Learning Bayesian Networks: The Combination of Knowledge and Statistical Data

David Heckerman **Dan Geiger*** **David M. Chickering**
Microsoft Research, Bldg 9S
Redmond, WA 98052-6399
heckerma@microsoft.com, dang@cs.technion.ac.il, dmax@cs.ucla.edu

Abstract

We describe algorithms for learning Bayesian networks from a combination of user knowledge and statistical data. The algorithms have two components: a scoring metric and a search procedure. The scoring metric takes a network structure, statistical data, and a user's prior knowledge, and returns a score proportional to the posterior probability of the network structure given the data. The search procedure generates networks for evaluation by the scoring metric. Our contributions are threefold. First, we identify two important properties of metrics, which we call *score equivalence* and *parameter modularity.* These properties have been mostly ignored, but when combined, greatly simplify the encoding of a user's prior knowledge. In particular, a user can express his knowledge—for the most part—as a single *prior Bayesian network* for the domain. Second, we describe greedy hill-climbing and annealing search algorithms to be used in conjunction with scoring metrics. In the special case where each node has at most one parent, we show that heuristic search can be replaced with a polynomial algorithm to identify the networks with the highest score. Third, we describe a methodology for evaluating Bayesian-network learning algorithms. We apply this approach to a comparison of our metrics and search procedures.

1 Introduction

The fields of Artificial Intelligence and Statistics share a common goal of modeling real-world phenomena. Whereas AI researchers have emphasized a knowledge-based approach to achieving this goal, statisticians have traditionally emphasized a data-based approach. In this paper, we present a unification of these two approaches. In particular, we develop algorithms based on Bayesian principles that take as input (1) a user's prior knowledge expressed—for the most part—as a *prior Bayesian network* and (2) statistical data, and returns an improved Bayesian network.

Several researchers have examined methods for learning Bayesian networks from data, including Cooper and Herskovits (1991) and Cooper and Herskovits (1992) (herein referred to as CH), Buntine (1991) (herein referred to as Buntine), and Spiegelhalter et al. (1993) (herein referred to as SDLC). (Each of these references contain an excellent review of additional related approaches.) These methods all have the same basic components: a scoring metric and a search procedure. The metric computes a score that is proportional to the posterior probability of a network structure, given data and a user's prior knowledge. The search procedure generates networks for evaluation by the scoring metric. These methods use these two components to identify a network or set of networks with high posterior probabilities, and these networks are then used to predict future events.

In this paper, we concentrate on identifying a single Bayesian network with a high posterior probability. Our methods are generalized easily to multiple networks using techniques described in CH and in Madigan and Raferty (1994). In Section 2, we develop scoring metrics. Although we restrict ourselves to domains containing only discrete variables, as we show in Geiger and Heckerman (1994), our metrics can be generalized to domains containing both discrete and continuous variables. A major contribution of this paper is that we develop our metrics from a set of consistent properties and assumptions. Two of these, called parameter modularity and score equivalence, have been ignored for the most part, and their combined ramifications have not been explored. The assumption of *parameter modularity,* which has been made implicitly by CH, Buntine, and SDLC, addresses the relationship among prior distributions of parameters for different Bayesian-network structures. The property of *score equivalence* says that two Bayesian-network structures that represent the same set of independence and de-

*Author's primary affiliation: Computer Science Department, Technion, Haifa 32000, Israel.

pendence assertions should receive the same score. We provide justifications for these assumptions, and show that when combined with other reasonable assumptions about learning Bayesian networks, these assumptions provide a straightforward method for combining user knowledge and statistical data that makes use of a prior network. Our approach is to be contrasted with those of CH and Buntine who do not make use of a prior network, and to those of CH and SDLC who do not satisfy the property of score equivalence.

Our identification of the principle of score equivalence arises from a subtle distinction between two types of Bayesian networks. The first type, called *belief networks*, represents only assertions of independence and dependence. The second type, called *causal networks*, represents assertions of cause and effect as well as assertions of independence and dependence. In this paper, we argue that metrics for belief networks should satisfy score equivalence, whereas metrics for causal networks need not.

Our score-equivalent metric for belief networks is similar to metrics described by Dawid and Lauritzen (1993) and Madigan and Raferty (1994), except that our metric scores directed networks, whereas their metrics score undirected networks. In this paper, we concentrate on directed models rather than on undirected models, because we believe that users find the former easier to build and interpret.

In Section 3, we examine methods for finding networks with high scores. 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 a greedy hill-climbing algorithm and an annealing algorithm for the general case.

Finally, in Section 4, we describe a methodology for evaluating learning algorithms. We apply this methodology to a comparison of our metrics and search methods.

2 Scoring Metrics

2.1 Belief Networks and Notation

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 $x_i = k$ when we observe 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 $x_i = k_i, x_i \in X$. We use $p(x = \vec{k}_X | y = \vec{k}_Y, \xi)$ to denote the probability of a person with background knowledge ξ for the observation $x = \vec{k}_X$, given the observation $y = \vec{k}_Y$. 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 space* of U is the set of all instances of U. The *joint probability distribution* over U is the probability distribution over the joint space of U.

A belief network represents a joint probability distribution over U by encoding assertions of conditional independence and dependence as well as probability distributions for variables. From the chain rule of probability, we know

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

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

$$\begin{gathered} p(x_i|x_1, \ldots, x_{i-1}, \xi) = p(x_i|\Pi_i, \xi) \\ \forall\, Q \subset \Pi_i : \; p(x_i|x_1, \ldots, x_{i-1}, \xi) \neq p(x_i|Q, \xi) \end{gathered} \qquad (2)$$

A belief network is a pair (B_S, B_P), where B_S is a belief-network structure that encodes the assertions of conditional independence and dependence in Equations 2, and B_P is a set of probability distributions corresponding to that structure. In particular, 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 the remainder of this paper, we use x_i to refer to both the variable and its corresponding node in a graph, unless otherwise stated.) Associated with node x_i in B_S are the probability distributions $p(x_i|\Pi_i, \xi)$. B_P is the union of these distributions. Combining Equations 1 and 2, we see that any belief network for U uniquely determines a joint probability distribution for U. That is,

$$p(x_1, \ldots, x_n|\xi) = \prod_{i=1}^{n} p(x_i|\Pi_i, \xi) \qquad (3)$$

2.2 Metrics for Belief Networks

We are interested in computing a score for a belief-network structure, given a sequence of instances of U. We call a single instance of some or all of the variables in U a *case*. We call a sequence of cases $C_1, \ldots, C_m$ a *database*. If all variables in a case are observed, we say that the case is *complete*. If all cases in a database are complete, we say that the database is *complete*.

