. Suppose we are given the data set D in Table 12, and we want to predict the next case—that is, compute $p(\mathbf{x}_{N+1}|D)$. Let us assert that only two network-structure hypotheses have appreciable probability: the hypothesis corresponding to the network structure in Figure 3 (S_1), and the hypothesis corresponding to the same structure with an arc added from *Age* to *Gas* (S_2). Furthermore, let us assert that these two hypotheses are equally likely—that is, $p(S_1^h) = p(S_2^h) = 0.5$. In addition, let us use the parameter priors given by Equation 43, where $\alpha = 10$ and $p(\mathbf{x}|S_c^h)$ is given by the prior network in Figure 3. Using Equations 34 and 35, we obtain $p(S_1^h|D) = 0.26$ and $p(S_2^h|D) = 0.74$. Because we have only two models to consider, we can model average according to Equation 33:

$$p(\mathbf{x}_{N+1}|D) = 0.26\ p(\mathbf{x}_{N+1}|D, S_1^h) + 0.74\ p(\mathbf{x}_{N+1}|D, S_2^h)$$

where $p(\mathbf{x}_{N+1}|D, S^h)$ is given by Equation 27. (We don't display these probability distributions.) If we had to choose one model, we would choose S_2, assuming the posterior-probability criterion is appropriate. Note that the data favors the presence of the arc from *Age* to *Gas* by a factor of three. This is not surprising, because in the two cases in the data set where fraud is absent and gas was purchased recently, the card holder was less than 30 years old.

An application of model selection, described by Spirtes and Meek (1995), is illustrated in Figure 6. Figure 6a is a hand-constructed Bayesian network for the domain of ICU ventilator management,

Table 1: An imagined data set for the fraud problem.

Case	Fraud	Gas	Jewelry	Age	Sex
1	no	no	no	30-50	female
2	no	no	no	30-50	male
3	yes	yes	yes	>50	male
4	no	no	no	30-50	male
5	no	yes	no	<30	female
6	no	no	no	<30	female
7	no	no	no	>50	male
8	no	no	yes	30-50	female
9	no	yes	no	<30	male
10	no	no	no	<30	female

called the Alarm network (Beinlich et al., 1989). Figure 6c is a random sample from the Alarm network of size 10,000. Figure 6b is a simple prior network for the domain. This network encodes mutual independence among the variables, and (not shown) uniform probability distributions for each variable.

