Learning Bayesian Networks: The Combination of Knowledge and Statistical Data

David Heckerman
heckerma@microsoft.com

Dan Geiger
dang@cs.technion.ac.il

David M. Chickering
dmax@cs.ucla.edu

March 1994 (Revised February 1995)

Technical Report
MSR-TR-94-09

Microsoft Research
Advanced Technology Division
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

To appear in *Machine Learning,* 1995.

Abstract

We describe a Bayesian approach for learning Bayesian networks from a combination of prior knowledge and statistical data. First and foremost, we develop a methodology for assessing informative priors needed for learning. Our approach is derived from a set of assumptions made previously as well as the assumption of *likelihood equivalence*, which says that data should not help to discriminate network structures that represent the same assertions of conditional independence. We show that likelihood equivalence when combined with previously made assumptions implies that the user's priors for network parameters can be encoded in a single Bayesian network for the next case to be seen—a *prior network*—and a single measure of confidence for that network. Second, using these priors, we show how to compute the relative posterior probabilities of network structures given data. Third, we describe search methods for identifying network structures with high posterior probabilities. We describe polynomial algorithms for finding the highest-scoring network structures in the special case where every node has at most $k = 1$ parent. For the general case ($k > 1$), which is NP-hard, we review heuristic search algorithms including local search, iterative local search, and simulated annealing. Finally, we describe a methodology for evaluating Bayesian-network learning algorithms, and apply this approach to a comparison of various approaches.

Keywords: Bayesian networks, learning, Dirichlet, likelihood equivalence, maximum branching, heuristic search

1 Introduction

A Bayesian network is an annotated directed graph that encodes probabilistic relationships among distinctions of interest in an uncertain-reasoning problem [Howard and Matheson, 1981, Pearl, 1988]. The representation formally encodes the joint probability distribution for its domain, yet includes a human-oriented qualitative structure that facilitates communication between a user and a system incorporating the probabilistic model. We discuss the representation in detail in Section 2. For over a decade, AI researchers have used Bayesian networks to encode expert knowledge. More recently, AI researchers and statisticians have begun to investigate methods for learning Bayesian networks, including Bayesian methods [Cooper and Herskovits, 1991, Buntine, 1991, Spiegelhalter et al., 1993, Dawid and Lauritzen, 1993, Heckerman et al., 1994], quasi-Bayesian methods [Lam and Bacchus, 1993, Suzuki, 1993], and nonBayesian methods [Pearl and Verma, 1991, Spirtes et al., 1993].

In this paper, we concentrate on the Bayesian approach, which takes prior knowledge and combines it with data to produce one or more Bayesian networks. Our approach is illustrated in Figure 1 for the problem of ICU ventilator management. Using our method, a user specifies his prior knowledge about the problem by constructing a Bayesian network,

called a *prior network*, and by assessing his confidence in this network. A hypothetical prior network is shown in Figure 1b (the probabilities are not shown). In addition, a database of cases is assembled as shown in Figure 1c. Each case in the database contains observations for every variable in the user's prior network. Our approach then takes these sources of information and learns one (or more) new Bayesian networks as shown in Figure 1d. To appreciate the effectiveness of the approach, note that the database was generated from the Bayesian network in Figure 1a known as the Alarm network [Beinlich et al., 1989]. Comparing the three network structures, we see that the structure of the learned network is much closer to that of the Alarm network than is the structure of the prior network. In effect, our learning algorithm has used the database to "correct" the prior knowledge of the user.

Our Bayesian approach can be understood as follows. Suppose we have a domain of discrete variables $\{x_1, \ldots, x_n\} = U$, and a database of cases $\{C_1, \ldots, C_m\} = D$. Further, suppose that we wish to determine the joint distribution $p(C|D, \xi)$—the probability distribution of a new case C, given the database and our current state of information ξ. Rather than reason about this distribution directly, we imagine that the data is a random sample from an unknown Bayesian network structure B_s with unknown parameters. Using B_s^h to denote the hypothesis that the data is generated by network structure B_s, and assuming the hypotheses corresponding to all possible network structures form a mutually exclusive and collectively exhaustive set, we have

$$p(C|D, \xi) = \sum_{\text{all } B_s^h} p(C|D, B_s^h, \xi) \cdot p(B_s^h|D, \xi)$$

In practice, it is impossible to sum over all possible network structures. Consequently, we attempt to identify a small subset H of network-structure hypotheses that account for a large fraction of the posterior probability of the hypotheses. Rewriting the previous equation, we obtain

$$p(C|D, \xi) \approx c \sum_{B_s^h \in H} p(C|D, B_s^h, \xi) \cdot p(B_s^h|D, \xi)$$

where c is the normalization constant $1/[\sum_{B_s^h \in H} p(B_s^h|D, \xi)]$. From this relation, we see that only the relative posterior probabilities of hypotheses matter. Thus, rather than compute a posterior probability, which would entail summing over all structures, we can compute a *Bayes' factor*—$p(B_s^h|D, \xi)/p(B_{s0}^h|D, \xi)$—where B_{s0} is some reference structure such as the one containing no arcs, or simply $p(D, B_s^h|\xi) = p(B_s^h|\xi)\ p(D|B_s^h, \xi)$. In the latter case, we have

$$p(C|D, \xi) \approx c' \sum_{B_s^h \in H} p(C|D, B_s^h, \xi) \cdot p(D, B_s^h|\xi) \tag{1}$$

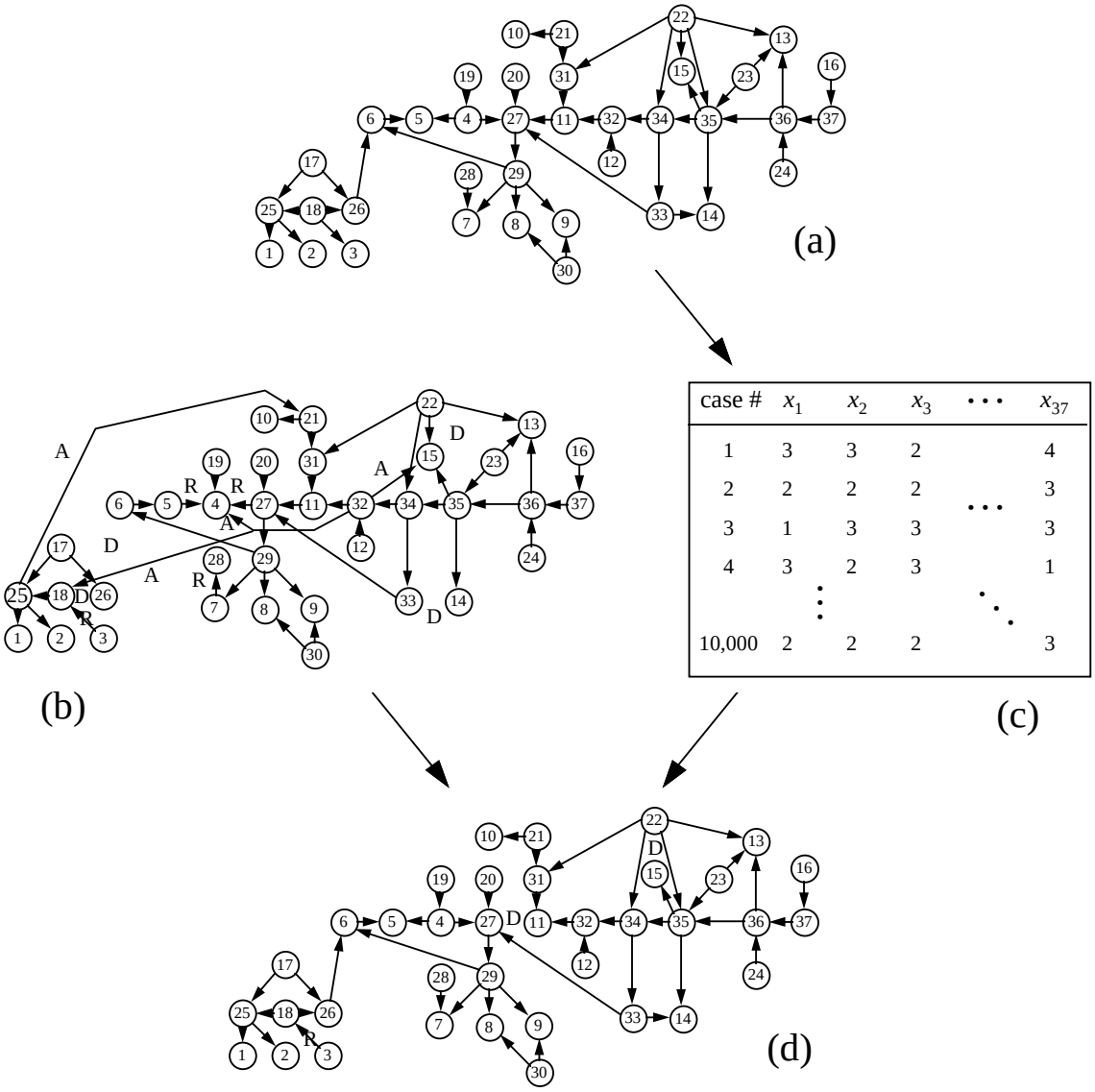

Figure 1: (a) The Alarm network structure. (b) A prior network encoding a user's beliefs about the Alarm domain. (c) A 10,000-case database generated from the Alarm network. (d) The network learned from the prior network and a 10,000 case database generated from the Alarm network. Arcs that are added, deleted, or reversed with respect to the Alarm network are indicated with A, D, and R, respectively.

where c' is another normalization constant $1/[\sum_{B_s^h \in H} p(D, B_s^h|\xi)]$.

In short, the Bayesian approach to learning Bayesian networks amounts to searching for network-structure hypotheses with high relative posterior probabilities. Many non-Bayesian approaches use the same basic approach, but optimize some other measure of how well the structure fits the data. In general, we refer to such measures as *scoring metrics*. We refer to any formula for computing the relative posterior probability of a network-structure hypothesis as a *Bayesian scoring metric*.

The Bayesian approach is not only an approximation for $p(C|D,\xi)$ but a method for learning network structure. When $|H| = 1$, we learn a single network structure: the MAP (*maximum a posteriori*) structure of U. When $|H| > 1$, we learn a collection of network structures. As we discuss in Section 4, learning network structure is useful, because we can sometimes use structure to infer causal relationships in a domain, and consequently predict the effects of interventions.

One of the most challenging tasks in designing a Bayesian learning procedure is identifying classes of easy-to-assess informative priors for computing the terms on the right-hand-side of Equation 1. In the first part of the paper (Sections 3 through 6), we explicate a set of assumptions for discrete networks—networks containing only discrete variables—that leads to such a class of informative priors. Our assumptions are based on those made by Cooper and Herskovits (1991, 1992)—herein referred to as CH—Spiegelhalter et al. (1993) and Dawid and Lauritzen (1993)—herein referred to as SDLC—and Buntine (1991). These researchers assumed *parameter independence*, which says that the parameters associated with each node in a Bayesian network are independent, *parameter modularity*, which says that if a node has the same parents in two distinct networks, then the probability density functions of the parameters associated with this node are identical in both networks, and the *Dirichlet assumption*, which says that all network parameters have a Dirichlet distribution. We assume parameter independence and parameter modularity, but instead of adopting the Dirichlet assumption, we introduce an assumption called *likelihood equivalence*, which says that data should not help to discriminate network structures that represent the same assertions of conditional independence. We argue that this property is necessary when learning acausal Bayesian networks and is often reasonable when learning causal Bayesian networks. We then show that likelihood equivalence, when combined with parameter independence and several weak conditions, implies the Dirichlet assumption. Furthermore, we show that likelihood equivalence constrains the Dirichlet distributions in such a way that they may be obtained from the user's prior network—a Bayesian network for the next case to be seen—and a single equivalent sample size reflecting the user's confidence in his prior network.

Our result has both a positive and negative aspect. On the positive side, we show that parameter independence, parameter modularity, and likelihood equivalence lead to a simple approach for assessing priors that requires the user to assess only one equivalent sample size for the entire domain. On the negative side, the approach is sometimes too simple: a user may have more knowledge about one part of a domain than another. We argue that the assumptions of parameter independence and likelihood equivalence are sometimes too strong, and suggest a framework for relaxing these assumptions.

A more straightforward task in learning Bayesian networks is using a given informative prior to compute $p(D, B_s^h|\xi)$ (i.e., a Bayesian scoring metric) and $p(C|D, B_s^h, \xi)$. When databases are complete—that is, when there is no missing data—these terms can be derived in closed form. Otherwise, well-known statistical approximations may be used. In this paper, we consider complete databases only, and derive closed-form expressions for these terms. A result is a likelihood-equivalent Bayesian scoring metric, which we call the BDe metric. This metric is to be contrasted with the metrics of CH and Buntine which do not make use of a prior network, and to the metrics of CH and SDLC which do not satisfy the property of likelihood equivalence.

In the second part of the paper (Section 7), we examine methods for finding networks with high scores. The methods can be used with any scoring metric. We describe polynomial algorithms for finding the highest-scoring networks in the special case where every node has at most one parent. In addition, we describe local-search and annealing algorithms for the general case, which is known to be NP-hard.

Finally, in Sections 8 and 9, we describe a methodology for evaluating learning algorithms. We use this methodology to compare various scoring metrics and search methods.

We note that several researchers (e.g., Dawid and Lauritzen, 1993, and Madigan and Raferty, 1994) have developed methods for learning undirected network structures as described in (e.g.) Lauritzen (1982). In this paper, we concentrate on learning directed models, because we can sometimes use them to infer causal relationships, and because most users find them easier to interpret.

2 Background

In this section, we introduce notation and background material that we need for our discussion, including a description of Bayesian networks, exchangeability, multinomial sampling, and the Dirichlet distribution. A summary of our notation is given after the Appendix on page 53.

Throughout this discussion, we consider a domain U of n discrete variables $x_1, \ldots, x_n$.

We use lower-case letters to refer to variables and upper-case letters to refer to sets of variables. We write $x_i = k$ to denote that variable x_i is in state k. When we observe the state for every variable in set X, we call this set of observations an *instance* of X; and we write $X = \vec{k}_X$ as a shorthand for the observations $x_i = k_i, x_i \in X$. The *joint space* of U is the set of all instances of U. We use $p(X = \vec{k}_X | Y = \vec{k}_Y, \xi)$ to denote the probability that $X = \vec{k}_X$ given $Y = \vec{k}_Y$ for a person with current state of information ξ. We use $p(X|Y, \xi)$ to denote the set of probabilities for all possible observations of X, given all possible observations of Y. The *joint probability distribution* over U is the probability distribution over the joint space of U.

A Bayesian network for domain U represents a joint probability distribution over U. The representation consists of a set of *local* conditional distributions combined with a set of conditional independence assertions that allow us to construct a global joint probability distribution from the local distributions. In particular, by the chain rule of probability, we have

$$p(x_1, \ldots, x_n|\xi) = \prod_{i=1}^{n} p(x_i|x_1, \ldots, x_{i-1}, \xi) \tag{2}$$

For each variable x_i, let $\Pi_i \subseteq \{x_1, \ldots, x_{i-1}\}$ be a set of variables that renders x_i and $\{x_1, \ldots, x_{i-1}\}$ conditionally independent. That is,

$$p(x_i|x_1, \ldots, x_{i-1}, \xi) = p(x_i|\Pi_i, \xi) \tag{3}$$

A *Bayesian-network structure* B_s encodes the assertions of conditional independence in Equations 3. Namely, B_s is a directed acyclic graph such that (1) each variable in U corresponds to a node in B_s, and (2) the parents of the node corresponding to x_i are the nodes corresponding to the variables in Π_i. (In this paper, we use x_i to refer to both the variable and its corresponding node in a graph.) A *Bayesian-network probability set* B_p is the collection of local distributions $p(x_i|\Pi_i, \xi)$ for each node in the domain. *A Bayesian network for U is the pair* (B_s, B_p). Combining Equations 2 and 3, we see that any Bayesian network for U uniquely determines a joint probability distribution for U. That is,

$$p(x_1, \ldots, x_n|\xi) = \prod_{i=1}^{n} p(x_i|\Pi_i, \xi) \tag{4}$$

When a variable has only two states, we say that it is binary. A Bayesian network for three binary variables x_1, x_2, and x_3 is shown in Figure 2. We see that $\Pi_1 = \emptyset, \Pi_2 = \{x_1\}$, and $\Pi_3 = \{x_2\}$. Consequently, this network represents the conditional-independence assertion $p(x_3|x_1, x_2, \xi) = p(x_3|x_2, \xi)$.

It can happen that two Bayesian-network structures represent the same constraints of conditional independence—that is, every joint probability distribution encoded by one structure can be encoded by the other, and vice versa. In this case, the two network structures are

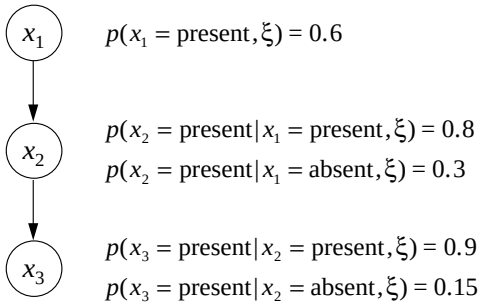

Figure 2: A Bayesian network for three binary variables (taken from CH). The network represents the assertion that x_1 and x_3 are conditionally independent given x_2. Each variable has states "absent" and "present."

said to be *equivalent* [Verma and Pearl, 1990]. For example, the structures $x_1 \rightarrow x_2 \rightarrow x_3$ and $x_1 \leftarrow x_2 \leftarrow x_3$ both represent the assertion that x_1 and x_3 are conditionally independent given x_2, and are equivalent. In some of the technical discussions in this paper, we shall require the following characterization of equivalent networks, proved in the Appendix.