Our scoring metrics are based on six assumptions, the first of which is the following:

Assumption 1 *All variables in U are discrete.*

Our next assumption involves the concept of exchangeability. We say that a database is *exchangeable* if any database obtained by a permutation of case numbers of the original database has the same probability as the original database. Essentially, the assumption that a database is exchangeable is an assertion that the processes generating the data do not change in time.

Assumption 2 *All complete databases for U are exchangeable.*

Under Assumptions 1 and 2, De Finetti (1937) showed that any complete database has a multinomial distribution. That is, the probability of any complete database may be computed *as if* it were a multinomial sample from the joint space of U. We use $\phi_{\vec{k}}$ to denote the multinomial parameter for the event $U = \vec{k}$, and $\Phi = \cup_{\vec{k}} \phi_{\vec{k}}$ to denote the collection of all parameters. We can think of $\phi_{\vec{k}}$ as the long-run fraction of cases where $U = \vec{k}$ was observed. Howard presents an alternative interpretation [Howard, 1988].

We shall find it convenient to define parameters for subsets and conditional subsets of variables. In particular, for any two disjoint sets of variables $X, Y \subseteq U$, we use $\theta_{X=\vec{k_X}|Y=\vec{k_Y}}$ to denote the long-run fraction of cases where $X = \vec{k_X}$ among those cases in which $Y = \vec{k_Y}$. When Y is empty, we omit the conditioning event in the notation. Furthermore, we use $\theta_{X|Y}$ to denote the collection of parameters $\theta_{X=\vec{k_X}|Y=\vec{k_Y}}$ for all instances of X and Y. Thus, for example, $\Theta_{x_1,\ldots,x_n} = \Phi$. In addition, we omit state assignments from our notation, when the meaning of a term is clear from context. For example, when we write $\theta_{x_2|x_1} \neq \theta_{x_2}$, we mean that the inequality holds for a least one state of x_1 and one state of x_2.

A Bayesian measure of the goodness of a network structure is its posterior probability given a database:

$$p(B_S|D,\xi) = c\, p(B_S|\xi)\, p(D|B_S,\xi)$$

where $c = 1/p(D|\xi) = 1/\sum_{B_S} p(B_S|\xi)\, p(D|B_S,\xi)$ is a normalization constant. For even small domains, however, there are too many network structures to sum over in order to determine the constant. Therefore we use $p(B_S|\xi)\, p(D|B_S,\xi) = p(D, B_S|\xi)$ as our score.

More problematic is our use of the term B_S as an argument of a probability. In particular, B_S is a belief-network structure, not an event. Thus, we need a definition of an event B_S^e that corresponds to structure B_S (the superscript "e" stands for event). We propose the following definition.

Definition *The event B_S^e corresponding to a belief-network structure B_S holds true iff*

$$\begin{gathered} \theta_{x_i|x_1,\ldots,x_{i-1}} = \theta_{x_i|\Pi_i} \\ \forall\, Q \subset \Pi_i :\ \theta_{x_i|x_1,\ldots,x_{i-1}} \neq \theta_{x_i|Q} \end{gathered} \tag{4}$$

In words, we say that the event B_S^e holds true if and only if B_S is a belief-network structure for the multinomial parameters (i.e., long-run fractions) for U—that is, if and only if these parameters satisfy the independence and dependence assertions of B_S. For example, the event B_S^e for the belief network $x_1 \rightarrow x_2 \rightarrow x_3$ corresponds to the assertions

$$\theta_{x_3|x_1,x_2} = \theta_{x_3|x_2} \quad \theta_{x_2|x_1} \neq \theta_{x_2} \quad \theta_{x_3|x_2} \neq \theta_{x_3}$$

The definition seems reasonable enough. In particular, as learning network structure involves the repeated observation of cases, we are interested in learning the conditional independencies and dependencies that apply to the long-run fractions of U. CH, Buntine, and SDLC apparently all use this definition implicitly. To our knowledge, however, we are the first researchers to make this assumption explicit.

We can now define belief-network metrics.

Definition *A belief-network metric produces the score $p(D, B_S^e|\xi)$, the user's posterior probability of database D and the event B_S^e, given background knowledge ξ.*

Our definition of the event B_S^e places an important restriction on the scores produced by a belief-network metric. When two belief-network structures represent the same assertions of conditional independence and dependence, we say that they are *isomorphic*. For example, consider the domain consisting of only variables x and y. If we reverse the arc in the belief network for this domain where x points to y, we obtain a network that represents the same assertion as the original network: x and y are dependent. Given the definition of B_S^e, it follows that the events B_{S1}^e and B_{S2}^e are equivalent if and only if the structures B_{S1} and B_{S2} are isomorphic. That is, the relation of isomorphism induces an equivalence class on the set of events B_S^e. We call this property *event equivalence*.

Proposition 1 (Event Equivalence)
Belief-network structures B_{S1} and B_{S2} are isomorphic if and only if $B_{S1}^e = B_{S2}^e$.

As a consequence, if a belief-network metric is to be consistent with the rules of probability, then it must satisfy the property of *score equivalence*.

Proposition 2 (Score Equivalence) *The scores of two isomorphic belief-network structures must be equal.*

Technically then, we should score each belief-network-structure equivalence class, rather than each belief-network structure. Nonetheless, users find it intuitive to work with (i.e., construct and interpret) belief networks. Consequently, we continue our presentation in terms of belief networks, keeping Proposition 2 in mind.