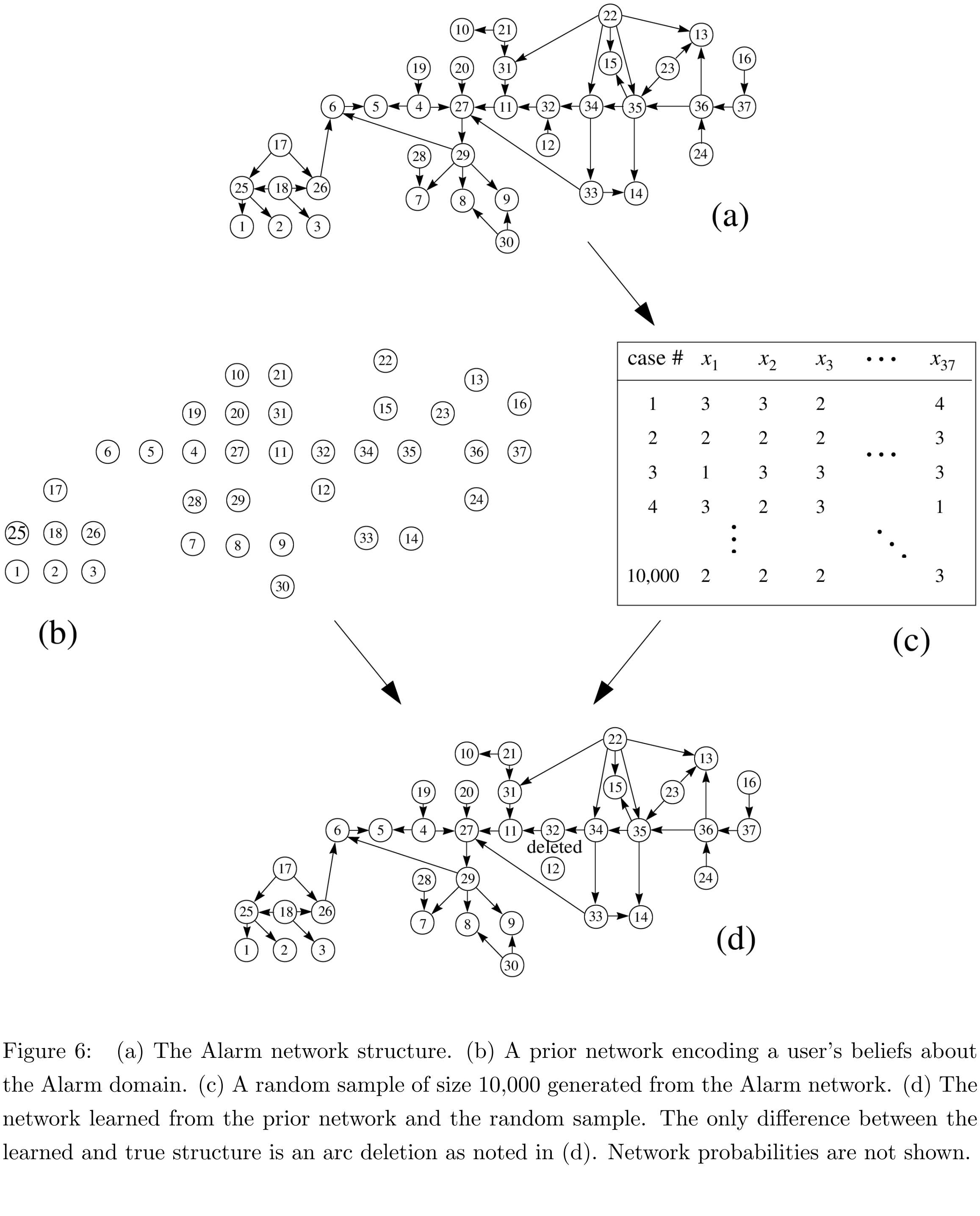
Figure 6d shows the most likely network structure found by a two-pass greedy search in equivalence-class space. In the first pass, arcs were added until the model score did not improve. In the second pass, arcs were deleted until the model score did not improve. Structure priors were uniform; and parameter priors were computed from the prior network using Equation 43 with $\alpha = 10$.

The network structure learned from this procedure differs from the true network structure only by a single arc deletion. In effect, we have used the data to improve dramatically the original model of the user.

13 Bayesian Networks for Supervised Learning

As we discussed in Section 5, the local distribution functions $p(x_i|\mathbf{pa}_i, \boldsymbol{\theta}_i, S^h)$ are essentially classification/regression models. Therefore, if we are doing supervised learning where the explanatory (input) variables cause the outcome (target) variable and data is complete, then the Bayesian-network and classification/regression approaches are identical.

When data is complete but input/target variables do not have a simple cause/effect relationship, tradeoffs emerge between the Bayesian-network approach and other methods. For example, consider the classification problem in Figure 5. Here, the Bayesian network encodes dependencies between findings and ailments as well as among the findings, whereas another classification model such as a decision tree encodes only the relationships between findings and ailment. Thus, the decision tree may produce more accurate classifications, because it can encode the necessary relationships with fewer parameters. Nonetheless, the use of local criteria for Bayesian-network model selection mitigates this advantage. Furthermore, the Bayesian network provides a more natural representation



case #	x_1	x_2	x_3	• • •	x_{37}
1	3	3	2		4
2	2	2	2	• • •	3
3	1	3	3		3
4	3	2	3		1
	⋮			⋱	
10,000	2	2	2		3

Figure 6: (a) The Alarm network structure. (b) A prior network encoding a user's beliefs about the Alarm domain. (c) A random sample of size 10,000 generated from the Alarm network. (d) The network learned from the prior network and the random sample. The only difference between the learned and true structure is an arc deletion as noted in (d). Network probabilities are not shown.