Theorem 1 *Let B_{s1} and B_{s2} be two Bayesian-network structures, and $R_{B_{s1},B_{s2}}$ be the set of edges by which B_{s1} and B_{s2} differ in directionality. Then, B_{s1} and B_{s2} are equivalent if and only if there exists a sequence of $|R_{B_{s1},B_{s2}}|$ distinct arc reversals applied to B_{s1} with the following properties:*

1. *After each reversal, the resulting network structure contains no directed cycles and is equivalent to B_{s2}*
2. *After all reversals, the resulting network structure is identical to B_{s2}*
3. *If $x \rightarrow y$ is the next arc to be reversed in the current network structure, then x and y have the same parents in both network structures, with the exception that x is also a parent of y in B_{s1}*

A drawback of Bayesian networks as defined is that network structure depends on variable order. If the order is chosen carelessly, the resulting network structure may fail to reveal many conditional independencies in the domain. Fortunately, in practice, Bayesian networks are typically constructed using notions of cause and effect. Loosely speaking, to construct a Bayesian network for a given set of variables, we draw arcs from cause variables to their immediate effects. For example, we would obtain the network structure in

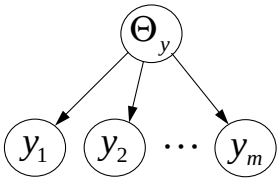

Figure 3: A Bayesian network showing the conditional-independence assertions associated with a multinomial sample.

Figure 2 if we believed that x_2 is the immediate causal effect of x_1 and x_3 is the immediate causal effect of x_2. In almost all cases, constructing a Bayesian network in this way yields a Bayesian network that is consistent with the formal definition. In Section 4 we return to this issue.

Now, let us consider exchangeability and random sampling. Most of the concepts we discuss can be found in Good (1965) and DeGroot (1970). Given a discrete variable y with r states, consider a finite sequence of observations $y_1, \ldots, y_m$ of this variable. We can think of this sequence as a database D for the one-variable domain $U = \{y\}$. This sequence is said to be *exchangeable* if a sequence obtained by interchanging any two observations in the sequence has the same probability as the original sequence. Roughly speaking, the assumption that a sequence is exchangeable is an assertion that the process(es) generating the data do not change in time.

Given an exchangeable sequence, De Finetti (1937) showed that there exists parameters $\Theta_y = \{\theta_{y=1}, \ldots, \theta_{y=r}\}$ such that

$$\theta_{y=k} > 0, k = 1, \ldots, r, \qquad \sum_{k=1}^{r} \theta_{y=k} = 1 \tag{5}$$

$$p(y_l = k | y_1, \ldots, y_{l-1}, \Theta_y, \xi) = \theta_{y=k} \tag{6}$$

That is, the parameters Θ_y render the individual observations in the sequence conditionally independent, and the probability that any given observation will be in state k is just $\theta_{y=k}$. The conditional independence assertion (Equation 6) may be represented as a Bayesian network, as shown in Figure 3. By the strong law of large numbers (e.g., DeGroot, 1970, p. 203), we may think of $\theta_{y=k}$ as the long-run fraction of observations where $y = k$, although there are other interpretations [Howard, 1988]. Also note that each parameter $\theta_{y=k}$ is positive (i.e., greater than zero).

A sequence that satisfies these conditions is a particular type of random sample known as an *r-dimensional multinomial sample with parameters* Θ_y [Good, 1965]. When $r = 2$, the

sequence is said to be a *binomial sample*. One example of a binomial sample is the outcome of repeated flips of a thumbtack. If we knew the long-run fraction of "heads" (point up) for a given thumbtack, then the outcome of each flip would be independent of the rest, and would have a probability of heads equal to this fraction. An example of a multinomial sample is the outcome of repeated rolls of a multi-sided die. As we shall see, learning Bayesian networks for discrete domains essentially reduces to the problem of learning the parameters of a die having many sides.

As Θ_y is a set of continuous variables, it has a probability density, which we denote $\rho(\Theta_y|\xi)$. Throughout this paper, we use $\rho(\cdot|\xi)$ to denote a probability density for a continuous variable or set of continuous variables. Given $\rho(\Theta_y|\xi)$, we can determine the probability that $y = k$ in the next observation. In particular, by the rules of probability we have

$$p(y = k|\xi) = \int p(y = k|\Theta_y, \xi)\ \rho(\Theta_y|\xi)\ d\Theta_y$$

Consequently, by condition 3 above, we obtain

$$p(y = k|\xi) = \int \theta_{y=k}\ \rho(\Theta_y|\xi)\ d\Theta_y \tag{7}$$

which is the mean or expectation of $\theta_{y=k}$ with respect to $\rho(\Theta_y|\xi)$, denoted $E(\theta_{y=k}|\xi)$.

Suppose we have a prior density for Θ_y, and then observe a database D. We may obtain the posterior density for Θ_y as follows. From Bayes' rule, we have

$$\rho(\Theta_y|D, \xi) = c \cdot p(D|\Theta_y, \xi)\rho(\Theta_y|\xi)$$

where c is a normalization constant. Using Equation 6 to rewrite the first term on the right hand side, we obtain

$$\rho(\Theta_y|D, \xi) = c \cdot \prod_{k=1}^{r} \theta_{y=k}^{N_k}\ \rho(\Theta_y|\xi) \tag{8}$$

where N_k is the number of times $x = k$ in D. Note that only the counts $N_1, \ldots, N_r$ are necessary to determine the posterior from the prior. These counts are said to be a *sufficient statistic* for the multinomial sample.

In addition, suppose we assess a density for two different states of information ξ_1 and ξ_2 and find that $\rho(\Theta_y|\xi_1) = \rho(\Theta_y|\xi_2)$. Then, for any multinomial sample D,

$$p(D|\xi_1) = \int p(D|\Theta_y, \xi_1)\ \rho(\Theta_y|\xi_1)\ d\Theta_y = p(D|\xi_2) \tag{9}$$

because $p(D|\Theta_y, \xi_1) = p(D|\Theta_y, \xi_2)$ by Equation 6. That is, if the densities for Θ_y are the same, then the probability of any two samples will be the same. The converse is also true. Namely, if $p(D|\xi_1) = p(D|\xi_2)$ for all databases D, then $\rho(\Theta_y|\xi_1) = \rho(\Theta_y|\xi_2)$.[1] We shall use this equivalence when we discuss likelihood equivalence.

[1] We assume this result is well known, although we haven't found a proof in the literature.

Given a multinomial sample, a user is free to assess any probability density for Θ_y. In practice, however, one often uses the Dirichlet distribution because it has several convenient properties. The parameters Θ_y *have a Dirichlet distribution with exponents* $N'_1, \ldots, N'_r$ when the probability density of Θ_y is given by

$$\rho(\Theta_y|\xi) = \frac{\Gamma(\sum_{k=1}^r N'_k)}{\prod_{k=1}^r \Gamma(N'_k)} \prod_{k=1}^r \theta_{y=k}^{N'_k-1}, \quad N'_k > 0 \tag{10}$$

where $\Gamma(\cdot)$ is the *Gamma* function, which satisfies $\Gamma(x+1) = x\Gamma(x)$ and $\Gamma(1) = 1$. When the parameters Θ_y have a Dirichlet distribution, we also say that $\rho(\Theta_y|\xi)$ *is Dirichlet.* The requirement that N'_k be greater than 0 guarantees that the distribution can be normalized. Note that the exponents N'_k are a function of the user's state of information ξ. Also note that, by Equation 5, the Dirichlet distribution for Θ_y is technically a density over $\Theta_y \backslash \{\theta_{y=k}\}$, for some k (the symbol $\backslash$ denotes set difference). Nonetheless, we shall write Equation 10 as shown. When $r = 2$, the Dirichlet distribution is also known as a beta distribution.

From Equation 8, we see that if the prior distribution of Θ_y is Dirichlet, then the posterior distribution of Θ_y given database D is also Dirichlet:

$$\rho(\Theta_y|D,\xi) = c \prod_{k=1}^r \theta_{y=k}^{N'_k+N_k-1}, \quad N'_i > 0 \tag{11}$$

where c is a normalization constant. We say that the Dirichlet distribution is closed under multinomial sampling, or that the Dirichlet distribution is a *conjugate family of distributions* for multinomial sampling. Also, when Θ_y has a Dirichlet distribution, the expectation of $\theta_{y=k_i}$—equal to the probability that $x = k_i$ in the next observation—has a simple expression:

$$E(\theta_{y=k}|\xi) = p(y=k|\xi) = \frac{N'_k}{N'} \tag{12}$$

where $N' = \sum_{k=1}^r N'_k$. We shall make use of these properties in our derivations.

A survey of methods for assessing a beta distribution is given by Winkler (1967). These methods include the direct assessment of the probability density using questions regarding relative densities and relative areas, assessment of the cumulative distribution function using fractiles, assessing the posterior means of the distribution given hypothetical evidence, and assessment in the form of an equivalent sample size. These methods can be generalized with varying difficulty to the nonbinary case.

In our work, we find one method based on Equation 12 particularly useful. The equation says that we can assess a Dirichlet distribution by assessing the probability distribution $p(y|\xi)$ for the next observation, and N'. In so doing, we may rewrite Equation 10 as

$$\rho(\Theta_y|\xi) = c \cdot \prod_{k=1}^r \theta_{y=k}^{N'p(y=k|\xi)-1} \tag{13}$$

where c is a normalization constant. Assessing $p(y|\xi)$ is straightforward. Furthermore, the following two observations suggest a simple method for assessing N'.

One, the variance of a density for Θ_y is an indication of how much the mean of Θ_y is expected to change, given new observations. The higher the variance, the greater the expected change. It is sometimes said that the variance is a measure of a user's *confidence* in the mean for Θ_y. The variance of the Dirichlet distribution is given by

$$Var(\theta_{y=k}|\xi) = \frac{p(y=k|\xi)(1-p(y=k|\xi))}{N'+1} \tag{14}$$

Thus, N' is a reflection of the user's confidence. Two, suppose we were initially completely ignorant about a domain—that is, our distribution $\rho(\Theta_y|\xi)$ was given by Equation 10 with each exponent $N'_k = 0$.[2] Suppose we then saw N' cases with sufficient statistics $N'_1, \ldots, N'_r$. Then, by Equation 11, our prior would be the Dirichlet distribution given by Equation 10.

Thus, we can assess N' as an *equivalent sample size*: the number of observations we would have had to have seen starting from complete ignorance in order to have the same confidence in Θ_y that we actually have. This assessment approach generalizes easily to many-variable domains, and thus is useful for our work. We note that some users at first find judgments of equivalent sample size to be difficult. Our experience with such users has been that they may be made more comfortable with the method by first using some other method for assessment (e.g., fractiles) on simple scenarios and by examining equivalent sample sizes implied by their assessments.

3 Bayesian Metrics: Previous Work

CH, Buntine, and SDLC examine domains where all variables are discrete and derive essentially the same Bayesian scoring metric and formula for $p(C|D, B_s^h, \xi)$ based on the same set of assumptions about the user's prior knowledge and the database. In this section, we present these assumptions and provide a derivation of $p(D, B_s^h|\xi)$ and $p(C|D, B_s^h, \xi)$.

Roughly speaking, the first assumption is that B_s^h is true iff the database D can be partitioned into a set of multinomial samples determined by the network structure B_s. In particular, B_s^h is true iff, for every variable x_i in U and every instance of x_i's parents Π_i in B_s, the observations of x_i in D in those cases where Π_i takes on the same instance constitute a multinomial sample. For example, consider a domain consisting of two binary variables x and y. (We shall use this domain to illustrate many of the concepts in this paper.) There are three network structures for this domain: $x \rightarrow y$, $x \leftarrow y$, and the empty

[2] This prior distribution cannot be normalized, and is sometimes called an *improper prior*. To be more precise, we should say that each exponent is equal to some number close to zero.

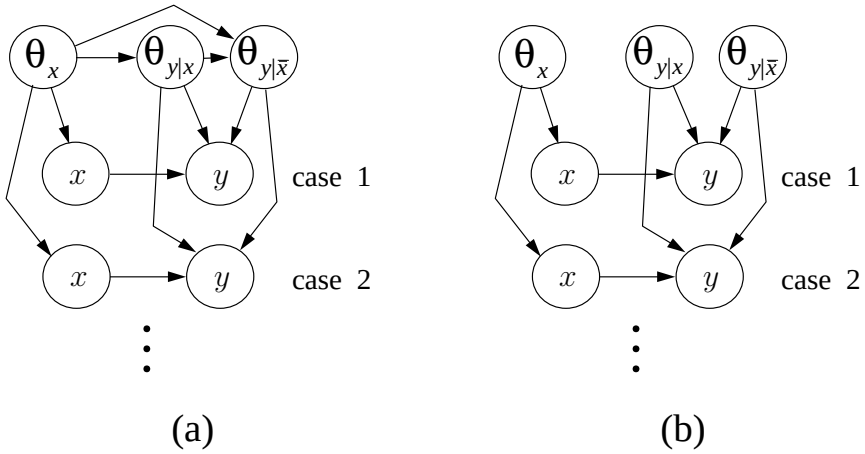

Figure 4: A Bayesian-network structure for a two-binary-variable domain $\{x, y\}$ showing conditional independencies associated with (a) the multinomial-sample assumption, and (b) the added assumption of parameter independence. In both figures, it is assumed that the network structure $x \rightarrow y$ is generating the database.

network structure containing no arc. The hypothesis associated with the empty network structure, denoted B^h_{xy}, corresponds to the assertion that the database is made up of two binomial samples: (1) the observations of x are a binomial sample with parameter θ_x, and (2) the observations of y are a binomial sample with parameter θ_y.

In contrast, the hypothesis associated with the network structure $x \rightarrow y$, denoted $B^h_{x \rightarrow y}$, corresponds to the assertion that the database is made up of at most three binomial samples: (1) the observations of x are a binomial sample with parameter θ_x, (2) the observations of y in those cases where x is true (if any) are a binomial sample with parameter $\theta_{y|x}$, and (3) the observations of y in those cases where x is false (if any) are a binomial sample with parameter $\theta_{y|\bar{x}}$. One consequence of the second and third assertions is that y in case C is conditionally independent of the other occurrences of y in D, given $\theta_{y|x}$, $\theta_{y|\bar{x}}$, and x in case C. We can graphically represent this conditional-independence assertion using a Bayesian-network structure as shown in Figure 4a.

Finally, the hypothesis associated with the network structure $x \leftarrow y$, denoted $B^h_{x \leftarrow y}$, corresponds to the assertion that the database is made up of at most three binomial samples: one for y, one for x given y is true, and one for x given y is false.

Before we state this assumption for arbitrary domains, we introduce the following

notation.[3] Given a Bayesian network B_s for domain U, let r_i be the number of states of variable x_i; and let $q_i = \prod_{x_l \in \Pi_i} r_l$ be the number of instances of Π_i. We use the integer j to index the instances of Π_i. Thus, we write $p(x_i = k|\Pi_i = j, \xi)$ to denote the probability that $x_i = k$, given the jth instance of the parents of x_i. Let θ_{ijk} denote the multinomial parameter corresponding to the probability $p(x_i = k|\Pi_i = j, \xi)$ $(\theta_{ijk} > 0, \sum_{k=1}^{r_i} \theta_{ijk} = 1)$. In addition, we define

$$\Theta_{ij} \equiv \cup_{k=1}^{r_i} \{\theta_{ijk}\}$$

$$\Theta_i \equiv \cup_{j=1}^{q_i} \{\Theta_{ij}\}$$

$$\Theta_{Bs} \equiv \cup_{i=1}^{n} \Theta_i$$

That is, the parameters in Θ_{Bs} correspond to the probability set B_p for a single-case Bayesian network.

Assumption 1 (Multinomial Sample) *Given domain U and database D, let D_l denote the first $l-1$ cases in the database. In addition, let x_{il} and Π_{il} denote the variable x_i and the parent set Π_i in the lth case, respectively. Then, for all network structures B_s in U, there exist positive parameters Θ_{Bs} such that, for $i = 1, \ldots, n$, and for all $k, k_1, \ldots, k_{i-1}$,*

$$p(x_{il} = k|x_{1l} = k_1, \ldots, x_{(i-1)l} = k_{i-1}, D_l, \Theta_{Bs}, B_s^h, \xi) = \theta_{ijk} \tag{15}$$

where j is the instance of Π_{il} consistent with $\{x_{1l} = k_1, \ldots, x_{(i-1)l} = k_{i-1}\}$.

There is an important implication of this assumption, which we examine in Section 4. Nonetheless, Equation 15 is all that we need (and all that CH, Buntine, and SDLC used) to derive a metric. Also, note that the positivity requirement excludes logical relationships among variables. We can relax this requirement, although we do not do so in this paper.

The second assumption is an independence assumption.

Assumption 2 (Parameter Independence) *Given network structure B_s, if $p(B_s^h|\xi) > 0$, then*

a. $\rho(\Theta_{Bs}|B_s^h, \xi) = \prod_{i=1}^{n} \rho(\Theta_i|B_s^h, \xi)$

b. For $i = 1, \ldots, n$: $\rho(\Theta_i|B_s^h, \xi) = \prod_{j=1}^{q_i} \rho(\Theta_{ij}|B_s^h, \xi)$

Assumption 2a says that the parameters associated with each variable in a network structure are independent. We call this assumption *global parameter independence* after Spiegelhalter and Lauritzen (1990). Assumption 2b says that the parameters associated with each

[3] Whenever possible we use CH's notation.

instance of the parents of a variable are independent. We call this assumption *local parameter independence*, again after Spiegelhalter and Lauritzen. We refer to the combination of these assumptions simply as parameter independence. The assumption of parameter independence for our two-binary-variable domain is shown in the Bayesian-network structure of Figure 4b.