Given, a belief-network structure B_S, we need the following notation.[1] Let r_i be the number of states of variable x_i. 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 these instances. 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

$$\theta_{ijk} \equiv \theta_{x_i=k|\Pi_i=j}$$

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

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

$$\Theta_{B_S} \equiv \cup_{i=1}^{n}{}_{j=1}^{q_i}\Theta_{ij}$$

The set Θ_{B_S} corresponds to the parameter set B_P for belief-network structure B_S, as defined in Section 2.1. Here, however, these parameters are long-run fractions, not (subjective) probabilities.

The following assumptions—also made by CH, Buntine, and SDLC—allow us to derive simple closed-form formulas for metrics.

Assumption 3 *All databases are complete.*

Spiegelhalter et al. (1993) provide an excellent survey of approximations that circumvent this assumption.

Assumption 4 (Parameter Independence) *For all belief-network structures B_S,*

$$\rho(\Theta_{B_S}|B_S^e,\xi) = \prod_i \prod_j \rho(\Theta_{ij}|B_S^e,\xi)$$

This assumption, called *parameter independence*, says that the long-run conditional fractions associated with a given belief-network structure are independent, except for the obvious dependence among the parameters for a given variable (which must sum to one). Given Assumption 3, it follows that the parameters remain independent when cases are observed.

A general metric now follows. Applying the chain rule, we obtain

$$p(D|B_S,\xi) = \prod_{l=1}^{m} p(C_l|C_1,\ldots,C_{l-1},B_S^e,\xi) \quad (5)$$

where C_i is the ith case in the database. Conditioning on the parameters of the belief-network structure B_S, and using the fact that parameters remain independent, given cases, we have

$$p(C_l|C_1,\ldots,C_{l-1},B_S^e,\xi) = \int_{\Theta_{B_S}} \Big\{p(C_l|\Theta_{B_S},B_S^e,\xi) \cdot \textstyle\prod_i \prod_j \rho(\Theta_{ij}|C_1,\ldots,C_{l-1},B_S^e,\xi)\Big\} \quad (6)$$

Because each case in D is complete, we have

$$p(C_l|\Theta_{B_S},B_S^e,\xi) = \prod_i \prod_j \prod_k \theta_{ijk}{}^{\alpha_{lijk}} \quad (7)$$

where α_{lijk} is 1 if and only if $x_i = k$ and $\Pi_i = j$ in case C_l, and 0 otherwise. Plugging Equation 7 into Equation 6 and the result into Equation 5 yields

$$p(D,B_S^e|\xi) = p(B_S^e|\xi) \cdot \prod_i \prod_j \prod_k \prod_l < \theta_{ijk}|C_1,\ldots,C_{l-1},B_S^e,\xi >^{\alpha_{lijk}} \quad (8)$$

where <> denotes expectation with respect to Θ_{ij}.

One difficulty in applying Equation 8 is that, in general, a user must provide prior distributions for every parameter set Θ_{ij} associated with every structure B_S. To reduce the number of prior distributions, we make the following assumption.

Assumption 5 (Parameter Modularity) *If x_i has the same parents in any two belief-network structures B_{S1} and B_{S2}, then for $j = 1,\ldots,q_i$,*

$$\rho(\Theta_{ij}|B_{S1}^e,\xi) = \rho(\Theta_{ij}|B_{S2}^e,\xi)$$

We call this property *parameter modularity,* because it says that the densities for parameters Θ_{ij} depend only on the structure of the belief network that is local to variable x_i—namely, Θ_{ij} only depends on the parents of x_i. For example, consider two belief networks for binary nodes x and y. Let B_{S1} be the network with an arc pointing from x to y, and B_{S2} be the network with no arc between x and y. Then $\rho(\theta_x|B_{S1}^e,\xi) = \rho(\theta_x|B_{S2}^e,\xi)$ because x has the same parents (namely, none) in both belief networks. In contrast, the assumption does not hold for the parameters for y. That is, in B_{S1}, we can have $\rho(\theta_{y|x}|B_{S1}^e,\xi) \neq \rho(\theta_{y|\bar{x}}|B_{S1}^e,\xi)$. In B_{S2}, however, x and y are independent. Consequently, $\rho(\theta_{y|x}|B_{S2},\xi) = \rho(\theta_{y|\bar{x}}|B_{S2},\xi)$. Thus, either $\rho(\theta_{y|x}|B_{S1}^e,\xi) \neq \rho(\theta_{y|x}|B_{S2}^e,\xi)$ or $\rho(\theta_{y|\bar{x}}|B_{S1}^e,\xi) \neq \rho(\theta_{y|\bar{x}}|B_{S2}^e,\xi)$ The failure of this assumption for the parameters of y is consistent with the fact that y has different parents in B_{S1} and B_{S2}.

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). Also, in the context of causal networks, the assumption has a compelling justification (see Section 2.5).

Given Assumption 3, parameter modularity holds, even when previous cases have been observed. Consequently, we can rewrite Equation 8 as

$$p(D,B_S^e|\xi) = p(B_S^e|\xi) \cdot \prod_i \prod_j \prod_k \prod_l < \theta_{ijk}|C_1,\ldots,C_{l-1},\xi >^{\alpha_{lijk}} \quad (9)$$

We call Equation 9 the Bd metric, which stands for *B*ayesian metric for *d*iscrete variables.

In making the assumptions of parameter independence and parameter modularity, we have—in effect—specified the prior densities for the multinomial parameters in terms of the structure of a belief network. Consequently, there is the possibility that this specification violates the property of score equivalence. In Heckerman et al. (1994), however, we show that our assumptions and score equivalence are consistent. In particular, we show that the conditional densities $\rho(\theta_{ijk}|C_1,\ldots,C_{l-1},\xi)$ constructed from our assumptions and the property of event equivalence always guarantee score equivalence.

In Heckerman et al. (1994), we provide greater detail about this general metric. Here, we concentrate on a special case where each parameter set Θ_{ij} has a Dirichlet distribution.