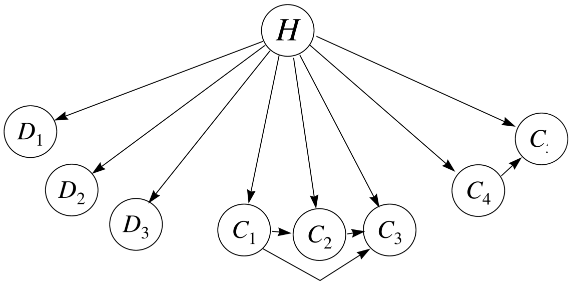


Figure 7: A Bayesian-network structure for AutoClass. The variable H is hidden. Its possible states correspond to the underlying classes in the data.

in which to encode prior knowledge, thus giving this model a possible advantage for sufficiently small sample sizes. Another argument, based on bias–variance analysis, suggests that neither approach will dramatically outperform the other (Friedman, 1996).

Singh and Provan (1995) compare the classification accuracy of Bayesian networks and decision trees using complete data sets from the University of California, Irvine Repository of Machine Learning data sets. Specifically, they compare C4.5 with an algorithm that learns the structure and probabilities of a Bayesian network using a variation of the Bayesian methods we have described. The latter algorithm includes a model-selection phase that discards some input variables. They show that, overall, Bayesian networks and decisions trees have about the same classification error. These results support the argument of Friedman (1996).

When the input variables cause the target variable and data is incomplete, the dependencies between input variables becomes important, as we discussed in the introduction. Bayesian networks provide a natural framework for learning about and encoding these dependencies. Unfortunately, no studies have been done comparing these approaches with other methods for handling missing data.

14 Bayesian Networks for Unsupervised Learning

The techniques described in this paper can be used for unsupervised learning. A simple example is the AutoClass program of Cheeseman and Stutz (1995), which performs data clustering. The idea behind AutoClass is that there is a single hidden (i.e., never observed) variable that causes the observations. This hidden variable is discrete, and its possible states correspond to the underlying classes in the data. Thus, AutoClass can be described by a Bayesian network such as the one in Figure 7. For reasons of computational efficiency, Cheeseman and Stutz (1995) assume that the discrete variables (e.g., D_1, D_2, D_3 in the figure) and user-defined sets of continuous variables (e.g., $\{C_1, C_2, C_3\}$ and $\{C_4, C_5\}$) are mutually independent given H. Given a data set D, AutoClass searches over variants of this model (including the number of states of the hidden variable) and selects a variant whose (approximate) posterior probability is a local maximum.

AutoClass is an example where the user presupposes the existence of a hidden variable. In other situations, we may be unsure about the presence of a hidden variable. In such cases, we can score

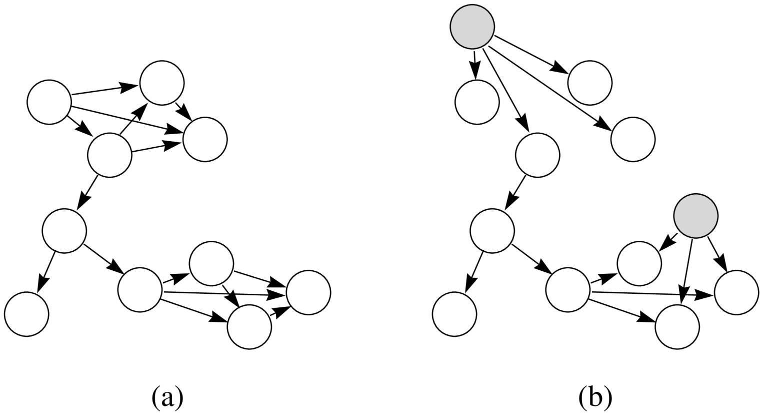


Figure 8: (a) A Bayesian-network structure for observed variables. (b) A Bayesian-network structure with hidden variables (shaded) suggested by the network structure in (a).

models with and without hidden variables to reduce our uncertainty. We illustrate this approach on a real-world case study in Section 16. Alternatively, we may have little idea about what hidden variables to model. The search algorithms of Spirtes *et al.* (1993) provide one method for identifying possible hidden variables in such situations. Martin and VanLehn (1995) suggest another method.