As we shall see, Assumption 2 greatly simplifies the computation of $p(D, B_s^h|\xi)$. The assumption is reasonable for some domains, but not for others. In Section 5.6, we describe a simple characterization of the assumption that provides a test for deciding whether the assumption is reasonable in a given domain.

The third assumption was also made to simplify computations.

Assumption 3 (Parameter Modularity) *Given two network structures B_{s1} and B_{s2} such that $p(B_{s1}^h|\xi) > 0$ and $p(B_{s2}^h|\xi) > 0$, if x_i has the same parents in B_{s1} and B_{s2}, then*

$$\rho(\Theta_{ij}|B_{s1}^h, \xi) = \rho(\Theta_{ij}|B_{s2}^h, \xi) \qquad j = 1, \ldots, q_i$$

We call this property parameter modularity, because it says that the densities for parameters Θ_{ij} depend only on the structure of the network that is local to variable x_i—namely, Θ_{ij} only depends on x_i and its parents. For example, consider the network structure $x \rightarrow y$ and the empty structure for our two-variable domain. In both structures, x has the same set of parents (the empty set). Consequently, by parameter modularity, $\rho(\theta_x|B_{x\rightarrow y}^h, \xi) = \rho(\theta_x|B_{xy}^h, \xi)$. We note that CH, Buntine, and SDLC implicitly make the assumption of parameter modularity (Cooper and Herskovits, 1992, Equation A6, p. 340; Buntine, 1991, p. 55; Spiegelhalter et al., 1993, pp. 243-244).

The fourth assumption restricts each parameter set Θ_{ij} to have a Dirichlet distribution:

Assumption 4 (Dirichlet) *Given a network structure B_s such that $p(B_s^h|\xi) > 0$, $\rho(\Theta_{ij}|B_s^h, \xi)$ is Dirichlet for all $\Theta_{ij} \subseteq \Theta_{Bs}$. That is, there exists exponents N'_{ijk}, which depend on B_s^h and ξ, that satisfy*

$$\rho(\Theta_{ij}|B_s^h, \xi) = c \cdot \prod_k \theta_{ijk}^{N'_{ijk}-1}$$

where c is a normalization constant.

When every parameter set of B_s has a Dirichlet distribution, we simply say that $\rho(\Theta_{Bs}|B_s^h, \xi)$ is Dirichlet. Note that, by the assumption of parameter modularity, we do not require Dirichlet exponents for every network structure B_s. Rather we require exponents only for every node and for every possible parent set of each node.

Assumptions 1 through 4 are assumptions about the domain. Given Assumption 1, we can compute $p(D|\Theta_{Bs}, B_s^h, \xi)$ as a function of Θ_{Bs} for any given database (see Equation 18). Also, as we show in Section 5, Assumptions 2 through 4 determine $\rho(\Theta_{Bs}|B_s^h, \xi)$ for every network structure B_s. Thus, from the relation

$$p(D, B_s^h|\xi) = p(B_s^h|\xi) \int p(D|\Theta_{Bs}, B_s^h, \xi)\ \rho(\Theta_{Bs}|B_s^h, \xi)\ d\Theta_{Bs} \tag{16}$$

these assumptions in conjunction with the prior probabilities of network structure $p(B_s^h|\xi)$ form a complete representation of the user's prior knowledge for purposes of computing $p(D, B_s^h|\xi)$. By a similar argument, we can show that Assumptions 1 through 4 also determine the probability distribution $p(C|D, B_s^h, \xi)$ for any given database and network structure.

In contrast, the fifth assumption is an assumption about the database.

Assumption 5 (Complete Data) *The database is complete. That is, it contains no missing data.*

This assumption was made in order to compute $p(D, B_s^h|\xi)$ and $p(C|D, B_s^h, \xi)$ in closed form. In this paper, we concentrate on complete databases for the same reason. Nonetheless, the reader should recognize that, given Assumptions 1 through 4, these probabilities can be computed—in principle—for any complete or incomplete database. In practice, these probabilities can be approximated for incomplete databases by well-known statistical methods. Such methods include filling in missing data based on the data that is present [Titterington, 1976, Spiegelhalter and Lauritzen, 1990], the EM algorithm [Dempster et al., 1977], and Gibbs sampling (i.e., Markov chain Monte Carlo methods) [York, 1992, Madigan and Rafferty, 1994].

Let us now explore the consequences of these assumptions. First, from the multinomial-sample assumption and the assumption of no missing data, we obtain

$$p(C_l|D_l, \Theta_{Bs}, B_s^h, \xi) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{1_{lijk}} \tag{17}$$

where $1_{lijk} = 1$ if $x_i = k$ and $\Pi_i = j$ in case C_l, and $1_{lijk} = 0$ otherwise. Thus, if we let N_{ijk} be the number of cases in database D in which $x_i = k$ and $\Pi_i = j$, we have

$$p(D|\Theta_{Bs}, B_s^h, \xi) = \prod_i \prod_j \prod_k \theta_{ijk}^{N_{ijk}} \tag{18}$$

From this result, it follows that the parameters Θ_{Bs} remain independent given database D, a property we call *posterior parameter independence.* In particular, from the assumption of parameter independence, we have

$$\rho(\Theta_{Bs}|D, B_s^h, \xi) = c \cdot p(D|\Theta_{Bs}, B_s^h, \xi) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \rho(\Theta_{ij}|B_s^h, \xi) \tag{19}$$

where c is some normalization constant. Combining Equations 18 and 19, we obtain

$$\rho(\Theta_{Bs}|D, B_s^h, \xi) = c \cdot \prod_i \prod_j \left[\rho(\Theta_{ij}|B_s^h, \xi) \prod_k \theta_{ijk}^{N_{ijk}} \right] \tag{20}$$

and posterior parameter independence follows. We note that, by Equation 20 and the assumption of parameter modularity, parameters remain modular a posteriori as well.

Given these basic relations, we can derive a metric and a formula for $p(C|D, B_s^h, \xi)$. From the rules of probability, we have

$$p(D|B_s^h, \xi) = \prod_{l=1}^{m} p(C_l|D_l, B_s^h, \xi) \tag{21}$$

From this equation, we see that the Bayesian scoring metric can be viewed as a form of cross validation, where rather than use $D \setminus \{C_l\}$ to predict C_l, we use only cases $C_1, \ldots, C_{l-1}$ to predict C_l.

Conditioning on the parameters of the network structure B_s, we obtain

$$p(C_l|D_l, B_s^h, \xi) = \int p(C_l|D_l, \Theta_{Bs}, B_s^h, \xi) \cdot \rho(\Theta_{Bs}|D_l, B_s^h, \xi) \; d\Theta_{Bs} \tag{22}$$

Using Equation 17 and posterior parameter independence to rewrite the first and second terms in the integral, respectively, and interchanging integrals with products, we get

$$p(C_l|D_l, B_s^h, \xi) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \int \prod_{k=1}^{r_i} \theta_{ijk}^{1_{lijk}} \cdot \rho(\Theta_{ij}|D_l, B_s^h, \xi) \; d\Theta_{ij} \tag{23}$$

When $1_{lijk} = 1$, the integral is the expectation of θ_{ijk} with respect to the density $\rho(\Theta_{ij}|D_l, B_s^h, \xi)$. Consequently, we have

$$p(C_l|D_l, B_s^h, \xi) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} E(\theta_{ijk}|D_l, B_s^h, \xi)^{1_{lijk}} \tag{24}$$

To compute $p(C|D, B_s^h, \xi)$ we set $l = m + 1$ and interpret C_{m+1} to be C. To compute $p(D|B_s^h, \xi)$, we combine Equations 21 and 24 and rearrange products obtaining

$$p(D|B_s^h, \xi) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \prod_{l=1}^{m} E(\theta_{ijk}|C_1, \ldots, C_{l-1}, B_s^h, \xi)^{1_{lijk}} \tag{25}$$

Thus, all that remains is to determine to the expectations in Equations 24 and 25. Given the Dirichlet assumption (Assumption 4), this evaluation is straightforward. Combining the Dirichlet assumption and Equation 18, we obtain

$$\rho(\Theta_{ij}|D, B_s^h, \xi) = c \cdot \prod_k \theta_{ijk}^{N'_{ijk}+N_{ijk}-1} \tag{26}$$

where c is another normalization constant. Note that the counts N_{ijk} are a sufficient statistic for the database. Also, as we discussed in Section 2, the Dirichlet distributions are conjugate for the database: The posterior distribution of each parameter Θ_{ij} remains in the Dirichlet family. Thus, applying Equations 12 and 26 to Equation 24 with $l = m + 1$, $C_{m+1} = C$, and $D_{m+1} = D$, we obtain

$$p(C_{m+1}|D, B_s^h, \xi) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \left(\frac{N'_{ijk} + N_{ijk}}{N'_{ij} + N_{ij}} \right)^{1_{m+1,ijk}} \tag{27}$$

where

$$N'_{ij} \equiv \sum_{k=1}^{r_i} N'_{ijk} \qquad N_{ij} \equiv \sum_{k=1}^{r_i} N_{ijk}$$

Similarly, from Equation 25, we obtain the scoring metric

$$\begin{aligned} p(D, B_s^h|\xi) &= p(B_s^h|\xi) \cdot \prod_{i=1}^{n} \prod_{j=1}^{q_i} \left\{ \left[\frac{N'_{ij1}}{N'_{ij}} \cdot \frac{N'_{ij1}+1}{N'_{ij}+1} \cdots \frac{N'_{ij1}+N_{ij1}-1}{N'_{ij}+N_{ij1}-1} \right] \cdot \right. \\ &\quad \left[\frac{N'_{ij2}}{N'_{ij}+N_{ij1}} \cdot \frac{N'_{ij2}+1}{N'_{ij}+N_{ij1}+1} \cdots \frac{N'_{ij2}+N_{ij2}-1}{N'_{ij}+N_{ij1}+N_{ij2}-1} \right] \cdots \\ &\quad \left. \left[\frac{N'_{ijr_i}}{N'_{ij}+\sum_{k=1}^{r_i-1} N_{ijk}} \cdot \frac{N'_{ijr_i}+1}{N'_{ij}+\sum_{k=1}^{r_i-1} N_{ijk}+1} \cdots \frac{N'_{ijr_i}+N_{ijr_i}-1}{N_{ij}+N'_{ij}-1} \right] \right\} \\ &= p(B_s^h|\xi) \cdot \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(N'_{ij})}{\Gamma(N'_{ij}+N_{ij})} \cdot \prod_{k=1}^{r_i} \frac{\Gamma(N'_{ijk}+N_{ijk})}{\Gamma(N'_{ijk})} \end{aligned} \tag{28}$$

We call Equation 28 the BD (*B*ayesian *D*irichlet) metric.

As is apparent from Equation 28, the exponents N'_{ijk} in conjunction with $p(B_s^h|\xi)$ completely specify a user's current knowledge about the domain for purposes of learning network structures. Unfortunately, the specification of N'_{ijk} for all possible variable–parent configurations and for all values of i, j, and k is formidable, to say the least. CH suggest a simple uninformative assignment $N'_{ijk} = 1$. We shall refer to this special case of the BD metric as the K2 metric. Buntine (1991) suggests the uninformative assignment $N'_{ijk} = N'/(r_i \cdot q_i)$. We shall examine this special case again in Section 5.2. In Section 6, we address the assessment of the priors on network structure $p(B_s^h|\xi)$.

4 Acausal networks, causal networks, and likelihood equivalence

In this section, we examine another assumption for learning Bayesian networks that has been previously overlooked.

Before we do so, it is important to distinguish between acausal and causal Bayesian networks. Although Bayesian networks have been formally described as a representation of conditional independence, as we noted in Section 2, people often construct them using notions of cause and effect. Recently, several researchers have begun to explore a formal causal semantics for Bayesian networks (e.g., Pearl and Verma, 1991, Pearl, 1995, Spirtes et al., 1993, Druzdzel and Simon, 1993, and Heckerman and Shachter, 1994). They argue that the representation of causal knowledge is important not only for assessment, but for prediction as well. In particular, they argue that causal knowledge—unlike statistical knowledge—allows one to derive beliefs about a domain after intervention. For example, most of us believe that smoking causes lung cancer. From this knowledge, we infer that if we stop smoking, then we decrease our chances of getting lung cancer. In contrast, if we knew only that there was a statistical correlation between smoking and lung cancer, then we could not make this inference. The formal semantics of cause and effect proposed by these researchers is not important for this discussion. The interested reader should consult the references given.

First, let us consider acausal networks. Recall our assumption that the hypothesis B_s^h is true iff the database D is a collection of multinomial samples determined by the network structure B_s. This assumption is equivalent to saying that (1) the database D is a multinomial sample from the joint space of U with parameters Θ_U, and (2) the hypothesis B_s^h is true iff the parameters Θ_U satisfy the conditional-independence assertions of B_s. We can think of condition 2 as a definition of the hypothesis B_s^h.

For example, in our two-binary-variable domain, regardless of which hypothesis is true, we may assert that the database is a multinomial sample from the joint space $U = \{x, y\}$ with parameters $\Theta_U = \{\theta_{xy}, \theta_{x\bar{y}}, \theta_{\bar{y}x}, \theta_{\bar{x}\bar{y}}\}$. Furthermore, given the hypothesis $B_{x\rightarrow y}^h$—for example—we know that the parameters Θ_U are unconstrained (except that they must sum to one), because the network structure $x \rightarrow y$ represents no assertions of conditional independence. In contrast, given the hypothesis B_{xy}^h, we know that the parameters Θ_U must satisfy the independence constraints $\theta_{xy} = \theta_x\theta_y$, $\theta_{x\bar{y}} = \theta_x\theta_{\bar{y}}$, and so on.

Given this definition of B_s^h for acausal Bayesian networks, it follows that if two network structures B_{s1} and B_{s2} are equivalent, then $B_{s1}^h = B_{s2}^h$. For example, in our two-variable domain, both the hypotheses $B_{x\rightarrow y}^h$ and $B_{x\leftarrow y}^h$ assert that there are no constraints on the parameters Θ_U. Consequently, we have $B_{x\rightarrow y}^h = B_{x\leftarrow y}^h$. In general, we call this property *hypothesis equivalence.*[4]

[4]One technical flaw with the definition of B_s^h is that hypotheses are not mutually exclusive. For example, in our two-variable domain, the hypotheses B_{xy}^h and $B_{x\rightarrow y}^h$ both include the possibility $\theta_y = \theta_{y|x}$. This flaw is potentially troublesome, because mutual exclusivity is important for our Bayesian interpretation of

In light of this property, we should associate each hypothesis with an equivalence class of structures rather than a single network structure. Also, given the property of hypothesis equivalence, we have *prior equivalence:* if network structures B_{s1} and B_{s2} are equivalent, then $p(B_{s1}^h|\xi) = p(B_{s2}^h|\xi)$; *likelihood equivalence:* if B_{s1} and B_{s2} are equivalent, then for all databases D, $p(D|B_{s1}^h, \xi) = p(D|B_{s2}^h, \xi)$; and *score equivalence:* if B_{s1} and B_{s2} are equivalent, then $p(D, B_{s1}^h|\xi) = p(D, B_{s2}^h|\xi)$.

Now, let us consider causal networks. For these networks, the assumption of hypothesis equivalence is unreasonable. In particular, for causal networks, we must modify the definition of B_s^h to include the assertion that each nonroot node in B_s is a direct causal effect of its parents. For example, in our two-variable domain, the causal networks $x \rightarrow y$ and $x \leftarrow y$ represent the same constraints on Θ_U (i.e., none), but the former also asserts that x causes y, whereas the latter asserts that y causes x. Thus, the hypotheses $B_{x\rightarrow y}^h$ and $B_{x\leftarrow y}^h$ are not equal. Indeed, it is reasonable to assume that these hypotheses—and the hypotheses associated with any two different causal-network structures—are mutually exclusive.

Nonetheless, for many real-world problems that we have encountered, we have found it reasonable to assume likelihood equivalence. That is, we have found it reasonable to assume that data cannot distinguish between equivalent network structures. Of course, for any given problem, it is up to the decision maker to assume likelihood equivalence or not. In Section 5.6, we describe a characterization of likelihood equivalence that suggests a simple procedure for deciding whether the assumption is reasonable in a given domain.

Because the assumption of likelihood equivalence is appropriate for learning acausal networks in all domains and for learning causal networks in many domains, we adopt this assumption in our remaining treatment of scoring metrics. As we have stated it, likelihood equivalence says that, for any database D, the probability of D is the same given hypotheses corresponding to any two equivalent network structures. From our discussion surrounding Equation 9, however, we may also state likelihood equivalence in terms of Θ_U:

Assumption 6 (Likelihood Equivalence) *Given two network structures B_{s1} and B_{s2} such that $p(B_{s1}^h|\xi) > 0$ and $p(B_{s2}^h|\xi) > 0$, if B_{s1} and B_{s2} are equivalent, then $\rho(\Theta_U|B_{s1}^h, \xi) = \rho(\Theta_U|B_{s2}^h, \xi)$.*[5]

We close this section with a few additional remarks about inferring causal relationships.

network learning (see Equation 1). Nonetheless, because the densities $\rho(\Theta_{Bs}|B_s^h, \xi)$ must be integrable and hence bounded, the overlap of hypotheses will be of measure zero, and we may use Equation 1 without modification. For example, in our two-binary-variable domain, given the hypothesis $B_{x\rightarrow y}^h$, the probability that B_{xy}^h is true (i.e., $\theta_y = \theta_{y|x}$) has measure zero.