A *complete belief-network* is one with no missing edges—that is, one containing no conditional independence assertions of the form given in Equation 4. From our definition of the event B_S^e, we know that the event associated with any complete belief-network structure is the same; and we use $B_{S_C}^e$ to denote this event.

Assumption 6 *For every complete belief-network structure B_{S_C}, and for all $\Theta_{ij} \subseteq \Theta_{B_{S_C}}$, $\rho(\Theta_{ij}|B_{S_C}^e,\xi)$ has a Dirichlet distribution. Namely, there exists exponents N'_{ijk} such that*

$$\rho(\Theta_{ij}|B_{S_C}^e,\xi) = c \cdot \prod_k {\theta_{ijk}}^{N'_{ijk}}$$

where c is a normalization constant.

From this assumption and our assumption of parameter modularity, it follows that for *every* belief-network structure B_S, and for all $\Theta_{ij} \subseteq \Theta_{B_S}$, $\rho(\Theta_{ij}|B_S^e,\xi)$ has a Dirichlet distribution. When every such parameter set of B_S has this distribution, we simply say that $\rho(\Theta_{B_S}|B_S^e,\xi)$ is Dirichlet.

Combining our previous assumptions with this consequence of Assumption 6, we obtain

$$\rho(\Theta_{ij}|D,B_S^e,\xi) = c \cdot \prod_k {\theta_{ijk}}^{N'_{ijk}+N_{ijk}} \tag{10}$$

where N_{ijk} is the number of cases in D where $x_i = k$ and $\Pi_i = j$, and c is some other normalization constant. Thus, if the prior distribution for Θ_{ij} has a Dirichlet distribution, then so does the posterior distribution for Θ_{ij}. 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. Given this family,

$$< \theta_{ijk}|D,\xi >= \frac{N'_{ijk} + N_{ijk} + 1}{N'_{ij} + N_{ij} + r_i} \tag{11}$$

where $N_{ijk} = \sum_{k=1}^{r_i} N_{ij}$, and $N_{ijk} = \sum_{k=1}^{r_i} N_{ij}$. Substituting Equation 11 into each term of Equation 9, and performing the sum over l, we obtain

$$p(D,B_S^e|\xi) = p(B_S^e|\xi) \cdot \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{\Gamma(N'_{ij}+r_i)}{\Gamma(N'_{ij}+N_{ij}+r_i)} \cdot \prod_{k=1}^{r_i} \frac{\Gamma(N'_{ijk}+N_{ijk}+1)}{\Gamma(N'_{ijk}+1)} \tag{12}$$

where Γ is the *Gamma* function, which satisfies $\Gamma(x+1) = x\Gamma(x)$. We call Equation 12 the BD metric ("D" stands for *D*irichlet).

In the following section, we show that the property of event equivalence imposes the following constraints on the exponents N'_{ijk}:

$$N'_{ijk} + 1 = K \cdot p(x_i = k, \Pi_i = j|B_{S_C}^e,\xi) \tag{13}$$

where K is a constant to be described, and B_{S_C} is the event associated with any complete belief-network structure. Substituting this restriction into Equation 12, we obtain the following metric for belief networks.

Theorem 1 *Given Assumptions 1 through 6,*

$$p(D,B_S^e|\xi) = p(B_S^e|\xi) \cdot \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{\Gamma(K p(\Pi_i=j|B_{S_C}^e,\xi))}{\Gamma(N_{ij}+K p(\Pi_i=j|B_{S_C}^e,\xi))} \cdot \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk}+K p(x_i=k,\Pi_i=j|B_{S_C}^e,\xi))}{\Gamma(K p(x_i=k,\Pi_i=j|B_{S_C}^e,\xi))} \tag{14}$$

where B_{S_C} is any complete belief-network structure.

We call Equation 14 the BDe metric ("e" stands for score equivalence). We show that this metric satisfies score equivalence in Heckerman et al. (1994). We note that Buntine presented without derivation the special case of Equation 14 obtained by letting $p(U|B_{S_C}^e,\xi)$ be uniform, and noted the property of score equivalence. Also, Equation 12 is the general metric developed by Cooper and Herskovits (1992) with the exception that they require the exponents N'_{ijk} to be integers. CH also present a special case of BD wherein each N'_{ijk} is set to zero, yielding a uniform Dirichlet distribution on each density $\rho(\Theta_{ij}|B_S^e,\xi)$. This special case does not exhibit the property of score equivalence.

2.3 The Combination of Knowledge and Statistical Data: The Prior Belief Network

In this section, we show how the property of event equivalence and our assumptions lead to a straightforward method for generating the exponents N'_{ijk} from a user's prior knowledge. We thereby provide a methodology for the combination of user knowledge and statistical data.

In Heckerman et al. (1994) (Theorem 7), we show that if $\rho(\Theta_{B_{S_C}}|\xi)$ is Dirichlet for every complete belief-network structure B_{S_C}, then the density of the parameters for the joint space—$\rho(\Phi|\xi)$—also has a Dirichlet distribution. In the previous section, we assumed that $\rho(\Theta_{B_{S_C}}|B_{S_C}^e,\xi)$ is Dirichlet for every complete belief-network structure B_{S_C}. Furthermore, by the definition of the event B_S^e, we know that events B_{SC1}^e and B_{SC2}^e for two complete belief-network structures B_{SC1} and B_{SC2}, are identical. Consequently, we can apply this theorem to conclude that $\rho(\Phi|B_{S_C}^e,\xi)$ has a Dirichlet distribution. That is,

$$\rho(\Phi|B_{S_C}^e,\xi) = \rho(\Theta_{x_1,\ldots,x_n}|B_{S_C}^e,\xi) = c \cdot \prod_{x_1,\ldots,x_n} [\theta_{x_1,\ldots,x_n}]^{e_{x_1,\ldots,x_n}} \tag{15}$$