Their approach is based on the observation that if a set of variables are mutually dependent, then a simple explanation is that these variables have a single hidden common cause rendering them mutually independent. Thus, to identify possible hidden variables, we first apply some learning technique to select a model containing no hidden variables. Then, we look for sets of mutually dependent variables in this learned model. For each such set of variables (and combinations thereof), we create a new model containing a hidden variable that renders that set of variables conditionally independent. We then score the new models, possibly finding one better than the original. For example, the model in Figure 8a has two sets of mutually dependent variables. Figure 8b shows another model containing hidden variables suggested by this model.

15 Learning Causal Relationships

As we have mentioned, the causal semantics of a Bayesian network provide a means by which we can learn causal relationships. In this section, we examine these semantics, and provide a basic discussion on how causal relationships can be learned. We note that these methods are new and controversial.[16] For critical discussions on both sides of the issue, see Spirtes *et al.* (1993), Pearl (1995), and Humphreys and Freedman (1995).

For purposes of illustration, suppose we are marketing analysts who want to know whether we should increase the placement of a magazine ad to increase sales of a product. If seeing the ad is a

[16] There was certainly controversy in 1996 when I wrote this tutorial. Now in 2020, these ideas are much more accepted. I leave this comment here as the only purpose of this revision is to correct errors in the original.

cause of busying the product, then we want to show more of it. More generally, causal knowledge is at the heart of understanding what will happen when we intervene. For detailed discussions of the meaning of cause and the close connections between causal knowledge and the consequences of intervention, see Pearl (1995) and Heckerman and Shachter (1995).

How do we learn causal relationships from data? One approach is to actually intervene, and note the consequences. In the classic embodiment of this approach, we perform a randomized experiment. In the ad example, we would (1) select a set of individuals at random, (2) for each individual, show them the ad if and only if a coin flip comes up heads, and (3) note any difference in buying behavior between the two groups. Unfortunately, randomized experiments can be expensive (as in this example), unethical, or difficult to obtain compliance.

Over the last century, many researchers have developed methods for inferring cause and effect. Here, we consider a new approach that makes use of a Bayesian network with causal semantics to infer causal knowledge from observational data alone. The basic idea is to *assume* a connection between causal and probabilistic relationships in a directed network. More precisely, we say that a directed acyclic graph $\mathcal{C}$ is a *causal graph for variables* $\mathbf{X}$ if the nodes in $\mathcal{C}$ are in a one-to-one correspondence with $\mathbf{X}$, and there is an arc from node X to node Y in $\mathcal{C}$ if and only if X is a direct cause of Y. We now assume that if $\mathcal{C}$ is a causal graph for $\mathbf{X}$, then $\mathcal{C}$ is also a Bayesian-network structure for the joint physical probability distribution of $\mathbf{X}$. This assumption, known as the *causal Markov condition*, is considered to be quite weak (*i.e.,* quite reasonable) by many philosophers (Spirtes *et al.*, 1993). We can infer causal relationships from observational data by first inferring probabilistic independence relationships as we have described earlier in this tutorial, and then inferring causal relationships using this assumption.[17] As we shall see, we can not always learn causal relationships with this approach. Whether we can, will depend on the probabilistic relationships we find.

To illustrate this approach, let us return to the ad example. Let variables *Ad* (A) and *Buy* (B) represent whether or not an individual has seen the advertisement and has purchased the product, respectively. Assuming these variables are dependent, there are four simple causal explanations: (1) A is a cause of B, (2) B is a cause of A, (3) there are one or more hidden common causes of A and B (e.g., the person's gender), and (4) A and B are causes for data selection. This last explanation is an example of *selection bias*. Condition 4 would occur, for example, if our data set failed to include instances where A and B are false. These four causal explanations for the presence of the arcs are illustrated in Figure 9a. Of course, more complicated explanations—such as the presence of a hidden common cause and selection bias—are possible.