[5]Using the same convention as for the Dirichlet distribution, we write $\rho(\Theta_U|B_s^h, \xi)$ to denote a density over a set of the nonredundant parameters in Θ_U.

Given the distinction between statistical and causal dependence, it would seem impossible to learn causal networks from data produced by observation alone. For example, consider the simple three-variable domain $U = \{x_1, x_2, x_3\}$. If we find through the observation of data that the network structure $x_1 \rightarrow x_3 \leftarrow x_2$ is very likely, then we cannot conclude that x_1 and x_2 are causes for x_3. Rather, it may be the case that there is a hidden common cause of x_1 and x_3 as well as a hidden common cause of x_2 and x_3. If, however, we assume that every statistical association derives from causal interaction, and that there are no hidden common causes, then we can interpret learned networks as causal networks. In our example, under these assumptions, we can infer that x_1 and x_2 are causes for x_3.[6]

Under the assumption of likelihood equivalence, the ratio of posterior probabilities of two equivalent network structures must be equal to the ratio of their prior probabilities. Consequently, if the priors on network structures are not too different, then typically, learning will produce many equivalent network structures each having a large relative posterior probability. Furthermore, even for domains where the assumption of likelihood equivalence does not hold, there is a good chance that more than one hypothesis will have a large relative posterior probability. In such situations, we find it reasonable to average the causal assertions contained in individual learned networks. For example, in our three-variable domain, let us suppose that the data supports only the network structure $x_1 \rightarrow x_2 \rightarrow x_3$ and its equivalent cousins $x_1 \leftarrow x_2 \leftarrow x_3$ and $x_1 \leftarrow x_2 \rightarrow x_3$. If each of the hypotheses corresponding to these structures have the same prior probability, then the posterior probability of each hypothesis will be 1/3, and we infer that the proposition x_2 causes x_3 has probability 2/3. Under these same conditions, the proposition that both x_1 and x_3 are causes of x_2 has probability 0.

5 The BDe Metric

The assumption of likelihood equivalence when combined the previous assumptions introduces constraints on the Dirichlet exponents N'_{ijk}. The result is a likelihood-equivalent specialization of the BD metric, which we call the BDe metric. In this section, we derive this metric. In addition, we show that, as a consequence of the exponent constraints, the user may construct an informative prior for the parameters of all network structures merely by building a Bayesian network for the next case to be seen and by assessing an equivalent sample size. Most remarkable, we show that Dirichlet assumption (Assumption 4) is not needed to obtain the BDe metric.

[6] Note that, in some circumstances, we can identify causes and effects from network structure even when there are hidden common causes. See Pearl (1995) for a discussion.

5.1 Informative Priors

In this section, we show how the added assumption of likelihood equivalence simplifies the construction of informative priors.

Before we do so, we need to define the concept of a complete network structure. A *complete network structure* is one that has no missing edges—that is, it encodes no assertions of conditional independence. In a domain with n variables, there are $n!$ complete network structures. An important property of complete network structures is that all such structures for a given domain are equivalent.

Now, for a given domain U, suppose we have assessed the density $\rho(\Theta_U|B^h_{sc},\xi)$, where B_{sc} is some complete network structure for U. Given parameter independence, parameter modularity, likelihood equivalence, and one additional assumption, it turns out that we can compute the prior $\rho(\Theta_{Bs}|B^h_s,\xi)$ for *any* network structure B_s in U from the given density.

To see how this computation is done, consider again our two-binary-variable domain. Suppose we are given a density for the parameters of the joint space $\rho(\theta_{xy},\theta_{\bar{x}y},\theta_{x\bar{y}}|B^h_{x\to y},\xi)$. From this density, we construct the parameter densities for each of the three network structures in the domain. First, consider the network structure $x \to y$. A parameter set for this network structure is $\{\theta_x,\theta_{y|x},\theta_{y|\bar{x}}\}$. These parameters are related to the parameters of the joint space by the following relations:

$$\theta_{xy} = \theta_x\theta_{y|x} \qquad \theta_{\bar{x}y} = (1-\theta_x)(\theta_{y|\bar{x}}) \qquad \theta_{x\bar{y}} = \theta_x(1-\theta_{y|x})$$

Thus, we may obtain $\rho(\theta_x,\theta_{y|x},\theta_{y|\bar{x}}|B^h_{x\to y},\xi)$ from the given density by changing variables:

$$\rho(\theta_x,\theta_{y|x},\theta_{y|\bar{x}}|B^h_{x\to y},\xi) = J_{x\to y}\cdot\rho(\theta_{xy},\theta_{\bar{x}y},\theta_{x\bar{y}}|B^h_{x\to y},\xi) \tag{29}$$

where $J_{x\to y}$ is the Jacobian of the transformation

$$J_{x\to y} = \begin{vmatrix} \partial\theta_{xy}/\partial\theta_x & \partial\theta_{\bar{x}y}/\partial\theta_x & \partial\theta_{x\bar{y}}/\partial\theta_x \\ \partial\theta_{xy}/\partial\theta_{y|x} & \partial\theta_{\bar{x}y}/\partial\theta_{y|x} & \partial\theta_{x\bar{y}}/\partial\theta_{y|x} \\ \partial\theta_{xy}/\partial\theta_{y|\bar{x}} & \partial\theta_{\bar{x}y}/\partial\theta_{y|\bar{x}} & \partial\theta_{x\bar{y}}/\partial\theta_{y|\bar{x}} \end{vmatrix} = \theta_x(1-\theta_x) \tag{30}$$

The Jacobian $J_{B_{sc}}$ for the transformation from Θ_U to Θ_{Bsc}, where B_{sc} is an arbitrary complete network structure, is given in the Appendix (Theorem 12).

Next, consider the network structure $x \leftarrow y$. Assuming that the hypothesis $B^h_{x\leftarrow y}$ is also possible, we obtain $\rho(\theta_{xy},\theta_{\bar{x}y},\theta_{x\bar{y}}|B^h_{x\leftarrow y},\xi) = \rho(\theta_{xy},\theta_{\bar{x}y},\theta_{x\bar{y}}|B^h_{x\to y},\xi)$ by likelihood equivalence. Therefore, we can compute the density for the network structure $x \leftarrow y$ using the Jacobian $J_{x\leftarrow y} = \theta_y(1-\theta_y)$.

Finally, consider the empty network structure. Given the assumption of parameter independence, we may obtain the densities $\rho(\theta_x|B^h_{xy},\xi)$ and $\rho(\theta_y|B^h_{xy},\xi)$ separately. To

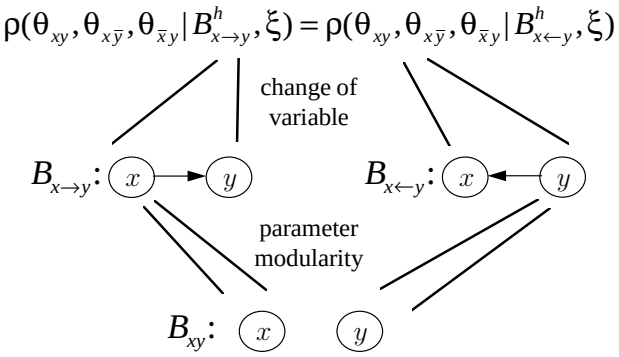

Figure 5: A computation of the parameter densities for the three network structures of the two-binary-variable domain $\{x, y\}$. The approach computes the densities from $\rho(\theta_{xy}, \theta_{x\bar{y}}, \theta_{\bar{x}y}|B^h_{x\to y}, \xi)$, using likelihood equivalence, parameter independence, and parameter modularity.

obtain the density for θ_x, we first extract $\rho(\theta_x|B^h_{x\to y}, \xi)$ from the density for the network structure $x \to y$. This extraction is straightforward, because by parameter independence, the parameters for $x \to y$ must be independent. Then, we use parameter modularity, which says that $\rho(\theta_x|B^h_{xy}, \xi) = \rho(\theta_x|B^h_{x\to y}, \xi)$. To obtain the density for θ_y, we extract $\rho(\theta_y|B^h_{x\leftarrow y}, \xi)$ from the density for the network structure $x \leftarrow y$, and again apply parameter modularity. The approach is summarized in Figure 5.

In this construction, it is important that both hypotheses $B^h_{x\to y}$ and $B^h_{x\leftarrow y}$ have nonzero prior probabilities, lest we could not make use of likelihood equivalence to obtain the parameter densities for the empty structure. In order to take advantage of likelihood equivalence in general, we adopt the following assumption.

Assumption 7 (Structure Possibility) *Given a domain U, $p(B^h_{sc}|\xi) > 0$ for all complete network structures B_{sc}.*

Note that, in the context of acausal Bayesian networks, there is only one hypothesis corresponding to the equivalence class of complete network structures. In this case, Assumption 7 says that this single hypothesis is possible. In the context of causal Bayesian networks, the assumption implies that each of the $n!$ complete network structures is possible. Although we make the assumption of structure possibility as a matter of convenience, we have found it to be reasonable in many real-world network-learning problems.

Given this assumption, we can now describe our construction method in general.

Theorem 2 *Given domain U and a probability density $\rho(\Theta_U|B^h_{sc},\xi)$ where B_{sc} is some complete network structure for U, the assumptions of parameter independence (Assumption 2), parameter modularity (Assumption 3), likelihood equivalence (Assumption 6), and structure possibility (Assumption 7) uniquely determine $\rho(\Theta_{Bs}|B^h_s,\xi)$ for any network structure B_s in U.*

Proof: Consider any B_s. By Assumption 2, if we determine $\rho(\Theta_{ij}|B^h_s,\xi)$ for every parameter set Θ_{ij} associated with B_s, then we determine $\rho(\Theta_{Bs}|B^h_s,\xi)$. So consider a particular Θ_{ij}. Let Π_i be the parents of x_i in B_s, and $B_{sc'}$ be a complete belief-network structure with variable ordering Π_i, x_i followed by the remaining variables. First, using Assumption 7, we recognize that the hypothesis $B^h_{sc'}$ is possible. Consequently, we use Assumption 6 to obtain $\rho(\Theta_U|B^h_{sc'},\xi) = \rho(\Theta_U|B^h_{sc},\xi)$. Next, we change variables from Θ_U to $\Theta_{Bsc'}$ yielding $\rho(\Theta_{Bsc'}|B^h_{sc'},\xi)$. Using parameter independence, we then extract the density $\rho(\Theta_{ij}|B^h_{sc'},\xi)$ from $\rho(\Theta_{Bsc'}|B^h_{sc'},\xi)$. Finally, because x_i has the same parents in B_s and $B_{sc'}$, we apply parameter modularity to obtain the desired density: $\rho(\Theta_{ij}|B^h_s,\xi) = \rho(\Theta_{ij}|B^h_{sc'},\xi)$. To show uniqueness, we note that the only freedom we have in choosing $B_{sc'}$ is that the parents of x_i can be shuffled with one another and nodes following x_i in the ordering can be shuffled with one another. The Jacobian of the change-of-variable from Θ_U to $\Theta_{Bsc'}$ has the same terms in Θ_{ij} regardless of our choice. □

5.2 Consistency and the BDe Metric

In our procedure for generating priors, we cannot use an arbitrary density $\rho(\Theta_U|B^h_{sc},\xi)$. In our two-variable domain, for example, suppose we use the density

$$\rho(\theta_{xy},\theta_{\bar{x}y},\theta_{x\bar{y}}|B^h_{x\to y},\xi) = \frac{c}{(\theta_{xy}+\theta_{x\bar{y}})(1-(\theta_{xy}+\theta_{x\bar{y}}))} = \frac{c}{\theta_x(1-\theta_x)}$$

where c is a normalization constant. Then, using Equations 29 and 30, we obtain

$$\rho(\theta_x,\theta_{y|x},\theta_{y|\bar{x}}|B^h_{x\to y},\xi) = c$$

for the network structure $x \to y$, which satisfies parameter independence and the Dirichlet assumption. For the network structure $y \to x$, however, we have

$$\rho(\theta_y,\theta_{x|y},\theta_{x|\bar{y}}|B^h_{x\leftarrow y},\xi) = \frac{c\cdot\theta_y(1-\theta_y)}{\theta_x(1-\theta_x)} = \frac{c\cdot\theta_y(1-\theta_y)}{(\theta_y\theta_{x|y}+(1-\theta_y)\theta_{x|\bar{y}})(1-(\theta_y\theta_{x|y}+(1-\theta_y)\theta_{x|\bar{y}}))}$$

This density satisfies neither parameter independence nor the Dirichlet assumption.

In general, if we do not choose $\rho(\Theta_U|B^h_{sc},\xi)$ carefully, we may not satisfy both parameter independence and the Dirichlet assumption. Indeed, the question arises: Is there any

choice for $\rho(\Theta_U|B_{sc}^h,\xi)$ that is consistent with these assumptions? The following theorem and corollary answers this question in the affirmative. (In the remainder of Section 5, we require additional notation. We use $\theta_{X=\vec{k}_X|Y=\vec{k}_Y}$ to denote the multinomial parameter corresponding to the probability $p(X=\vec{k}_X|Y=\vec{k}_Y,\xi)$. X and Y may be single variables, and $\vec{k}_X$ and $\vec{k}_Y$ are often implicit. Also, we use $\Theta_{X|Y=\vec{k}_Y}$ to denote the set of multinomial parameters corresponding to the probability distribution $p(X|Y=\vec{k}_Y,\xi)$, and $\Theta_{X|Y}$ to denote the parameters $\Theta_{X|Y=\vec{k}_Y}$ for all instances of $\vec{k}_Y$. When Y is empty, we omit the conditioning bar.)

Theorem 3 *Given a domain $U=\{x_1,\ldots,x_n\}$ with multinomial parameters Θ_U, if the density $\rho(\Theta_U|\xi)$ is Dirichlet—that is, if*

$$\rho(\Theta_U|\xi) = c\cdot\prod_{x_1,\ldots,x_n}[\theta_{x_1,\ldots,x_n}]^{N'_{x_1,\ldots,x_n}-1} \tag{31}$$

then, for any complete network structure B_{sc} in U, the density $\rho(\Theta_{Bsc}|\xi)$ is Dirichlet and satisfies parameter independence. In particular,

$$\rho(\Theta_{Bsc}|\xi) = c\cdot\prod_{i=1}^{n}\prod_{x_1,\ldots,x_i}[\theta_{x_i|x_1,\ldots,x_{i-1}}]^{N'_{x_i|x_1,\ldots,x_{i-1}}-1} \tag{32}$$

where

$$N'_{x_i|x_1,\ldots,x_{i-1}} = \sum_{x_{i+1},\ldots,x_n} N'_{x_1,\ldots,x_n} \tag{33}$$

Proof: Let B_{sc} be any complete network structure for U. Reorder the variables in U so that the ordering matches this structure, and relabel the variables $x_1,\ldots,x_n$. Now, change variables from $\Theta_{x_1,\ldots,x_n}$ to Θ_{Bsc} using the Jacobian given by Theorem 12. The dimension of this transformation is $[\prod_{i=1}^n r_i]-1$ where r_i are the number of instances of x_i. Substituting the relationship $\theta_{x_1,\ldots,x_n}=\prod_{i=1}^n\theta_{x_i|x_1,\ldots,x_{i-1}}$, and multiplying with the Jacobian, we obtain

$$\rho(\Theta_{Bsc}|\xi) = c\cdot\left\{\prod_{x_1,\ldots,x_n}\left[\prod_{i=1}^{n}\theta_{x_i|x_1,\ldots,x_{i-1}}\right]^{N'_{x_1,\ldots,x_n}-1}\right\}\cdot\left\{\prod_{i=1}^{n-1}\prod_{x_1,\ldots,x_i}[\theta_{x_i|x_1,\ldots,x_{i-1}}]^{[\prod_{j=i}^n r_j]-1}\right\}$$

which implies Equation 32. Collecting the powers of $\theta_{x_i|x_1,\ldots,x_{i-1}}$, and using $\prod_{j=i+1}^n r_j = \sum_{x_{i+1},\ldots,x_n}1$, we obtain Equation 33. □

Corollary 4 *Let U be a domain with multinomial parameters Θ_U, and B_{sc} be a complete network structure for U such that $p(B_{sc}^h|\xi)>0$. If $\rho(\Theta_U|B_{sc}^h,\xi)$ is Dirichlet, then $\rho(\Theta_{Bsc}|B_{sc}^h,\xi)$ is Dirichlet and satisfies parameter independence.*

Given these results, we can compute the Dirichlet exponents N'_{ijk} using a Dirichlet distribution for $\rho(\Theta_U|B^h_{sc},\xi)$ in conjunction with our method for constructing priors described in Theorem 2. Namely, suppose we desire the exponent N'_{ijk} for a network structure where x_i has parents Π_i. Let $B_{sc'}$ be a complete network structure where x_i has these parents. By likelihood equivalence, we have $\rho(\Theta_U|B^h_{sc'},\xi) = \rho(\Theta_U|B^h_{sc},\xi)$. As we discussed in Section 2, we may write the exponents for $\rho(\Theta_U|B^h_{sc},\xi)$ as follows:

$$N'_{x_1,\ldots,x_n} = N'\ p(x_1,\ldots,x_n|B^h_{sc},\xi) \tag{34}$$

where N' is the user's equivalent sample size for the $\rho(\Theta_U|B^h_{sc},\xi)$. Furthermore, by definition, N'_{ijk} is the Dirichlet exponent for θ_{ijk} in $B_{sc'}$. Consequently, from Equations 33 and 34, we have

$$N'_{ijk} = N'\ p(x_i = k, \Pi_i = j|B^h_{sc},\xi)$$

We call the BD metric with this restriction on the exponents the BDe metric ("e" for likelihood *e*quivalence). To summarize, we have the following theorem.