Also in Heckerman et al. (1994) (Theorem 5), we show that the converse is true. Namely, that if Equation 15 holds, then the density $\rho(\Theta_{B_{S_C}}|B_{S_C}^e,\xi)$ is Dirichlet for every complete network structure B_{S_C}. Furthermore, we show that

$$e_{x_i|x_1,\ldots,x_{i-1}} + 1 = \sum_{x_{i+1},\ldots,x_n} (e_{x_1,\ldots,x_n} + 1) \tag{16}$$

where $e_{x_i|x_1,\ldots,x_{i-1}}$ is the exponent of $\theta_{x_i|x_1,\ldots,x_{i-1}}$ in the Dirichlet distributions for $\rho(\Theta_{B_{S_C}}|B_{S_C}^e,\xi)$. The term N'_{ijk}, however, is just $e_{x_i=k|\Pi_i=j}$. Thus, using a complete belief-network structure wherein Π_i are the parents of x_i, we obtain the constraint

$$N'_{ijk} + 1 = \sum_{\{e_{x_1,\ldots,x_n} | x_i=k, \Pi_i=j\}} (e_{x_1,\ldots,x_n} + 1) \quad (17)$$

where the sum ranges over all instances of U that are consistent with $x_i = k$ and $\Pi_i = j$. Therefore, if we can assess the exponents in the Dirichlet distribution for Φ, then we obtain all terms N'_{ijk}.

Winkler (1967) describes several methods for assessing a Beta distribution, which is the Dirichlet distribution for a binary variable. 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.

We find the last method to be particularly well suited to the assessment of the exponents $e_{x_1,\ldots,x_n}$. The method in based on the fact that the mean of $\theta_{x_1,\ldots,x_n}$ with respect to the density $\rho(\theta_{x_1,\ldots,x_n} | B_{S_C}^e, \xi)$ is equal to the user's prior probability for $x_1, \ldots, x_n$. Consequently, we have

$$\frac{e_{x_1,\ldots,x_n} + 1}{\sum_{x_1,\ldots,x_n} (e_{x_1,\ldots,x_n} + 1)} = p(x_1, \ldots, x_n | B_{S_C}^e, \xi)$$

from which we obtain

$$e_{x_1,\ldots,x_n} + 1 = K \; p(x_1, \ldots, x_n | B_{S_C}^e, \xi) \quad (18)$$

where

$$K = \sum_{x_1,\ldots,x_n} (e_{x_1,\ldots,x_n} + 1) \quad (19)$$

Thus, a user can assess the exponents $e_{x_1,\ldots,x_n}$ by assessing the joint probability distribution for U and the constant K. From Equations 17 and 18, we obtain the constraint given in the previous section (Equation 13):

$$N'_{ijk} + 1 = K \; p(x_i = k, \Pi_i = j | B_{S_C}^e, \xi)$$

A user can assess the joint probability distribution $p(x_1, \ldots, x_n | B_{S_C}^e, \xi)$ by constructing a belief network for U, given $B_{S_C}^e$. We call this network the user's *prior belief network*. At first glance, there seems to be a contradiction in asking the user to construct such a belief network—which may contain assertions of independence—under the assertion that $B_{S_C}^e$ is true. The assertions of independence in the prior network, however, refer to independencies in the next case. In contrast, the assertion of full dependence $B_{S_C}^e$ refers to long-run fractions. For example, consider the domain containing only binary variables x and y. By definition, if $B_{S_C}^e$ is true, then $\theta_{x|y} \neq \theta_{x|\neg y}$. Nonetheless, it may be the case that

$$< \theta_{x|y} | B_S^e, \xi > = \int \theta_{x|y} \; \rho(\theta_{x|y} | B_S^e, \xi)$$
$$= < \theta_{x|\neg y} | B_S^e, \xi > = \int \theta_{x|\neg y} \; \rho(\theta_{x|\neg y} | B_S^e, \xi)$$

As we have mentioned, however, the terms $< \theta_{x|y} | B_S^e, \xi >$ and $< \theta_{x|\neg y} | B_S^e, \xi >$ are just the user's probabilities for x given y and x given $\neg y$, respectively, in the next case. Hence, the user's prior network would contain no arc between x and y.

In any given problem, it is likely that $p(U | B_{S_C}^e, \xi)$ will not be equal to $p(U|\xi)$, because the latter represents an average over all conditioning events B_S^e. In principle, this makes the assessment of the prior network problematic, because a user is likely to prefer to assess the network without having to condition on the event $B_{S_C}^e$. In practice, it remains to be seen if this difference poses a significant problem.

To see how a user can assess K, consider the following observation. Suppose a user was initially ignorant about a domain—that is, his distribution $\rho(\Phi | B_{S_C}^e, \xi)$ was given by Equation 15 with each exponent $e_{x_1,\ldots,x_n} = -1$.[2] Then, from Equation 19, K must be the number of cases he has seen since he was ignorant. Sometimes, however, the user may have obtained knowledge about a domain through word of mouth, through common sense reasoning, or by reading texts. To handle such cases, we note that K is related to the user's confidence in his assessment of the prior belief network for U—the higher the value of K, the greater the user's confidence. Therefore, the user can assess K by judging the number of cases he would have had to have seen, starting from ignorance, to obtain his actual degree of confidence in the prior belief network. These judgments can be difficult, but can be made accurate by *calibrating* the user against other, more tedious methods for assessing Dirichlet distributions [Winkler, 1967].

The constant K is often called an *equivalent sample size*. It acts as a gain control for learning—the smaller the value of K, the more quickly BDe will favor network structures that differ from the prior belief-network structure. The constraints on the parameters N'_{ijk} (Equation 13) have a simple interpretation in terms of equivalent sample sizes. Namely, by an argument similar to that in the previous paragraph, we can think of the term

$$K_{ij} \equiv \sum_{k=1}^{r_i} (N'_{ijk} + 1) = N'_{ij} + r_i \quad (20)$$

as the equivalent sample size for the parameter set Θ_{ij}—the multinomial parameters for x_i, given that we have observed the jth instance of Π_i. From Equation 13, we see that

$$K_{ij} = K \cdot p(\Pi_i = j | B_S^e, \xi) \quad (21)$$

[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 -1 plus some number close to zero. Also note that many researchers consider ignorance to be the situation described by a constant density function—that is, all exponents $e_{x_1,\ldots,x_n} = 0$. This difference is not important for our discussion.