When we have observations only for A and B, it is not possible to distinguish among these various causal hypotheses. Suppose, however, that we observe two additional variables: *Income* (I) and *Location* (L), which represent the income and geographic location of the possible purchaser, respectively. Furthermore, suppose we learn (with high probability) the Bayesian network shown in Figure 9b. Given the causal Markov condition, the *only* causal explanation for the conditional-independence and conditional-dependence relationships encoded in this Bayesian network is that

[17] This approach also requires the assumption of *faithfulness*, which says that causal relationships do not accidentally produce probabilistic independence. In our Bayesian approach to learning networks, however, faithfulness follows from our assumption that $p(\boldsymbol{\theta}_s|S^h)$ is a probability density function.

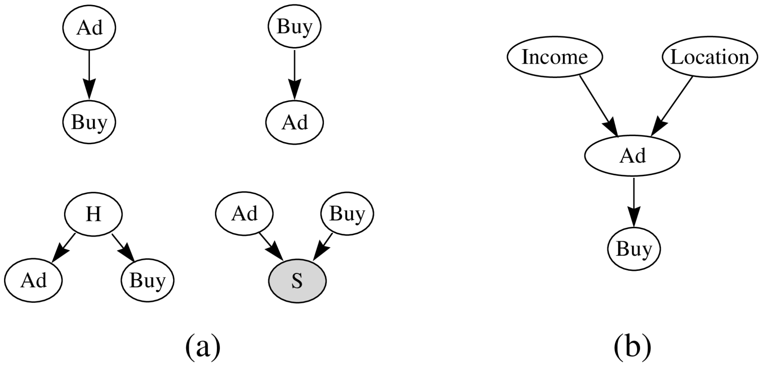


Figure 9: (a) Causal graphs showing four explanations for an observed dependence between *Ad* and *Buy*. The node H corresponds to a hidden common cause of *Ad* and *Buy*. The shaded node S indicates that the case has been included in the data set. (b) A Bayesian network for which *Ad* causes *Buy* is the only causal explanation, given the causal Markov condition.

Ad is a cause for *Buy*. More precisely, none of the other explanations described in the previous paragraph, or combinations thereof, produce the probabilistic relationships encoded in Figure 9b.

16 A Case Study: College Plans

Real-world applications of techniques that we have discussed can be found in Madigan and Raftery (1994), Lauritzen et al. (1994), Singh and Provan (1995), and Friedman and Goldszmidt (1996). Here, we consider an application that comes from a study by Sewell and Shah (1968), who investigated factors that influence the intention of high school students to attend college. The data have been analyzed by several groups of statisticians, including Whittaker (1990) and Spirtes *et al.* (1993), all of whom have used non-Bayesian techniques.

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 here is to understand the (possibly causal) relationships among these variables.

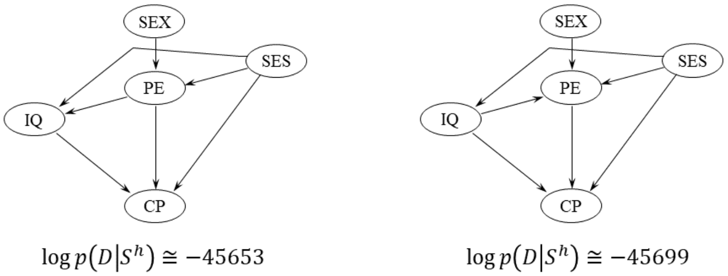
The data are described by the sufficient statistics in Table 16. Each entry denotes the number of cases in which the five variables take on some particular configuration. The first entry corresponds to the configuration SEX=male, SES=low, IQ=low, PE=low, and CP=yes. The remaining entries correspond to 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.

As a first pass, we analyzed the data assuming there are no hidden variables. To generate priors for network parameters, we used the method described in Section 10.1 with an equivalent sample size of 5 and a prior network where $p(\mathbf{x}|S_c^h)$ is uniform. (The results were not sensitive to the choice

Table 2: 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

Reproduced by permission from the University of Chicago Press. ©1968 by The University of Chicago. All rights reserved.