Theorem 5 (BDe Metric) *Given domain U, suppose that $\rho(\Theta_U|B^h_{sc},\xi)$ is Dirichlet with equivalent sample size N' for some complete network structure B_{sc} in U. Then, for any network structure B_s in U, Assumptions 1 through 3 and 5 through 7 imply*

$$p(D, B^h_s|\xi) = p(B^h_s|\xi) \cdot \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(N'_{ij})}{\Gamma(N'_{ij}+N_{ij})} \cdot \prod_{k=1}^{r_i} \frac{\Gamma(N'_{ijk}+N_{ijk})}{\Gamma(N'_{ijk})}$$

where

$$N'_{ijk} = N'\ p(x_i = k, \Pi_i = j|B^h_{sc},\xi) \tag{35}$$

Theorem 3 shows that parameter independence, likelihood equivalence, structure possibility, and the Dirichlet assumption are consistent for complete network structures. Nonetheless, these assumptions and the assumption of parameter modularity may not be consistent for all network structures. To understand the potential for inconsistency, note that we obtained the BDe metric for all network structures using likelihood equivalence applied only to complete network structures in combination with the other assumptions. Thus, it could be that the BDe metric for incomplete network structures is not likelihood equivalent. Nonetheless, the following theorem shows that the BDe metric is likelihood equivalent for all network structures—that is, given the other assumptions, likelihood equivalence for incomplete structures is implied by likelihood equivalence for complete network structures. Consequently, our assumptions are consistent.

Theorem 6 *For all domains U and all network structures B_s in U, the BDe metric is likelihood equivalent.*

Proof: Given a database D, equivalent sample size N', joint probability distribution $p(U|B_{sc}^h, \xi)$, and a subset X of U, consider the following function of X:

$$l(X) = \prod_{\vec{k}_X} \Gamma\left(N'\, p(X = \vec{k}_X | B_{sc}^h, \xi) + N_{\vec{k}_X}\right)$$

where $\vec{k}_X$ is an instance of X, and $N_{\vec{k}_X}$ is the number of cases in D in which $X = \vec{k}_X$. Then, the likelihood term of the BDe metric becomes

$$p(D|B_s^h, \xi) = \prod_{i=1}^{n} \frac{l(\{x_i\} \cup \Pi_i)}{l(\Pi_i)} \tag{36}$$

Now, by Theorem 1, we know that a network structure can be transformed into an equivalent structure by a series of arc reversals. Thus, we can demonstrate that BDe metric satisfies likelihood equivalence in general, if we can do so for the case where two equivalent structures differ by a single arc reversal. So, let B_{s1} and B_{s2} be two equivalent network structures that differ only in the direction of the arc between x_i and x_j (say $x_i \rightarrow x_j$ in B_{s1}). Let R be the set of parents of x_i in B_{s1}. By Theorem 1, we know that $R \cup \{x_i\}$ is the set of parents of x_j in B_{s1}, R is the set of parents of x_j in B_{s2}, and $R \cup \{x_j\}$ is the set of parents of x_i in B_{s2}. Because the two structures differ only in the reversal of a single arc, the only terms in the product of Equation 36 that can differ are those involving x_i and x_j. For B_{s1}, these terms are

$$\frac{l(\{x_i\} \cup R)}{l(R)} \frac{l(\{x_i, x_j\} \cup R)}{l(\{x_i\} \cup R)}$$

whereas for B_{s2}, they are

$$\frac{l(\{x_j\} \cup R)}{l(R)} \frac{l(\{x_i, x_j\} \cup R)}{l(\{x_j\} \cup R)}$$

These terms are equal, and hence $p(D|B_{s1}^h, \xi) = p(D|B_{s2}^h, \xi)$. □

We note that Buntine's (1991) metric is a special case of the BDe metric where every instance of the joint space, conditioned on B_{sc}^h, is equally likely. We call this special case the BDeu metric ("u" for *u*niform joint distribution). Buntine noted that this metric satisfies the property of likelihood equivalence.

5.3 The Prior Network

To calculate the terms in the BDe metric (or to construct informative priors for a more general metric that can handle missing data), we need priors on network structures $p(B_s^h|\xi)$ and the Dirichlet distribution $\rho(\Theta_U|B_{sc}^h, \xi)$. In Section 6, we provide a simple method for assessing priors on network structures. Here, we concentrate on the assessment of the Dirichlet distribution for Θ_U.

Recall from Sections 2 and 5.2 that we can assess this distribution by assessing a single equivalent sample size N' for the domain and the joint distribution of the domain for the next case to be seen ($p(U|B_{sc}^h, \xi)$), where both assessments are conditioned on the state of information $B_{sc}^h \cup \xi$. As we have discussed, the assessment of equivalent sample size is straightforward. Furthermore, a user can assess $p(U|B_{sc}^h, \xi)$ by building a Bayesian network for U given B_{sc}^h. We call this network the user's *prior network.*

The unusual aspect of this assessment is the conditioning hypothesis B_{sc}^h. Whether we are dealing with acausal or causal Bayesian networks, this hypothesis includes the assertion that there are no independencies in the long run. Thus, at first glance, there seems to be a contradiction in asking the user to construct a prior network—which may contain assertions of independence—under the assertion that B_{sc}^h is true. Nonetheless, there is no contradiction, because the assertions of independence in the prior network refer to independencies in the next case to be seen, whereas the assertion of full dependence B_{sc}^h refers to the long run.

To help illustrate this point, let us consider the following acausal example. Suppose a person repeatedly rolls a four-sided die with labels $1, 2, 3$, and 4. In addition, suppose that he repeatedly does one of the following: (1) rolls the die once and reports "x = true" iff the die lands 1 or 2, and "y = true" iff the die lands 1 or 3, or (2) rolls the die twice and reports "x = true" iff the die lands 1 or 2 on the first roll and reports "y = true" iff the die lands 1 or 3 on the second roll. In either case, the multinomial assumption is reasonable. Furthermore, condition 2 corresponds to the hypothesis B_{xy}^h: x and y are independent in the long run, whereas condition 1 corresponds to the hypothesis $B_{x \to y}^h = B_{x \leftarrow y}^h$: x and y are dependent in the long run.[7] Also, given these correspondences, parameter modularity and likelihood equivalence are reasonable. Finally, let us suppose that the parameters of the multinomial sample have a Dirichlet distribution so that parameter independence holds. Thus, this example fits the assumptions of our learning approach. Now, if we have no reason to prefer one outcome of the die to another on the next roll, then we will have $p(y|x, B_{x \to y}^h, \xi) = p(y|B_{x \to y}^h, \xi)$. That is, our prior network will contain no arc between x and y, even though, given $B_{x \to y}^h$, x and y are almost certainly dependent in the long run.

We expect that most users would prefer to construct a prior network without having to condition on B_{sc}^h. In the previous example, it is possible to ignore the conditioning hypothesis, because $p(U|B_{x \to y}^h, \xi) = p(U|B_{xy}^h, \xi) = p(U|\xi)$. In general, however, a user cannot ignore this hypothesis. In our four-sided die example, the joint distributions $p(U|B_{x \to y}^h, \xi)$ and $p(U|B_{xy}^h, \xi)$ would have been different had we not been indifferent about the die outcomes.

[7] Actually, as we have discussed, $B_{x \to y}^h$ includes the possibility that x and y are independent, but only with a probability of measure zero.

We have had little experience with training people to condition on B^h_{sc} when constructing a prior network. Nonetheless, stories like the four-side die may help users make the necessary distinction for assessment.

5.4 A Simple Example

Consider again our two-binary-variable domain. Let $B_{x\to y}$ and $B_{y\to x}$ denote the network structures where x points to y and y points to x, respectively. Suppose that $N' = 12$ and that the user's prior network gives the joint distribution $p(x, y|B^h_{x\to y}, \xi) = 1/4, p(x, \bar{y}|B^h_{x\to y}, \xi) = 1/6, p(\bar{x}, y|B^h_{x\to y}, \xi) = 1/4$ and $p(\bar{x}, \bar{y}|B^h_{x\to y}, \xi) = 1/3$. Also, suppose we observe two cases: $C_1 = \{x, y\}$ and $C_2 = \{x, \bar{y}\}$. Let $i = 1$ (2) refer to variable x (y), and $k = 1$ (2) denote the true (false) state of a variable. Thus, for the network structure $x \to y$, we have the Dirichlet exponents $N'_{111} = 5$, $N'_{112} = 7$, $N'_{211} = 3$, $N'_{212} = 2$, $N'_{221} = 3$, and $N'_{222} = 4$, and the sufficient statistics $N_{111} = 2, N_{112} = 0, N_{211} = 1, N_{212} = 1, N_{221} = 0$, and $N_{222} = 0$. Consequently, we obtain

$$p(D|B^h_{x\to y}, \xi) = \frac{11!}{13!}\frac{6!}{4!}\frac{6!}{6!} \cdot \frac{4!}{6!}\frac{3!}{2!}\frac{2!}{1!} \cdot \frac{6!}{6!}\frac{2!}{2!}\frac{3!}{3!} = \frac{1}{26}$$

For the network structure $x \leftarrow y$, we have the Dirichlet exponents $N'_{111} = 3$, $N'_{112} = 3$, $N'_{121} = 2$, $N'_{122} = 4$, $N'_{211} = 6$, and $N'_{212} = 6$, and the sufficient statistics $N_{111} = 1, N_{112} = 0, N_{121} = 1, N_{122} = 0, N_{211} = 1$, and $N_{212} = 1$. Consequently, we have

$$p(D|B^h_{x\leftarrow y}, \xi) = \frac{11!}{13!}\frac{6!}{5!}\frac{6!}{5!} \cdot \frac{5!}{6!}\frac{3!}{2!}\frac{2!}{2!} \cdot \frac{5!}{6!}\frac{2!}{1!}\frac{3!}{3!} = \frac{1}{26}$$

As required, the BDe metric exhibits the property of likelihood equivalence.

In contrast, the K2 metric (all $N'_{ijk} = 1$) does not satisfy this property. In particular, given the same database, we have

$$p(D|B^h_{x\to y}, \xi) = \frac{1!}{3!}\frac{2!}{0!}\frac{0!}{0!} \cdot \frac{1!}{3!}\frac{1!}{0!}\frac{1!}{0!} \cdot \frac{1!}{1!}\frac{0!}{0!}\frac{0!}{0!} = \frac{1}{18}$$

$$p(D|B^h_{x\leftarrow y}, \xi) = \frac{1!}{3!}\frac{1!}{0!}\frac{1!}{0!} \cdot \frac{1!}{2!}\frac{1!}{0!}\frac{0!}{0!} \cdot \frac{1!}{2!}\frac{1!}{0!}\frac{0!}{0!} = \frac{1}{24}$$

5.5 Elimination of the Dirichlet Assumption

In Section 5.2, we saw that when $\rho(\Theta_U|B^h_{sc}, \xi)$ is Dirichlet, then $\rho(\Theta_{Bsc}|B^h_{sc}, \xi)$ is consistent with parameter independence, the Dirichlet assumption, likelihood equivalence, and structure possibility. Therefore, it is natural to ask whether there are any other choices for $\rho(\Theta_U|B^h_{sc}, \xi)$ that are similarly consistent. Actually, because the Dirichlet assumption is so strong, it is more fitting to ask whether there are any other choices for $\rho(\Theta_U|B^h_{sc}, \xi)$ that are

consistent with all but the Dirichlet assumption. In this section, we show that, if each density function is positive (i.e., the range of each function includes only numbers greater than zero), then a Dirichlet distribution for $\rho(\Theta_U|B^h_{sc},\xi)$ is the only consistent choice. Consequently, we show that, under these conditions, the BDe metric follows without the Dirichlet assumption.

First, let us examine this question for our two-binary-variable domain. Combining Equations 29 and 30 for the network structure $x \rightarrow y$, the corresponding equations for the network structure $x \leftarrow y$, likelihood equivalence, and structure possibility, we obtain

$$\rho(\theta_x, \theta_{y|x}, \theta_{y|\bar{x}}|B^h_{x\rightarrow y},\xi) = \frac{\theta_x(1-\theta_x)}{\theta_y(1-\theta_y)}\ \rho(\theta_y, \theta_{x|y}, \theta_{x|\bar{y}}|B^h_{x\leftarrow y},\xi) \tag{37}$$

where

$$\begin{aligned} \theta_y &= \theta_x\theta_{y|x} + (1-\theta_x)\theta_{y|\bar{x}} \\ \theta_{x|y} &= \frac{\theta_{xy}}{\theta_x\theta_{y|x}+(1-\theta_x)\theta_{y|\bar{x}}} \\ \theta_{x|\bar{y}} &= \frac{\theta_{x\bar{y}}}{1-(\theta_x\theta_{y|x}+(1-\theta_x)\theta_{y|\bar{x}})} \end{aligned} \tag{38}$$

Applying parameter independence to both sides of Equation 37, we get

$$f_x(\theta_x)f_{y|x}(\theta_{y|x})f_{y|\bar{x}}(\theta_{y|\bar{x}}) = \frac{\theta_x(1-\theta_x)}{\theta_y(1-\theta_y)}f_y(\theta_y)f_{x|y}(\theta_{x|y})f_{x|\bar{y}} \tag{39}$$

where f_x, $f_{y|x}$, $f_{y|\bar{x}}$, f_y, $f_{x|y}$, and $f_{x|\bar{y}}$ are unknown density functions. Equations 38 and 39 define a *functional equation*. Methods for solving such equations have been well studied (see, e.g., Aczel, 1966). In our case, Geiger and Heckerman (1995) show that, if each function is positive, then the only solution to Equations 38 and 39 is for $\rho(\theta_{xy}, \theta_{x\bar{y}}\theta_{\bar{x}y}|B^h_{x\rightarrow y},\xi)$ to be a Dirichlet distribution. In fact, they show that even when x and/or y have more than two states, the only solution consistent with likelihood equivalence is the Dirichlet.

Theorem 7 (Geiger and Heckerman, 1995) *Let Θ_{xy}, $\Theta_x \cup \Theta_{y|x}$, and $\Theta_y \cup \Theta_{x|y}$ be (positive) multinomial parameters related by the rules of probability. If*

$$f_x(\Theta_x)\ \prod_{k=1}^{r_x} f_{y|x=k}(\Theta_{y|x=k}) = \left[\frac{\prod_{k=1}^{r_x}\theta_{x=k}^{r_y-1}}{\prod_{l=1}^{r_y}\theta_{y=l}^{r_x-1}}\right] f_y(\Theta_y)\ \prod_{l=1}^{r_y} f_{x|y=l}(\Theta_{x|y=l}) \tag{40}$$

where each function is a positive probability density function, then $\rho(\Theta_{xy}|\xi)$ is Dirichlet.

This result for two variables is easily generalized to the n-variable case, as we now demonstrate.

Theorem 8 *Let B_{sc1} and B_{sc2} be two complete network structures for U with variable orderings $(x_1, \ldots, x_n)$ and $(x_n, x_1, \ldots, x_{n-1})$, respectively. If both structures have (positive) multinomial parameters that obey*

$$\rho(\Theta_{Bsc}|\xi) = J_{B_{sc}} \cdot \rho(\Theta_U|\xi) \tag{41}$$

and positive densities $\rho(\Theta_{Bsc}|\xi)$ that satisfy parameter independence, then $\rho(\Theta_U|\xi)$ is Dirichlet.

Proof: The theorem is trivial for domains with one variable ($n = 1$), and is proved by Theorem 7 for $n = 2$. When $n > 2$, first consider the complete network structure B_{sc1}. Clustering the variables $X = \{x_1, \ldots, x_{n-1}\}$ into a single discrete variable with $q = \prod_{i=1}^{n-1} r_i$ states, we obtain the network structure $X \rightarrow x_n$ with multinomial parameters Θ_X and $\Theta_{x_n|X}$ given by

$$\theta_X = \prod_{i=1}^{n-1} \theta_{x_i|x_1,\ldots,x_{i-1}}$$
$$\theta_{x_n|X} = \theta_{x_n|x_1,\ldots,x_{n-1}}$$

By assumption, the parameters of B_{sc1} satisfy parameter independence. Thus, when we change variables from Θ_{Bsc1} to $\Theta_X \cup \Theta_{x_n|X}$ using the Jacobian given by Theorem 12, we find that the parameters for $X \rightarrow x_n$ also satisfy parameter independence. Now, consider the complete network structure B_{sc2}. With the same variable cluster, we obtain the network structure $x_n \rightarrow X$ with parameters Θ_{x_n} (as in the original network structure) and $\Theta_{X|x_n}$ given by

$$\theta_{X|x_n} = \prod_{i=1}^{n-1} \theta_{x_i|x_n,x_1,\ldots,x_{i-1}}$$

By assumption, the parameters of B_{sc2} satisfy parameter independence. Thus, when we change variables from Θ_{Bsc2} to $\Theta_{x_n} \cup \Theta_{X|x_n}$ (computing a Jacobian for each state of x_n), we find that the parameters for $x_n \rightarrow X$ again satisfy parameter independence. Finally, these changes of variable in conjunction with Equation 41 imply Equation 40. Consequently, by Theorem 7, $\rho(\Theta_{X,x_n}|B^h_{sc}, \xi) = \rho(\Theta_U|B^h_{sc}, \xi)$ is Dirichlet. □

Thus, we obtain the BDe metric without the Dirichlet assumption.