That is, the equivalent sample size for Θ_{ij} is just the overall equivalent sample size K times the probability that we see $\Pi_i = j$. We note for future discussion that

$$N'_{ijk} + 1 = K_{ij} \cdot p(x_i = k|\Pi_i = j, B^e_{S_C}, \xi) \qquad (22)$$

which follows from the fact that $(N'_{ijk} + 1)/K_{ij}$ is the expectation of θ_{ijk} given $B^e_{S_C}$.

SDLC describe an *ad hoc* approach for combining user knowledge with data that is closely related to ours. First, as we do, they asses a prior belief network (although they do not ask the user to condition on $B^e_{S_C}$). Then, for each variable x_i and each instance j of Π_i in the prior network, they allow the user to specify an equivalent sample size K_{ij}. From these assessments, SDLC compute equivalent sample sizes K_{ij} for other network structures. The description of this step requires some new notation. Let $\Pi_i(P)$ and $\Pi_i(B_S)$ denote the parents of x_i in the prior belief network and a given belief network B_S, respectively. Let $I_i = \Pi_i(P) \cap \Pi_i(B_S)$, $O_i = \Pi_i(P) \setminus I_i$, $N_i = \Pi_i(B_S) \setminus I_i$, indicating the *I*ntersecting, *O*ld, and *N*ew parents of x_i. Note that $O_i \cup I_i = \Pi_i(P)$ and $N_i \cup I_i = \Pi_i(B_S)$. With respect to the prior belief network, let $K_{ij(O)j(I)}$ denote the equivalent sample size for x_i, given the $j(O)$th instance of O_i and the $j(I)$th instance of I_i. First, SDLC *expand* this given equivalent sample size to include the parents N_i yielding

$$\begin{aligned} K_{ij(O)j(I)j(N)} &= K_{ij(O)j(I)} \\ &\cdot p(N_i = j(N)|O_i = j(O), I_i = j(I), \xi) \end{aligned} \qquad (23)$$

Then, they *contract* out the old parent nodes 0_i that are not in B_S to give

$$K_{ij(I)j(N)} = \sum_{j(O)} K_{ij(O)j(I)j(N)} \qquad (24)$$

where $K_{ij(I)j(N)}$ corresponds to an equivalent sample size K_{ij} for some j in the belief network B_S. Finally, they use Equation 22 (without conditioning on $B^e_{S_C}$) to obtain the exponents N'_{ijk}.

Their approach has two theoretical problems. One, they overlook the need to condition the assessment of the prior network on $B^e_{S_C}$. As mentioned, however, this problem may not be significant in practice. Two, their procedure will generate exponents that satisfy score equivalence if and only if the assessments of K_{ij} satisfy Equation 21, in which case, their approach is identical to our BDe metric. That is, their method allows the user too much freedom. Nonetheless, in some situations, one may want to sacrifice score equivalence for this greater latitude.

To complete the information needed to compute the BDe metric, the user must assess the prior probabilities on the network structures, $p(B^e_S|\xi)$. These assessments are logically independent of the assessment of the prior belief network, except in the limit as K approaches infinity, when the prior belief-network structure must receive a prior probability of one. Nonetheless, structures that closely resemble the prior belief network tend to have higher prior probabilities.

Here, we propose the following parametric formula for $p(B^e_S|\xi)$ that makes use of the prior belief 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 belief 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

$$p(B^e_S|\xi) = c\ \kappa^\delta \qquad (25)$$

where c is a normalization constant.

We choose this parametric form, because it facilitates efficient search. Applied as is, the formula destroys the property of score equivalence. In principle, however, we can recover the property by identifying within each equivalence class the belief-network structure with the highest prior probability, and then applying this prior probability to each belief-network structure in that equivalence class. When we use these metrics in conjunction with search, this solution is not necessary, because the search procedure will automatically favor the belief-network structure in an equivalence class with the highest prior probability (although the search procedure may not find this network structure).

2.4 Simple Example

Consider a domain U consisting of binary variables x and y. Let $B_{x\to y}$ and $B_{y\to x}$ denote the belief-network structures where x points to y and y points to x, respectively. Suppose that $K = 12$ and that the user's prior network gives the joint distribution $p(x, y|B^e_{x\to y}, \xi) = 1/4$, $p(x, \bar{y}|B^e_{x\to y}, \xi) = 1/4$, $p(\bar{x}, y|B^e_{x\to y}, \xi) = 1/6$, and $p(\bar{x}, \bar{y}|B^e_{x\to y}, \xi) = 1/3$. According to Equation 14, if we observe database D containing a single case with both x and y true, we obatin

$$p(D, B^e_{x\to y}|\xi) = p(B^e_{x\to y}|\xi) \cdot \frac{11!}{12!}\frac{6!}{5!}\frac{5!}{6!}\frac{3!}{2!}$$

$$p(D, B^e_{y\to x}|\xi) = p(B^e_{y\to x}|\xi) \cdot \frac{11!}{12!}\frac{5!}{4!}\frac{4!}{5!}\frac{3!}{2!}$$

Thus, as required, the BDe metric exhibits the property of score equivalence.

2.5 Causal Networks and Scoring Metrics

People often have knowledge about the causal relationships among variables in addition to knowledge about conditional independence. Such causal knowledge is stronger than is conditional-independence knowledge, because it allows us to derive beliefs about a domain after we intervene. For example, most of us believe that smoking causes lung cancer. From this belief, we infer that if we stop smoking, then we decrease our chances of getting lung cancer. In contrast, if we were to believe that there is only a statistical correlation between smoking and lung cancer, perhaps because there is a gene that causes both our desire to smoke and lung cancer, then we would infer that giving up cigarettes would not decrease our chances of getting lung cancer.