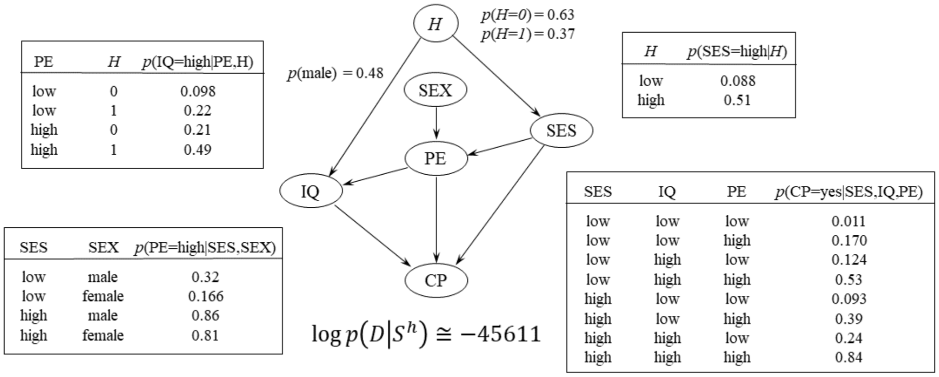
$\log p(D|S^h) \cong -45653$ $\log p(D|S^h) \cong -45699$

Figure 10: The two network structures without hidden variables with the highest marginal likelihoods.

of parameter priors. For example, none of the results reported in this section changed qualitatively for equivalent sample sizes ranging from 3 to 40.) We considered all possible structures except those where SEX and/or SES had parents, and/or CP had children. Because the data set was complete, we used Equations 34 and 35 to compute the marginal likelihoods. The two network structures with the highest marginal likelihoods are shown in Figure 10.

If we adopt the causal Markov assumption and also assume that there are no hidden variables, then the arcs in both graphs can be interpreted causally. Some results are not surprising—for example the causal influence of socioeconomic status and IQ on college plans. Other results are more interesting. For example, from either graph we conclude that sex influences college plans only indirectly through parental influence. Also, the two graphs differ only by the orientation of the arc between PE and IQ. Either causal relationship is plausible. We note that the second most likely graph was selected by Spirtes *et al.* (1993), who used a non-Bayesian approach to infer the network.

The most suspicious result is the suggestion that socioeconomic status has a direct influence on IQ. To question this result, we considered new network structures obtained from those in Figure 10



PE	H	p(IQ=high|PE,H)
low	0	0.098
low	1	0.22
high	0	0.21
high	1	0.49

H	p(SES=high|H)
low	0.088
high	0.51

SES	SEX	p(PE=high|SES,SEX)
low	male	0.32
low	female	0.166
high	male	0.86
high	female	0.81

SES	IQ	PE	p(CP=yes|SES,IQ,PE)
low	low	low	0.011
low	low	high	0.170
low	high	low	0.124
low	high	high	0.53
high	low	low	0.093
high	low	high	0.39
high	high	low	0.24
high	high	high	0.84

$$\log p(D|S^h) \cong -45611$$

Figure 11: One of the two hidden-variable network structures with the highest marginal likelihood. The other model, which has the same marginal likelihood, has the arc from H to SES reversed. Probabilities shown are MAP values. Some probability distributions are omitted.

by replacing this direct influence with a hidden variable pointing to both SES and IQ, and by inserting a hidden variable between SES and IQ. We also considered structures where the hidden variable pointed to SES, IQ, and PE, and none, one, or both of the connections SES—PE and PE—IQ were removed. For each structure, we varied the number of states of the hidden variable from two to six.

We computed the marginal likelihoods of these structures using the Cheeseman-Stutz (1995) variant of the Laplace approximation. To find the MAP $\tilde{\boldsymbol{\theta}}_s$, we used the EM algorithm, taking the largest local maximum from among 100 runs with different random initializations of $\boldsymbol{\theta}_s$. We then computed more accurate marginal likelihoods for the best networks using annealed importance sampling, a Monte-Carlo technique [**?**]. Among the structures considered, there were two with equal marginal likelihoods, much higher than the marginal likelihoods of the best structures without hidden variables. One is shown in Figure 11. The other has the arc from H to SES reversed. In the model not shown in the figure, SES is a cause of IQ as in the best model with no hidden variables, and H helps to capture the orderings of the states in SES and IQ that are ignored in the multinomial model. In the model shown in the figure, H again helps to capture the ordering of states, but is also a hidden common cause of IQ and SES (*e.g.*,, parent "quality."). Although observational data can not be used to discriminate between these two structures, the one where H is a hidden common cause of SES and IQ could be favored *a priori.*

17 Pointers to Literature and Software

Like all tutorials, this one is incomplete. For those readers interested in learning more about graphical models and methods for learning them, we offer the following additional references and pointers to software. Buntine (1996) provides another guide to the literature.