Theorem 9 *Assumptions 1 through 7—excluding the Dirichlet assumption (Assumption 4)—and the assumption that parameter densities are positive imply the BDe metric (Equations 28 and 35).*

Proof: Given parameter independence, likelihood equivalence, structure possibility, and positive densities, we have from Theorem 8 that $\rho(\Theta_U|B^h_{sc}, \xi)$ is Dirichlet. Thus, from Theorem 5, we obtain the BDe metric. □

The assumption that parameters are positive is important. For example, given a domain consisting of only logical relationships, we can have parameter independence, likelihood equivalence, and structure possibility, and yet $\rho(\Theta_U|B_{sc}^h, \xi)$ will not be Dirichlet.

5.6 Limitations of Parameter Independence and Likelihood Equivalence

There is a simple characterization of the assumption of parameter independence. Recall the property of posterior parameter independence, which says that parameters remain independent as long as complete cases are observed. Thus, suppose we have an uninformative Dirichlet prior for the joint-space parameters (all exponents very close to zero), which satisfies parameter independence. Then, if we observe one ore more complete cases, our posterior will also satisfy parameter independence. In contrast, suppose we have the same uninformative prior, and observe one or more incomplete cases. Then, our posterior will not be a Dirichlet distribution (in fact, it will be a linear combination of Dirichlet distributions) and will not satisfy parameter independence. In this sense, the assumption of parameter independence corresponds to the assumption that one's knowledge is equivalent to having seen only complete cases.

When learning causal Bayesian networks, there is a similar characterization of the assumption of likelihood equivalence. (Recall that, when learning acausal networks, the assumption must hold.) Namely, until now, we have considered only *observational data*: data obtained without intervention. Nonetheless, in many real-world studies, we obtain *experimental data*: data obtain by intervention—for example, by randomizing subjects into control and experimental groups. Although we have not developed the concepts in this paper to demonstrate the assertion, it turns out that if we start with the uninformative Dirichlet prior (which satisfies likelihood equivalence), then the posterior will satisfy likelihood equivalence if and only if we see no experimental data. In this sense, when learning causal Bayesian networks, the assumption of likelihood equivalence corresponds to the assumption that one's knowledge is equivalent to having seen only nonexperimental data.[8]

In light of these characterizations, we see that the assumptions of parameter independence and likelihood equivalence are unreasonable in many domains. For example, if we learn about a portion of domain by reading or through word of mouth, or simply apply common sense, then these assumptions should be suspect. In these situations, our methodology for determining an informative prior from a prior network and a single equivalent sample size is too simple.

To relax one or both of these assumptions when they are unreasonable, we can use

[8] These characterizations of parameter independence and likelihood equivalence, in the context of causal networks, are simplified for this presentation. Heckerman (1995) provides more detailed characterizations.

an *equivalent database* in place of an equivalent sample size. Namely, we ask a user to imagine that he was initially completely ignorant about a domain, having an uninformative Dirichlet prior. Then, we ask the user to specify a database D_e that would produce a posterior density that reflects his current state of knowledge. This database may contain incomplete cases and/or experimental data. Then, to score a real database D, we score the database $D_e \cup D$, using the uninformative prior and a learning algorithm that handles missing and experimental data such as Gibbs sampling.

It remains to be determined if this approach is practical. Needed is a compact representation for specifying equivalent databases that allows a user to accurately reflect his current knowledge. One possibility is to allow a user to specify a prior Bayesian network along with equivalent sample sizes (both experimental and nonexperimental) for each variable. Then, one could repeatedly sample equivalent databases from the prior network that satisfy these sample-size constraints, compute desired quantities (such as a scoring metric) from each equivalent database, and then average the results.

SDLC suggest a different method for accommodating nonuniform equivalent sample sizes. Their method produces Dirichlet priors that satisfy parameter independence, but not likelihood equivalence.

6 Priors for Network Structures

To complete the information needed to derive a Bayesian metric, the user must assess the prior probabilities of the network structures. Although these assessments are logically independent of the assessment of the prior network, structures that closely resemble the prior network will tend to have higher prior probabilities. Here, we propose the following parametric formula for $p(B_s^h|\xi)$ that makes use of the prior network.

Let δ_i denote the number of nodes in the symmetric difference of $\Pi_i(B_s)$ and $\Pi_i(P)$: $(\Pi_i(B_s) \cup \Pi_i(P)) \setminus (\Pi_i(B_s) \cap \Pi_i(P))$. Then B_s and the prior network differ by $\delta = \sum_{i=1}^{n} \delta_i$ arcs; and we penalize B_s by a constant factor $0 < \kappa \leq 1$ for each such arc. That is, we set

$$p(B_s^h|\xi) = c\ \kappa^{\delta} \tag{42}$$

where c is a normalization constant, which we can ignore when computing relative posterior probabilities. This formula is simple, as it requires only the assessment of a single constant κ. Nonetheless, we can imagine generalizing the formula by punishing different arc differences with different weights, as suggested by Buntine. Furthermore, it may be more reasonable to use a prior network constructed without conditioning on B_{sc}^h.

We note that this parametric form satisfies prior equivalence only when the prior network contains no arcs. Consequently, because the priors on network structures for acausal

networks must satisfy prior equivalence, we should not use this parameterization for acausal networks.

7 Search Methods

In this section, we examine methods for finding network structures with high posterior probabilities. Although our methods are presented in the context of Bayesian scoring metrics, they may be used in conjunction with other nonBayesian metrics as well. Also, we note that researchers have proposed network-selection criteria other than relative posterior probability (e.g., Madigan and Raferty, 1994), which we do not consider here.

Many search methods for learning network structure—including those that we describe—make use of a property of scoring metrics that we call decomposability. Given a network structure for domain U, we say that a measure on that structure is *decomposable* if it can be written as a product of measures, each of which is a function only of one node and its parents. From Equation 28, we see that the likelihood $p(D|B_s^h, \xi)$ given by the BD metric is decomposable. Consequently, if the prior probabilities of network structures are decomposable, as is the case for the priors given by Equation 42, then the BD metric will be decomposable. Thus, we can write

$$p(D, B_s^h|\xi) = \prod_{i=1}^{n} s(x_i|\Pi_i) \tag{43}$$

where $s(x_i|\Pi_i)$ is only a function of x_i and its parents. Given a decomposable metric, we can compare the score for two network structures that differ by the addition or deletion of arcs pointing to x_i, by computing only the term $s(x_i|\Pi_i)$ for both structures. We note that most known Bayesian and nonBayesian metrics are decomposable.

7.1 Special-Case Polynomial Algorithms

We first consider the special case of finding the l network structures with the highest score among all structures in which every node has at most one parent.

For each arc $x_j \rightarrow x_i$ (including cases where x_j is null), we associate a weight $w(x_i, x_j) \equiv \log s(x_i|x_j) - \log s(x_i|\emptyset)$. From Equation 43, we have

$$\begin{aligned} \log p(D, B_s^h|\xi) &= \sum_{i=1}^{n} \log s(x_i|\pi_i) \\ &= \sum_{i=1}^{n} w(x_i, \pi_i) + \sum_{i=1}^{n} \log s(x_i|\emptyset) \end{aligned} \tag{44}$$

where π_i is the (possibly) null parent of x_i. The last term in Equation 44 is the same for all network structures. Thus, among the network structures in which each node has at most one parent, ranking network structures by sum of weights $\sum_{i=1}^{n} w(x_i, \pi_i)$ or by score has the same result.

Finding the network structure with the highest weight ($l = 1$) is a special case of a well-known problem of finding *maximum branchings* described—for example—in Evans and Minieka (1991). The problem is defined as follows. A *tree-like network* is a connected directed acyclic graph in which no two edges are directed into the same node. The root of a tree-like network is a unique node that has no edges directed into it. A *branching* is a directed forest that consists of disjoint tree-like networks. A *spanning branching* is any branching that includes all nodes in the graph. A *maximum branching* is any spanning branching which maximizes the sum of arc weights (in our case, $\sum_{i=1}^{n} w(x_i, \pi_i)$). An efficient polynomial algorithm for finding a maximum branching was first described by Edmonds (1967), later explored by Karp (1971), and made more efficient by Tarjan (1977) and Gabow et al. (1984). The general case ($l > 1$) was treated by Camerini et al. (1980).

These algorithms can be used to find the l branchings with the highest weights regardless of the metric we use, as long as one can associate a weight with every edge. Therefore, this algorithm is appropriate for any decomposable metric. When using metrics that are score equivalent (i.e., both prior and likelihood equivalent), however, we have

$$s(x_i|x_j)s(x_j|\emptyset) = s(x_j|x_i)s(x_i|\emptyset)$$

Thus, for any two edges $x_i \rightarrow x_j$ and $x_i \leftarrow x_j$, the weights $w(x_i, x_j)$ and $w(x_j, x_i)$ are equal. Consequently, the directionality of the arcs plays no role for score-equivalent metrics, and the problem reduces to finding the l undirected forests for which $\sum w(x_i, x_j)$ is a maximum. For the case $l = 1$, we can apply a maximum spanning tree algorithm (with arc weights $w(x_i, x_j)$) to identify an undirected forest F having the highest score. The set of network structures that are formed from F by adding any directionality to the arcs of F such that the resulting network is a branching yields a collection of equivalent network structures each having the same maximal score. This algorithm is identical to the tree learning algorithm described by Chow and Liu (1968), except that we use a score-equivalent Bayesian metric rather than the mutual-information metric. For the general case ($l > 2$), we can use the algorithm of Gabow (1977) to identify the l undirected forests having the highest score, and then determine the l equivalence classes of network structures with the highest score.

7.2 Heuristic Search

A generalization of the problem described in the previous section is to find the l best networks from the set of all networks in which each node has no more than k parents. Unfortunately, even when $l = 1$, the problem for $k > 1$ is NP-hard. In particular, let us consider the following decision problem, which corresponds to our optimization problem with $l = 1$:

k-LEARN
INSTANCE: Set of variables U, database $D = \{C_1, \ldots, C_m\}$, where each C_i is an instance of all variables in U, scoring metric $M(D, B_s)$ and real value p.
QUESTION: Does there exist a network structure B_s defined over the variables in U, where each node in B_s has at most k parents, such that $M(D, B_s) \geq p$?

Höffgen (1993) shows that a similar problem for PAC learning is NP-complete. His results can be translated easily to show that k-LEARN is NP-complete for $k > 1$ when the BD metric is used. Chickering et al. (1995) show that k-LEARN is NP-complete, even when we use the likelihood-equivalent BDe metric and the constraint of prior equivalence.

Therefore, it is appropriate to use heuristic search algorithms for the general case $k > 1$. In this section, we review several such algorithms.

As is the case with essentially all search methods, the methods that we examine have two components: an initialization phase and a search phase. For example, let us consider the K2 search method (not to be confused with the K2 metric) described by CH. The initialization phase consists of choosing an ordering over the variables in U. In the search phase, for each node x_i in the ordering provided, the node from $\{x_i, \ldots, x_{i-1}\}$ that most increases the network score is added to the parent set of x_i, until no node increases the score or the size of Π_i exceeds a predetermined constant.

The search algorithms we consider make successive arc changes to the network, and employ the property of decomposability 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 contain 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 decomposable metric, if an arc to x_i is added or deleted, only $s(x_i|\Pi_i)$ need be evaluated to determine $\Delta(e)$. If an arc between x_i and x_j is reversed, then only $s(x_i|\Pi_i)$ and $s(x_j|\Pi_j)$ need be evaluated.

One simple heuristic search algorithm is *local search* [Johnson, 1985]. First, we choose a graph. 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 $\Delta(e)$. As we visit network structures, we retain l of them with the highest overall score. Using decomposable metrics, 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, a graph determined by one of the polynomial algorithms described in the previous section, and the prior network.

A potential problem with local search is getting stuck at a local maximum. Methods for avoiding local maxima include iterated hill-climbing and simulated annealing. In *iterated hill-climbing*, we apply local search until we hit a local maximum. Then, we randomly perturb the current network structure, and repeat the process for some manageable number of iterations. At all stages we retain the top l networks structures.

In one variant of *simulated annealing* described by Metropolis et al. (1953), we initialize the system to 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. Throughout the process, we retain the top l structures. 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.

Other methods for avoiding local maxima include best-first search [Korf, 1993] and Gibbs' sampling (e.g., Madigan and Raferty, 1994).

8 Evaluation Methodology

Our methodology for measuring the learning accuracy of scoring metrics and search procedures is as follows. We start with a given network, which we call the *gold-standard network*. Next, we generate a database D by repeated sampling from the given network. Then, we

use a Bayesian metric and search procedure to identify one or more network structures with high relative posterior probabilities or by some other criteria. We call these network structures the *learned networks.* Then, we use Equation 1 in conjunction with Equation 27 and the network scores to approximate $p(C|D,\xi)$, the joint probability distribution of the next case given the database. Finally, we quantitate learning accuracy by measuring the difference between the joint probability distribution of the gold-standard and $p(C|D,\xi)$.

A principled candidate for a measure of learning accuracy is expected utility. Namely, given a utility function, a series of decisions to be made under uncertainty, and a model of that uncertainty (i.e., one ore more Bayesian networks for U), we evaluate the expected utility of these decisions using the gold-standard and learned networks, and note the difference [Heckerman and Nathwani, 1992]. This utility function may include not only domain utility, but the costs of probabilistic inference as well [Horvitz, 1987]. Unfortunately, it is difficult if not impossible to construct utility functions and decision scenarios in practice. For example, a particular set of learned network structures may be used for a collection of decisions problems, some of which cannot be anticipated. Consequently, researchers have used surrogates for differences in utility, such as the mean square error, cross entropy, and differences in structure.

In this paper, we use two surrogate measures: cross-entropy and a structural difference. The cross-entropy measure [Kullback and Leibler, 1951] reflects how well the learned structures will predict the next case, whereas structural difference reflects the degree to which the learned structures have captured causal relationships.

Our cross-entropy measure is as follows. Let $p(U)$ denote the joint distribution of the gold-standard domain and $q(U)$ denote the joint distribution of the next case to be seen as predicted by the learned networks (i.e., $p(C|D,\xi)$). The cross entropy $H(p,q)$ is given by

$$H(p,q) = \sum_{x_1,\ldots,x_n} p(x_i,\ldots,x_n)\ \log \frac{p(x_i,\ldots,x_n)}{q(x_i,\ldots,x_n)} \tag{45}$$

Low values of cross entropy correspond to a learned distribution that is close to the gold standard. To evaluate Equation 45 efficiently, we construct a network structure B_s that is consistent with both the gold-standard and learned networks, using the algorithm described by Matzkevich and Abramson (1993). Next, we encode the joint distribution of the gold-standard and learned networks in this structure. Then, we compute the cross entropy, using the relation

$$H(p,q) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} p(x_i = k, \Pi_i = j)\ \log \frac{p(x_i = k|\Pi_i = j)}{q(x_i = k|\Pi_i = j)} \tag{46}$$

As mentioned, our structural-difference measure is intended to reflect the degree to which the learned structures have captured causal interactions. For a single learned network, the

structural difference we use is $\sum_{i=1}^{n} \delta_i$, where δ_i is the symmetric difference of the parents of x_i in the gold-standard network and the parents of x_i in the learned network. For multiple networks, we take the average of the structural-difference scores, weighted by the relative posterior probabilities of the learned networks.

SDLC describe an alternative evaluation method that does not make use of a gold-standard network. An advantage of our approach is that there exists a clear correct answer: the gold-standard network. Consequently, our method will always detect overfitting. One drawback of our approach is that, for small databases, it cannot discriminate a bad learning algorithm from a good learning algorithm applied to misleading data. Another problem with our method is that, by generating a database from a network, we guarantee that the assumption of exchangeability (time invariance) holds, and thereby bias results in favor of our scoring metrics. We can, however, simulate time varying databases in order to measure the sensitivity of our methods to the assumption of exchangeability (although we do not do so in this paper).

In several of our experiments described in the next section, we require a prior network. For these investigations, we construct prior networks by adding noise to the gold-standard network. We control the amount of noise with a parameter η. When $\eta = 0$, the prior network is identical to the gold-standard network, and as η increases, the prior network diverges from the gold-standard network. When η is large enough, the prior network and gold-standard networks are unrelated. Let (B_s, B_p) denote the gold-standard network. To generate the prior network, we first add 2η arcs to B_s, creating network structure B_{s1}. When we add an arc, we copy the probabilities from B_p to B_{p1} so as to maintain the same joint probability distribution for U. Next, we perturb each conditional probability in B_{p1} with noise. In particular, we convert each probability to log odds, add to it a sample from a normal distribution with mean zero and standard deviation η, convert the result back to a probability, and renormalize the probabilities. Then, we create another network structure B_{s2} by deleting 2η arcs and reversing 2η arcs that were present in the original gold-standard network. Next, we perform inference using the joint distribution determined by network (B_{s1}, B_{p1}) to populate the conditional probabilities for network (B_{s2}, B_{p2}). For example, if x has parents Y in B_{s1}, but x is a root node in B_{s2}, then we compute the marginal probability for x in B_{s1}, and store it with node x in B_{s2}. Finally, we return (B_{s2}, B_{p2}) as the prior network.