Causal networks, described by Pearl and Verma (1991), Spirtes et al. (1993), Druzdel and Simon (1993), and Heckerman and Shachter (1994) represent such causal relationships among variables. In particular, a causal network for U is a belief network for U, wherein it is asserted that each nonroot node x is caused by its parents. The precise meaning of cause and effect is not important for our discussion. The interested reader should consult the previous references.

Formally, we define a causal network to be a pair (C_S, C_P), where C_S is a causal-network structure and C_P is a set of probability distributions corresponding to that structure. In addition, we define C_S^e to be the event corresponding to C_S, and a metric for a causal network to be $p(D, C_S^e|\xi)$. In contrast to the case of belief networks, it is not appropriate to require the properties of event equivalence or score equivalence. For example, consider a domain containing two variables x and y. Both the causal network C_{S1} where x points to y and the causal network C_{S2} where y points to x represent the assertion that x and y are dependent. The network C_{S1}, however, in addition represents the assertion that x causes y, whereas the network C_{S2} represents the assertion that y causes x. Thus, the events C_{S1}^e are C_{S2}^e are not equal. Indeed, it is reasonable to assume that these events—and the events associated with any two different causal-network structures—are mutually exclusive.

In principle, then, a user may assign a different prior distribution to the parameters Θ_{C_S} to every complete causal-network structure, in effect choosing arbitrary values for the exponents N'_{ijk}. This approach leads to the Bd and BD metrics. For practical reasons, however, the assessment process should be constrained. SDLC's expansion–contraction method described in Section 2.3 is one approach, but is computationally expensive. CH's specialization of the BD metric, wherein they set each N'_{ijk} to zero is efficient, but ignores the prior network. We have explored a simple approach, wherein each K_{ij} is equal to K, a constant, and where the exponents N'_{ijk} are determined from a prior network using Equation 22. We call this metric the BDu metric ("u" stands for uniform equivalent sample sizes). Of course, the BDe metric may also be used to score causal networks.

Note that, in the context of causal networks, the assumption of parameter modularity (Assumption 5) has an appealing justification. Namely, we can imagine that a causal mechanism is responsible for the interaction between each node and its parents. The assumption of parameter modularity then follows from the assumption that each such causal mechanism is independent.

3 Methods for Finding Network Structures with High Scores

For a given database D and background knowledge ξ, Equation 12 with prior probabilities given by Equation 25 can be written

$$p(D, B_S^e|\xi) = \prod_{i=1}^{n} s(x_i|\Pi_i) \tag{26}$$

where $s(x_i|\Pi_i)$ is a function only of x_i and its parents. Therefore, we can compare the score for two network structures that differ by the addition or deletion of one arc pointing to x_i, by computing only the term $s(x_i|\Pi_i)$ for both structures. The algorithms that we examine make use of this property, which we call *score locality*. This property is due to the assumption that cases are complete and the assumption of parameter modularity.

3.1 Special Case Polynomial Algorithms

We first consider the special case of finding a network structure with the highest score among all structures in which every node has at most one parent. For each arc $x_j \to x_i$ (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 26, we have

$$\begin{aligned}\log p(D, B_S^e|\xi) &= \sum_{i=1}^{n} \log s(x_i|\pi_i) \\ &= \sum_{i=1}^{n} w(x_i, \pi_i) + \sum_{i=1}^{n} \log s(x_i|\emptyset)\end{aligned} \tag{27}$$

where π_i is the (possibly null) parent of x_i. The last term in the second line of Equation 27 is the same for all network structures. Thus, among the network structures in which each node has at most one parent, the one with the highest score is the one for which $\sum_{i=1}^{n} w(x_i, \pi_i)$ is a maximum.

Finding this network structure is a special case of a well-known problem of finding *maximum branchings*. A *tree-like network* is a 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 $\sum_{i=1}^{n} w(x_i, \pi_i)$. An efficient polynomial algorithm for finding a maximum branching was first described by Edmonds (1967).

Edmonds' algorithm can be used to find the maximum branching regardless of the score we use, as long as one can associate a weight with every edge. Therefore, this algorithm is appropriate for any metric. When scoring belief networks, however, due to the property of score equivalence, 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 $x_i \to x_j$ 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, and the problem reduces to finding an undirected forest for which $\sum w(x_i, x_j)$ is a maximum. Therefore, 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 isomorphic belief-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.

3.2 Heuristic Search

A generalization of the problem described in the previous section is to find the best network structure from the set of all structures in which each node has no more than k parents. Because networks with large parent sets are not very useful in practice, one might be tempted to generalize the previous algorithm to some small $k > 1$. Unfortunately, we conjecture that finding an optimal network structure is NP-hard for $k > 1$. Thus, we use heuristic search.

The search algorithms we consider make successive arc changes to the network structure, and employ the property of score locality 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 network structure, and $\Delta(e)$ to denote the change in log score of the network resulting from the modification $e \in E$. From the property of score locality, 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 method for search is a variant of the greedy hill-climbing algorithm described by Lam and Bacchus (1993). First, we choose a network structure (described in the following paragraph). 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 $\Delta(e)$. Using score locality, we can avoid recomputing all terms $\Delta(e)$ after every change. In particular, with an exception to be noted, if neither x_i, x_j, nor their parents are changed, then $\Delta(e)$ remains valid for all changes e involving these nodes. The exception occurs because some changes become possible and some become impossible due to the noncyclicity condition; the terms $\Delta(e)$ for these changes must be scored or invalidated, respectively. Candidates for the initial network structure include the empty graph, a random graph, a graph determined by one of the polynomial algorithms described in the previous section, a graph determined by Singh's method of initialization [Singh and Valtorta, 1993], and the prior network.