Spirtes *et al.* (1993) and Pearl (1995) use methods based on large-sample approximations to learn Bayesian networks. In addition, as we have discussed, they describe methods for learning causal relationships from observational data.

In addition to directed models, researchers have explored network structures containing undirected edges as a knowledge representation. These representations are discussed (e.g.) in Lauritzen (1982), Verma and Pearl (1990), Frydenberg (1990), Whittaker (1990), and Richardson (1997). Bayesian methods for learning such models from data are described by Dawid and Lauritzen (1993) and Buntine (1994).

Finally, several research groups have developed software systems for learning graphical models. For example, Scheines *et al.* (1994) have developed a software program called TETRAD II for learning about cause and effect. Badsberg (1992) and Højsgaard et al. (1994) have built systems that can learn with mixed graphical models using a variety of criteria for model selection. Thomas, Spiegelhalter, and Gilks (1992) have created a system called BUGS that takes a learning problem specified as a Bayesian network and compiles this problem into a Gibbs-sampler computer program.

Acknowledgments

I thank Radford Neal and Max Chickering for scoring the Sewall and Shah network structures with hidden variables, and Chris Meek for bringing this data set to my attention. I thank Max Chickering, Usama Fayyad, Eric Horvitz, Chris Meek, Koos Rommelse, and Padhraic Smyth for their comments on earlier versions of this manuscript.

Notation

$X, Y, Z, \ldots$	Variables or their corresponding nodes in a Bayesian network
$\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \ldots$	Sets of variables or corresponding sets of nodes
$X = x$	Variable X is in state x
$\mathbf{X} = \mathbf{x}$	The set of variables $\mathbf{X}$ is in configuration $\mathbf{x}$
$\mathbf{x}, \mathbf{y}, \mathbf{z}$	Typically refer to a complete case, an incomplete case, and missing data in a case, respectively
$\mathbf{X} \setminus \mathbf{Y}$	The variables in X that are not in Y
D	A data set: a set of cases
D_l	The first $l - 1$ cases in D
$p(\mathbf{x}\|\mathbf{y})$	The probability that $\mathbf{X} = \mathbf{x}$ given $\mathbf{Y} = \mathbf{y}$ (also used to describe a probability density, probability distribution, and probability density)
$\mathrm{E}_{p(\cdot)}(x)$	The expectation of x with respect to $p(\cdot)$
S	A Bayesian network structure (a directed acyclic graph)
$\mathbf{Pa}_i$	The variable or node corresponding to the parents of node X_i in a Bayesian network structure
$\mathbf{pa}_i$	A configuration of the variables $\mathbf{Pa}_i$
r_i	The number of states of discrete variable X_i
q_i	The number of configurations of $\mathbf{Pa}_i$
S_c	A complete network structure
S^h	The hypothesis corresponding to network structure S
θ_{ijk}	The multinomial parameter corresponding to the probability $p(X_i = x_i^k \| \mathbf{Pa}_i = \mathbf{pa}_i^j)$
$\boldsymbol{\theta}_{ij}$	$= (\theta_{ij2}, \ldots, \theta_{ijr_i})$
$\boldsymbol{\theta}_i$	$= (\boldsymbol{\theta}_{i1}, \ldots, \boldsymbol{\theta}_{iq_i})$
$\boldsymbol{\theta}_s$	$= (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n)$
α	An equivalent sample size
α_{ijk}	The Dirichlet hyperparameter corresponding to θ_{ijk}
α_{ij}	$= \sum_{k=1}^{r_i} \alpha_{ijk}$
N_{ijk}	The number of cases in data set D where $X_i = x_i^k$ and $\mathbf{Pa}_i = \mathbf{pa}_i^j$
N_{ij}	$= \sum_{k=1}^{r_i} N_{ijk}$