9 Experimental Results

We have implemented the metrics and search algorithms described in this paper. Our implementation is written in the C++ programming language, and runs under Windows NT^{TM} with a 90MHz Pentium processor. We have tested our algorithms on small networks ($n \leq 5$) as well as the Alarm network shown in Figure 1a. Here, we describe some of the more interesting results that we obtained using the Alarm network.

Figure 1 in the introduction shows an example learning cycle for the Alarm domain. The database was generated by sampling 10,000 cases from the Alarm network. The prior network was generated with $\eta = 2$. The learned network was the most likely network structure found using the BDe metric with $N' = 64$, and $\kappa = 1/17$, and local search initialized with the prior network. (The choice of these constants will become clear.)

To examine the effects of scoring metrics on learning accuracy, we measured the cross entropy and structural difference of learned networks with respect to the Alarm network for several variants of the BDe metric as well as the K2 metric. The results are shown in Figure 6. The metrics labeled BDe0, BDe2, and BDe4 correspond to the BDe metric with prior networks generated from the Alarm network with noise $\eta = 0, 2, 4$, respectively. In this comparison, we used local search initialized with a maximum branching and 10,000-case databases sampled from the Alarm network. For each value of equivalent sample size N' in the graph, the cross-entropy and structural-difference values shown in the figure represent an average across five learning instances, where in each instance we used a different database, and (for the BDe2 and BDe4 metrics) a different prior network. We made the prior parameter κ a function of N'—namely, $\kappa = 1/(N' + 1)$—so that it would take on reasonable values at the extremes of N'. (When $N' = 0$, reflecting complete ignorance, all network structures receive the same prior probability. Whereas, when N' is large, reflecting a high degree of confidence, the prior network structure receives a high prior probability.) When computing the prior probabilities of network structure for the K2 metric, we used Equation 42 with an empty prior network.

The qualitative behaviors of the BDe metrics were reasonable. When $\eta = 0$—that is, when the prior network was identical to the Alarm network—learning accuracy increased as the equivalent sample size N' increased. Also, learning accuracy decreased as the prior network deviated further from the gold-standard network, demonstrating the expected result that prior knowledge is useful. In addition, When $\eta \neq 0$, there was a value of N' associated with optimal accuracy, and this value decreased as η increased ($N' = 64$ for $\eta = 2$; $N' = 16$ for $\eta = 4$). This result is not surprising. Namely, if N' is too large, then the deviation between the true values of the parameters and their priors degrade performance. On the

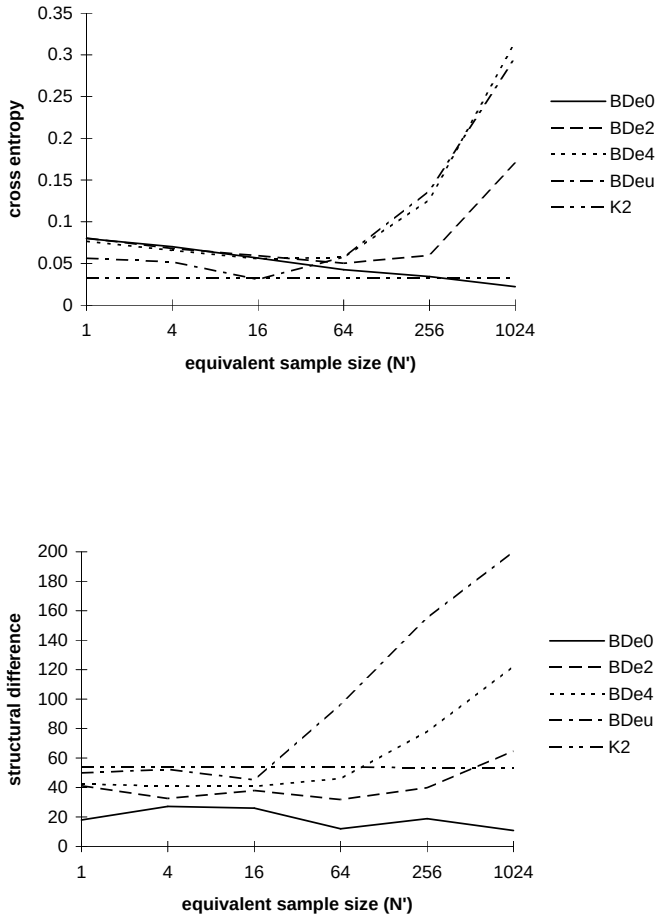

Figure 6: Cross entropy and structural difference of learned networks with respect to the Alarm network as a function the user's equivalent sample size N'. The metrics labeled BDe0, BDe2, and BDe4 correspond to the BDe metric with prior networks generated from the Alarm network with noise $\eta = 0, 2, 4$, respectively. Local search initialized with a maximum branching was applied to databases of size 10,000. Each data point represents an average over five learning instances. For all curves, the prior parameter κ was set to $1/(N' + 1)$. When computing the prior probabilities of network structure for K2, we used an empty prior graph.

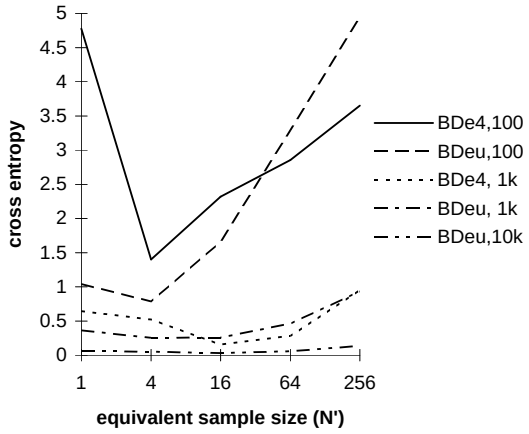

Figure 7: Cross entropy of learned networks with respect to the Alarm network as a function of the user's equivalent sample size for the BDe4 and BDeu metrics using databases of size 100, 1000, and 10000. The parameters of the experiment are the same as those in Figure 6.

other hand, if N' is too small, the metric is ignoring useful prior knowledge. Furthermore, as the deviation between the prior and gold-standard network structures increase, learning should be optimal for lower values of equivalent sample size. Results of this kind potentially can be used to calibrate users in the assessment of N'.

Quantitative results show that, for low values of N', all metrics perform about equally well, with K2 producing slightly lower cross entropies and the BDe metrics producing slightly lower structural differences. For large values of N', the BDe metrics did poorly, unless the gold-standard network was used as a prior network. These results suggest that the expert should pay close attention to the assessment of equivalent sample size when using the BDe metric. To provide a scale for cross entropy in the Alarm domain, note that the cross entropy of the Alarm network with an empty network for the domain whose marginal probabilities are determined from the Alarm network is 13.6.

The effect of database size is shown in Figure 7. As expected, the cross entropy of the learned networks with respect to the gold-standard network decreased as the database size increased.

Our metric comparisons revealed one surprising result. Namely, for databases ranging in size from 100 to 10000, we found learning performance to be better using an uninformative prior (BDeu) than that using the BDe metric with $\eta >\approx 4$. This result suggests that, unless the user's beliefs are close to the true model, we are better off ignoring those beliefs when establishing priors for a scoring metric.

In contrast, we found prior structural knowledge extremely useful for initializing search.

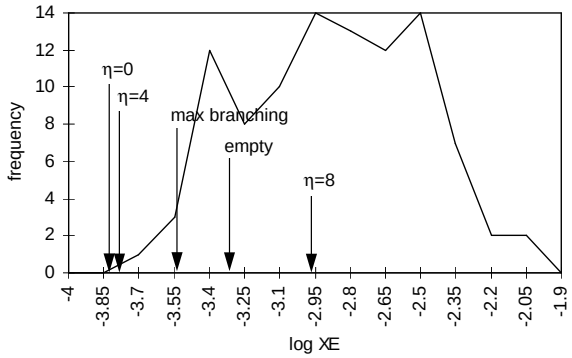

Figure 8: Cross entropy achieved by local search initialized with 100 random graphs, prior networks generated with different values of η, a maximum branching, and the empty graph. The BDeu metric with $N' = 16$ and $\kappa = 1/17$ was used in conjunction with a 10,000-case database.

To investigate the effects of search initialization on learning accuracy, we initialized local search with random structures, prior networks for different values of η, a maximum branching, and the empty graph. The results are shown in Figure 8. In this comparison, we used the BDeu metric with $N' = 16$, $\kappa = 1/17$, and a 10,000-case database. We created 100 random structures by picking orderings at random, and then, for a given ordering, placing in the structure each possible arc with probability $\kappa/(1 + \kappa)$. (This approach produced a distribution of random network structures that was consistent with the prior probability of network structures as determined by Equation 42).

The curve in Figure 8 is a histogram of the local maxima achieved with random-structure initialization. Prior networks for both $\eta = 0$ and $\eta = 4$ produced local maxima that fell at the extreme low end of this curve. Thus, even relatively inaccurate prior knowledge of structure helped the search algorithm to find good local maxima. Also, the maximum branching led to a local maximum with relatively low cross entropy, suggesting that these structures—which are produced in polynomial time—can be a good substitute for a prior network.

To investigate the effects of search algorithm on learning accuracy, we applied several search methods to 30 databases of size 10,000. For each search method, we used the BDeu metric with $N' = 16$ and $\kappa = 1/17$. The results are shown in Table 1. In The algorithm K2opt is CH's K2 search algorithm (described in Section 7.2) initialized with an ordering that is consistent with the Alarm network. The algorithm K2rev is the same algorithm initialized with the reversed ordering. We included the latter algorithm to gauge the sensi-

Table 1: Cross entropy, structural difference, and learning time (mean ± s.d.) for various search algorithms across 30 databases of size 10,000.

	cross entropy	structural difference	learning time
K2opt	0.026 ± 0.002	3.9 ± 1.4	1.9 min
K2rev	0.218 ± 0.009	139.3 ± 1.7	2.9 min
local	0.029 ± 0.005	45.0 ± 7.8	2.1 min
iterative local	0.026±0.003	42.0 ± 9.9	251 min
annealing	0.024 ± 0.007	19.5 ± 11.2	93 min

tivity of the K2 algorithm to variable order. Iterative local search used 30 restarts where, at each restart, the current network structure was modified with 100 random changes. (A single change was either an arc addition, deletion, or reversal.) The annealing algorithm used parameters $T_0 = 100, \alpha = 400, \beta = 200, \gamma = 0.95$, and $\delta = 120$. We found these parameters for iterative local search and annealing to yield reasonable learning accuracy after some experimentation. Local search, iterative local search, and annealing were initialized with a maximum branching.

K2opt obtained the lowest structural differences, whereas K2rev obtained the highest cross entropies and structural differences, illustrating the sensitivity of the K2 algorithm to variable ordering. All search algorithms—except K2rev—obtained low cross entropies. Local search performed about as well K2opt with respect to cross entropy, but not as well with respect to structural difference. The structural differences produced by annealing were significantly lower than those produced by local search. Nonetheless, annealing ran considerably slower than local search. Overall, local search performed well: It was both relatively accurate and fast, and did not require a variable ordering from the user.

Finally, to investigate the effects of using more than one network structure to represent a joint distribution, we applied the BDeu metric with $N' = 1$ and $\kappa = 1/2$ to a 25-case database. We searched for the network structures with the highest posterior probabilities using local search initialized with a maximum branching. The cross entropy of the l structures with the highest posterior probabilities with respect to the Alarm network as a function of l is shown in Figure 9. The cross entropy decreased until $l = 4$, at which point there was little additional improvement. For larger databases, the improvement was less. This result is not surprising, because given a large database, one network structure typically has a posterior probability far greater than the next most likely structure. Overall, however, we were surprised by the small amount of improvement.

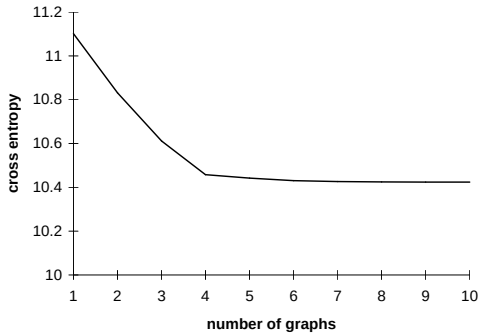

Figure 9: Cross entropy of learned networks with respect to the Alarm network as a function of the number of network structures used to represent the joint distribution. The BDeu metric with $N' = 1$ and $\kappa = 1/2$ was applied to a 25-case database. Local search initialized with a maximum branching was used. The network structures used to represent the joint were those with the highest posterior probabilities.

10 Summary

We have described a Bayesian approach for learning Bayesian networks from a combination of user knowledge and statistical data. We have described four contributions:

First, we developed a methodology for assessing informative priors on parameters for discrete networks. We developed our approach from the assumptions of parameter independence and parameter modularity made previously, as well as the assumption of *likelihood equivalence*, which says that data should not distinguish between network structures that represent the same assertions of conditional independence. This assumption is always appropriate when learning acausal Bayesian networks and is often appropriate when learning causal Bayesian networks. Rather surprising, we showed that likelihood equivalence, when combined with the parameter independence and other weak assumptions, implies that the parameters of the joint space must have a Dirichlet distribution. We showed, therefore, that the user may assess priors by constructing a single prior network and equivalent sample size for the domain. We noted that this assessment procedure is simple, but not sufficiently expressive for some domains. Consequently, we argued that the assumptions of parameter independence and likelihood equivalence are sometimes inappropriate, and sketched a possible approach for avoiding these assumptions.

Second, we combined informative priors from our construction to create the likelihood-equivalent BDe metric for complete databases. We note that our metrics and methods for constructing priors may be extended to nondiscrete domains [Geiger and Heckerman, 1994,

Heckerman and Geiger, 1994].

Third, we described search methods for identifying network structures with high posterior probabilities. We described polynomial algorithms for finding the highest-scoring network structures in the special case where every node has at most one parent. For the case where a node may have more than one parent, we reviewed heuristic search algorithms including local search, iterative local search, and simulated annealing.

Finally, we described a methodology for evaluating Bayesian-network learning algorithms. We applied this approach to a comparison of variants of the BDe metric and the K2 metric, and methods for search. We found that both the BDe and K2 metrics performed well. Rather surprising, we found that metrics using relatively uninformative priors performed as well as or better than metrics using more informative priors, unless that prior knowledge was extremely accurate. In contrast, we found that prior structural knowledge could be very useful for search, even when that knowledge deviates considerably from the true model. Also, we found that local search produced learned networks that were almost as good as those produced by iterative local search and annealing, but with far greater efficiency. Finally, we verified that learning multiple networks could improve predictive accuracy, although we were surprised by the small magnitude of improvement.

Acknowledgments

We thank Jack Breese, Wray Buntine, Greg Cooper, Eric Horvitz, and Steffen Lauritzen, for useful suggestions. Among other suggestions, Greg Cooper helped to develop the notion of an equivalent database. In addition, we thank Koos Rommelse for assisting with the implementation of the learning and evaluation algorithms.

References

[Aczel, 1966] Aczel, J. (1966). *Lectures on Functional Equations and Their Applications.* Academic Press, New York.

[Beinlich et al., 1989] Beinlich, I., Suermondt, H., Chavez, R., and Cooper, G. (1989). The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proceedings of the Second European Conference on Artificial Intelligence in Medicine,* London. Springer Verlag, Berlin.

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

[Camerini and Maffioli, 1980] Camerini, P. and Maffioli, L. F. F. (1980). The k best spanning arborescences of a network. *Networks*, 10:91–110.

[Chickering et al., 1995] Chickering, D., Geiger, D., and Heckerman, D. (1995). Learning Bayesian networks: Search methods and experimental results. In *Proceedings of Fifth Conference on Artificial Intelligence and Statistics,* Ft. Lauderdale, FL. Society for Artificial Intelligence in Statistics.

[Chow and Liu, 1968] Chow, C. and Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14:462–467.

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

[Cooper and Herskovits, 1991] Cooper, G. and Herskovits, E. (January, 1991). A Bayesian method for the induction of probabilistic networks from data. Technical Report SMI-91-1, Section on Medical Informatics, Stanford University.

[Dawid and Lauritzen, 1993] Dawid, A. and Lauritzen, S. (1993). Hyper Markov laws in the statistical analysis of decomposable graphical models. *Annals of Statistics*, 21:1272–1317.

[de Finetti, 1937] de Finetti, B. (1937). La prévision: See lois logiques, ses sources subjectives. *Annales de l'Institut Henri Poincaré*, 7:1–68. Translated in Kyburg and Smokler, 1964.

[Dempster et al., 1977] Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B 39:1–38.

[Druzdzel and Simon, 1993] Druzdzel, M. and Simon, H. (1993). Causality in Bayesian belief networks. In *Proceedings of Ninth Conference on Uncertainty in Artificial Intelligence,* Washington, DC, pages 3–11. Morgan Kaufmann.

[Edmonds, 1967] Edmonds, J. (1967). Optimum brachching. *J. Res. NBS*, 71B:233–240.

[Evans and Minieka, 1991] Evans, J. and Minieka, E. (1991). *Optimization algorithms for networks and graphs.* Marcel Dekker Inc., New York.

[Gabow, 1977] Gabow, H. (1977). Siam journal of computing. *Networks*, 6:139–150.

[Gabow et al., 1984] Gabow, H., Galil, Z., and Spencer, T. (1984). Efficient implementation of graph algorithms using contraction. In *Proceedings of FOCS.*

[Geiger and Heckerman, 1994] Geiger, D. and Heckerman, D. (1994). Learning Gaussian networks. In *Proceedings of Tenth Conference on Uncertainty in Artificial Intelligence,* Seattle, WA, pages 235–243. Morgan Kaufmann.

[Geiger and Heckerman, 1995] Geiger, D. and Heckerman, D. (Revised February, 1995). A characterization of the Dirichlet distribution applicable to learning Bayesian networks. Technical Report MSR-TR-94-16, Microsoft.

[Good, 1965] Good, I. (1965). *The Estimation of Probabilities.* MIT Press, Cambridge, MA.

[Heckerman, 1995] Heckerman, D. (February, 1995). Learning causal networks. Technical Report MSR-TR-95-04, Microsoft.

[Heckerman and Geiger, 1994] Heckerman, D. and Geiger, D. (December, 1994). Learning Bayesian networks. Technical Report MSR-TR-95-02, Microsoft.

[Heckerman et al., 1994] Heckerman, D., Geiger, D., and Chickering, D. (1994). Learning Bayesian networks: The combination of knowledge and statistical data. In *Proceedings of Tenth Conference on Uncertainty in Artificial Intelligence,* Seattle, WA, pages 293–301. Morgan Kaufmann.

[Heckerman and Nathwani, 1992] Heckerman, D. and Nathwani, B. (1992). An evaluation of the diagnostic accuracy of Pathfinder. *Computers and Biomedical Research*, 25:56–74.

[Heckerman and Shachter, 1994] Heckerman, D. and Shachter, R. (1994). A decision-based view of causality. In *Proceedings of Tenth Conference on Uncertainty in Artificial Intelligence,* Seattle, WA, pages 302–310. Morgan Kaufmann.

[Höffgen, 1993] Höffgen, K. (revised 1993). Learning and robust learning of product distributions. Technical Report 464, Fachbereich Informatik, Universität Dortmund.

[Horvitz, 1987] Horvitz, E. (1987). Reasoning about beliefs and actions under computational resource constraints. In *Proceedings of the Third Workshop on Uncertainty in Artificial Intelligence,* Seattle, WA. Association for Uncertainty in Artificial Intelligence, Mountain View, CA. Also in Kanal, L., Levitt, T., and Lemmer, J., editors, *Uncertainty in Artificial Intelligence 3,* pages 301–324. North-Holland, New York, 1989.

[Howard, 1988] Howard, R. (1988). Uncertainty about probability: A decision-analysis perspective. *Risk Analysis*, 8:91–98.

[Howard and Matheson, 1981] Howard, R. and Matheson, J. (1981). Influence diagrams. In Howard, R. and Matheson, J., editors, *Readings on the Principles and Applications of Decision Analysis*, volume II, pages 721–762. Strategic Decisions Group, Menlo Park, CA.

[Johnson, 1985] Johnson (1985). How fast is local search? In *FOCS*, pages 39–42.

[Karp, 1971] Karp, R. (1971). A simple derivation of Edmond's algrorithm for optimal branchings. *Networks*, 1:265–272.

[Korf, 1993] Korf, R. (1993). Linear-space best-first search. *Artificial Intelligence*, 62:41–78.

[Kullback and Leibler, 1951] Kullback, S. and Leibler, R. (1951). Information and sufficiency. *Ann. Math. Statistics*, 22:79–86.

[Lam and Bacchus, 1993] Lam, W. and Bacchus, F. (1993). Using causal information and local measures to learn Bayesian networks. In *Proceedings of Ninth Conference on Uncertainty in Artificial Intelligence,* Washington, DC, pages 243–250. Morgan Kaufmann.

[Lauritzen, 1982] Lauritzen, S. (1982). *Lectures on Contingency Tables.* University of Aalborg Press, Aalborg, Denmark.

[Madigan and Rafferty, 1994] Madigan, D. and Rafferty, A. (1994). Model selection and accounting for model uncertainty in graphical models using Occam's window. *Journal of the American Statistical Association*, 89.

[Matzkevich and Abramson, 1993] Matzkevich, I. and Abramson, B. (1993). Deriving a minimal I-map of a belief network relative to a target ordering of its nodes. In *Proceedings of Ninth Conference on Uncertainty in Artificial Intelligence,* Washington, DC, pages 159–165. Morgan Kaufmann.

[Metropolis et al., 1953] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953). *Journal of Chemical Physics*, 21:1087–1092.

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

[Pearl, 1995] Pearl, J. (1995). Causal diagrams for empirical research. *Biometrika*, to appear.

[Pearl and Verma, 1991] Pearl, J. and Verma, T. (1991). A theory of inferred causation. In Allen, J., Fikes, R., and Sandewall, E., editors, *Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, pages 441–452. Morgan Kaufmann, New York.

[Spiegelhalter et al., 1993] Spiegelhalter, D., Dawid, A., Lauritzen, S., and Cowell, R. (1993). Bayesian analysis in expert systems. *Statistical Science*, 8:219–282.

[Spiegelhalter and Lauritzen, 1990] Spiegelhalter, D. and Lauritzen, S. (1990). Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20:579–605.

[Spirtes et al., 1993] Spirtes, P., Glymour, C., and Scheines, R. (1993). *Causation, Prediction, and Search.* Springer-Verlag, New York.

[Suzuki, 1993] Suzuki, J. (1993). A construction of Bayesian networks from databases based on an mdl scheme. In *Proceedings of Ninth Conference on Uncertainty in Artificial Intelligence,* Washington, DC, pages 266–273. Morgan Kaufmann.

[Tarjan, 1977] Tarjan, R. (1977). Finding optimal branchings. *Networks*, 7:25–35.

[Titterington, 1976] Titterington, D. (1976). Updating a diagnostic system using unconfirmed cases. *Applied Statistics*, 25:238–247.

[Verma and Pearl, 1990] Verma, T. and Pearl, J. (1990). Equivalence and synthesis of causal models. In *Proceedings of Sixth Conference on Uncertainty in Artificial Intelligence,* Boston, MA, pages 220–227. Morgan Kaufmann.

[Winkler, 1967] Winkler, R. (1967). The assessment of prior distributions in Bayesian analysis. *American Statistical Association Journal*, 62:776–800.

[York, 1992] York, J. (1992). *Bayesian methods for the analysis of misclassified or incomplete multivariate discrete data.* PhD thesis, Department of Statistics, University of Washington, Seattle.

Appendix

In this section, we prove Theorem 1 and compute the Jacobian $J_{B_{sc}}$. To prove Theorem 1, we need the following definitions and preliminary results. In what follows we refer to Bayesian network structures simply as DAGs (directed acyclic graphs). An *edge* between x and y in a DAG refers to an adjacency between two nodes without regard to direction. A *v-structure* in a DAG is an ordered node triple (x, y, z) such that (1) the DAG contains the arcs $x \to y$ and $y \leftarrow z$, and (2) there is no edge between x and z.

Theorem 10 (Verma and Pearl, 1990, Theorem 1) *Two DAGs are equivalent if and only if they have identical edges and identical v-structures.*

Lemma 11 *Let D_1 and D_2 be equivalent DAGs. Let $R_{D_1}(D_2)$ be the subset of arcs in D_1 for which D_1 and D_2 differ in directionality. If $R_{D_1}(D_2)$ is not empty, then there exists an arc in $R_{D_1}(D_2)$ that can be reversed in D_1 such that the resulting graph remains equivalent to D_2. In particular, the following procedure finds such an edge. (Let $P_v = \{u | u \to v \in R_{D_1}(D_2)\}$.)*

1. *Perform a topological sort on the nodes in D_1*
2. *Let $y \in D_1$ be the minimal node with respect to the sort for which $P_y \neq \emptyset$*
3. *Let $x \in D_1$ be the maximal node with respect to the sort such that $x \in P_y$*

The arc $x \to y$ in D_1 is reversible.

Proof: Suppose $x \to y$ is not reversible. Then, by Theorem 10, reversing the arc either (1) creates a v-structure in the resulting graph that is not in D_2, (2) removes a v-structure from D_1 which is in D_2, (3) creates a cycle in the resulting graph.

Suppose reversing $x \to y$ creates a v-structure. Then there must exist an arc $w \to x$ in D_1 for which y and w are not adjacent. The arc $w \to x$ must be in $R_{D_1}(D_2)$, lest the v-structure (w, x, y) would exist in D_2 but not D_1. This fact, however, implies that x would have been chosen instead of y in step 2 above, because $w \in P_x$ and x comes before y in the topological sort.

The second case is impossible because a v-structure (x, y, v) cannot appear in both D_1 and D_2 if the arc between x and y have different orientations in D_1 and D_2.

Suppose reversing $x \to y$ creates a cycle in the resulting graph. This assumption implies that there is a directed path from x to y in D_1 that does not include the edge from x to y. Let $w \to y$ be the last arc in this path. Because $x \to y \in R_{D_1}(D_2)$, w and x must be adjacent in D_1, lest D_1 would contain a v-structure not in D_2. Because there is also a

directed path from x to w the edge between x and w must be oriented as $x \rightarrow w$, lest there would be a cycle in D_1. Because there are no directed cycles in D_2, however, either $x \rightarrow w$ or $w \rightarrow y$ must be in $R_{D_1}(D_2)$. If $x \rightarrow w$ is in $R_{D_1}(D_2)$, then w would have been chosen instead of y in step 2 above, because $x \in P_w$ and w precedes y in the sort. If $w \rightarrow y$ is in $R_{D_1}(D_2)$, then w would have been chosen instead of x in step 3 above, because $w \in P_y$ and w comes after x in the sort. □

Theorem 1 *Let D_1 and D_2 be two DAGs, and R_{D_1,D_2} be the set of edges by which D_1 and D_2 differ in directionality. Then, D_1 and D_2 are equivalent if and only if there exists a sequence of $|R_{D_1,D_2}|$ distinct arc reversals applied to D_1 with the following properties:*

1. *After each reversal, the resulting graph is a DAG and is equivalent to D_2*
2. *After all reversals, the resulting graph is identical to D_2*
3. *If $x \rightarrow y$ is the next arc to be reversed the current graph, then x and y have the same parents, with the exception that x is also a parent of y*

Proof: The if part of the theorem follows immediately from Theorem 10. Now, let $x \rightarrow y$ be an arc in D_1 that was chosen for reversal by the method described in Lemma 11. From the lemma, we know that after reversing this arc, D_1 will remain equivalent to D_2. Therefore, Condition 1 follows. Furthermore, each such edge reversal decreases by one the size of R_{D_1,D_2}. Thus, Condition 2 follows.

Suppose that Condition 3 does not hold. Then, there exists a node $w \neq x$ in D_1 that is either a parent of x or a parent of y, but not both. If $w \rightarrow x$ is in D_1, then w and y must be adjacent in D_2, lest (w, x, y) would be a v-structure in D_2 but not D_1. In D_1, however, w is not a parent of y; and thus there must be an arc from y to w, contradicting that D_1 is acyclic. If $w \rightarrow y$ is in D_1, a similar argument shows that reversing $x \rightarrow y$ in D_1 creates a cycle, which contradicts the result proven in Lemma 11. □

Theorem 12 *Let B_{sc} be any complete belief-network structure in domain U. The Jacobian for the transformation from Θ_U to $\Theta_{B_{sc}}$ is*

$$J_{B_{sc}} = \prod_{i=1}^{n-1} \prod_{x_1,\ldots,x_i} [\theta_{x_i|x_1,\ldots,x_{i-1}}]^{[\prod_{j=i+1}^{n} r_j]-1} \tag{47}$$

Proof. We proceed by induction using J_n to denote $J_{B_{sc}}$ for the n-variable case. When $n = 1$, the theorem holds trivially, because $\Theta_U = \Theta_{B_{sc}}$ and $J_1 = 1$. For the induction step, let us assume that the complete belief-network structure has variable ordering $x_1, \ldots, x_{n+1}$. First, we change variables from $\Theta_{x_1,\ldots,x_{n+1}}$ to $\Theta_{x_{n+1}|x_1,\ldots,x_n} \cup \Theta_{x_1,\ldots,x_n}$.

By definition, $\theta_{x_1,\ldots,x_{n+1}} = \theta_{x_1,\ldots,x_n} \cdot \theta_{x_{n+1}|x_1,\ldots,x_n}$. Thus, the Jacobian matrix consists of $\{\prod_{i=1}^{n} r_i\}$ block matrices of size r_{n+1} on the main diagonal, one for each instance of the first n variables, of the form

$$\begin{pmatrix} \theta_{x_{n+1}=1|x_1,\ldots,x_n} & \theta_{x_1,\ldots x_n} & 0 & \ldots & 0 \\ \theta_{x_{n+1}=2|x_1,\ldots,x_n} & 0 & \theta_{x_1,\ldots x_n} & \ldots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ \theta_{x_{n+1}=r_{n+1}-1|x_1,\ldots,x_n} & 0 & 0 & \ldots & \theta_{x_1,\ldots x_n} \\ 1-\sum_{i=1}^{r_{n+1}-1} \theta_{x_{n+1}=r_i|x_1,\ldots,x_n} & -\theta_{x_1,\ldots x_n} & -\theta_{x_1,\ldots x_n} & \ldots & -\theta_{x_1,\ldots x_n} \end{pmatrix}$$

The determinant of this matrix is the Jacobian given below:

$$J_{n+1}^{c} = \prod_{x_1,\ldots,x_n} [\theta_{x_1,\ldots,x_n}]^{r_{n+1}-1} \tag{48}$$

Next, we change variables from $\Theta_{x_1,\ldots,x_n}$ to $\Theta_{x_1} \cup \Theta_{x_2|x_1} \cup \ldots \cup \Theta_{x_n|x_1,\ldots,x_{n-1}}$, with the Jacobian J_n obtained from the induction hypothesis. The combined Jacobian for the transformation from $\Theta_{x_1,\ldots,x_{n+1}}$ to $\Theta_{x_1} \cup \ldots \cup \Theta_{x_{n+1}|x_1,\ldots,x_n}$ is $J_{n+1} = J_{n+1}^{c} \cdot J_n$. Consequently, we have

$$J_{n+1} = \prod_{x_1,\ldots,x_n} [\prod_{i=1}^{n} \theta_{x_i|x_1,\ldots,x_{i-1}}]^{r_{n+1}-1} \cdot \prod_{i=1}^{n-1} \prod_{x_1,\ldots,x_i} [\theta_{x_i|x_1,\ldots,x_{i-1}}]^{[\prod_{j=i+1}^{n} r_j]-1} \tag{49}$$

Collecting terms $\theta_{x_i|x_1,\ldots,x_i-1}$, we can rewrite Equation 49 as

$$J_{n+1} = \prod_{i=1}^{n} \prod_{x_1,\ldots,x_i} [\theta_{x_i|x_1,\ldots,x_{i-1}}]^{N_i'-1} \tag{50}$$

where

$$N_i' - 1 = \left[(r_{n+1}-1) \prod_{j=i+1}^{n} r_j \right] + \left[\prod_{j=i+1}^{n} r_j \right] - 1 = \left[\prod_{j=i+1}^{n+1} r_j \right] - 1 \tag{51}$$

Substituting Equation 51 into Equation 50 completes the induction. □

Notation

$x, y, z, \ldots$	Variables or their corresponding nodes in a Bayesian network
$X, Y, Z, \ldots$	Sets of variables or corresponding sets of nodes
$x = k$	Variable x is in state k
$X = \vec{k}$	The set of variables X takes on instance $\vec{k}$
$X \setminus Y$	The variables in X that are not in Y
U	A domain: a set of variables $\{x_1, \ldots, x_n\}$
C	A case: an instance of some or all of the variables U
D	A database: a set of cases $\{C_1, \ldots, C_m\}$
D_l	The first $l-1$ cases in D
$p(X = \vec{k}_X \mid Y = \vec{k}_Y, \xi)$	The probability that $X = \vec{k}_X$ given $Y = \vec{k}_Y$ for a person with current state of information ξ
$p(X \mid Y, \xi)$	The set of probability distributions for X given all instances of Y
B_s	A Bayesian network structure (a directed acyclic graph)
B_p	The probability set associated with B_s
Π_i	The parents of x_i in a Bayesian network structure
r_i	The number of states of variable x_i
q_i	The number of instances of Π_i
B_{sc}	A complete network structure
B_s^h	The hypothesis corresponding to network structure B_s
$\theta_{X=\vec{k}_X \mid Y=\vec{k}_Y}$	The multinomial parameter corresponding to $p(X = \vec{k}_X \mid Y = \vec{k}_Y, \xi)$ ($\vec{k}_X$ and $\vec{k}_Y$ are often implicit)
$\Theta_{X \mid Y=\vec{k}_Y}$	The multinomial parameters corresponding to the probability distribution $p(X \mid Y = \vec{k}_Y, \xi)$
$\Theta_{X \mid Y}$	The multinomial parameters $\Theta_{X \mid Y=\vec{k}_Y}$ for all instances of $\vec{k}_Y$
Θ_U	The multinomial parameters corresponding to the joint probability distribution $p(U \mid \xi)$
θ_{ijk}	$= \theta_{x_i=k \mid \Pi_i=j}$
Θ_{ij}	$= \cup_{k=1}^{r_i} \{\theta_{ijk}\}$
Θ_i	$= \cup_{j=1}^{q_i} \{\Theta_{ij}\}$
Θ_{Bs}	$= \cup_{i=1}^{n} \Theta_i$
$\rho(\Theta \mid \xi)$	The probability density of Θ given ξ
N'	An equivalent sample size
N'_{ijk}	The Dirichlet exponent of θ_{ijk} (see Assumption 4)
N'_{ij}	$= \sum_{k=1}^{r_i} N'_{ijk}$
N_{ijk}	The number of cases in a database where $x_i = k$ and $\Pi_i = j$
N_{ij}	$= \sum_{k=1}^{r_i} N_{ijk}$