Another search method suited to our task is a variant of simulated annealing, described by Metropolis et al. (1953). In this method, we initialize the system at some temperature T_0. Then, we pick some eligible change e at random, and evaluate the expression $p = e^{\frac{\Delta(e)}{T_0}}$. If $p > 1$, then we make the change e; otherwise, we make the change with probability p. We repeat this selection and evaluation process, n times or until we make m changes. If we make no changes in n repetitions, we stop searching. Otherwise, we lower the temperature by multiplying the current temperature T_0 by a decay factor $0 < \beta < 1$, and continue the search process. We stop searching if we have lowered the temperature more than l times. Thus, this algorithm is controlled by five parameters: T_0, n, m, l and β. 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 previously.

4 Experimental Results

We have implemented the BDe and BDu metrics as well as the search algorithms described in this paper. Our implementation is in the C++ programming language, and runs under Windows NT^{TM} with a 486-66Mz processor. We have tested our algorithms on small networks ($n \leq 5$) as well as the 36-node Alarm network for the domain of ICU ventilator management [Beinlich et al., 1989]. Here, we describe some of the more interesting results that we obtained using the Alarm network. We note that the comparison of the BDe and BDu metrics may be viewed as a comparison of two exact metrics in the context of causal networks, or the comparison of an exact and approximate (i.e., non-score equivalent) metric in the context of belief networks.

In our evaluations we start with a given network, which we call the *gold-standard network*. Next, we generate a database from the given network, using a Monte-Carlo technique. Then, we use a scoring metric and search procedure to identify a high-scoring network structure, and use the database and prior knowledge to populate the probabilities in the new network, called the *learned network*. In particular, we set each probability $p(x_i = k|\Pi_i = j)$ to be the posterior mean of θ_{ijk}, given the database.

A principled candidate for our accuracy measure 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., a network 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 to construct utility functions and decision scenarios in practice. Consequently, researchers have used surrogates for differences in utility, such as the mean square error and cross entropy.

In this paper, we use the cross-entropy measure. In particular, let $q(x_i, \ldots, x_n)$ and $p(x_i, \ldots, x_n)$ denote the probability of an instance of U obtained from the gold-standard and learned networks, respectively. Then we measure the accuracy of a learning algorithm using the cross entropy $H(q,p)$, given by

$$H(q,p) = \sum_{x_1,\ldots,x_n} q(x_i,\ldots,x_n) \log \frac{q(x_i,\ldots,x_n)}{p(x_i,\ldots,x_n)} \tag{28}$$

The lower the value of the cross entropy, the more accurate the algorithm. In Heckerman et al. (1994), we describe a method for computing the cross entropy of two networks that makes use of the network structures.

Other researchers have used a structural comparison of the gold-standard and learned networks—essentially, counting arc differences—as a surrogate for utility difference [Singh and Valtorta, 1993]. We have not found this measure to be as useful as cross entropy, because the former measure fails to quantify the strength of an arc. For example, although there may be an arc from node x to node y, the conditional probability of y given x may be almost the same for different values of x. In effect, the arc is a very *weak* one. The cross-entropy measure takes this effect into account, whereas a structural comparison does not. It can be argued that the presence of many weak arcs is undesirable, because it increases inference time significantly. We believe that such concerns should be expressed explicitly, by including cost of inference in the measure of network utility. Such information not only enhances evaluation, but it can be used in the scoring metrics themselves.

In our experiments, 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. To generate the prior network, we first add 2η arcs to the gold-standard network, creating network structure B_{S1}. When we add an arc, we copy the probabilities in 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 η arcs and reversing up to 2η arcs (a reversal may create a directed cycle, in which case, the reversal is not done). 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.

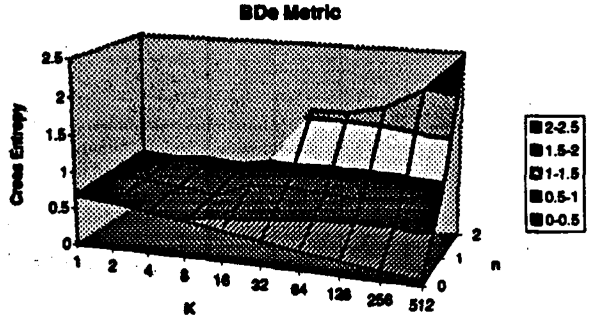

Figure 1: Cross entropy of learned networks with respect to the Alarm network (inverse learning accuracy) as a function the deviation of the prior-network from the Alarm network (η) and the user's equivalent sample size (K) for the BDe metric with prior parameter κ set to $(1/(K+1))^{10}$. Greedy search initialized with the prior network was applied to databases of size 500. Each data point represents an average over four learning instances.

Figure 1 shows the cross entropy of learned networks with respect to the Alarm network (inverse learning accuracy) as a function of the deviation of the prior-network from the gold- standard network (η) and the user's equivalent sample size (K) for the BDe metric. In this experiment, we used our greedy hill- climbing algorithm initialized with the prior network, and 500-case databases generated from the Alarm network. For each value of η and K, the cross-entropy values shown in the figure represent an average over four learning instances, where in each instance we used a different database and prior network. The databases and prior networks generated for a given value of η were used for all values of K. We made the prior parameter κ a function of K—namely, $\kappa = (1/(K+1))^{10}$—so that it would take on reasonable values at the extremes of K. (When $K = 0$, reflecting complete ignorance, all network structures receive the same prior probability. Whereas, in the limit as K approaches infinity, reflecting complete confidence, the prior network structure receives a prior probability of one.)

The qualitative behavior of the curve is reasonable.

Namely, when $\eta = 0$—that is, when the prior network was identical to the Alarm network—learning accuracy increased as the equivalent sample size K 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 K associated with optimal accuracy. This result is not surprising. If K is too large, then the deviation between the true values of the parameters and their priors degrade performance. On the other hand, if K is too small, the metric is ignoring useful prior knowledge. We speculate that results of this kind can be used to calibrate users in the assessment of K.

The results for the BDu metric were almost identical. At 27 of the 30 data points in Figure 3, the average cross entropies for the two metrics differed by less than 0.3. 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 (i.e., a network where all variables are independent) whose marginal probabilities are determined from the Alarm network is 13.6.

Learning times for the two metrics differed considerably. Table 1 shows average run times for the BDe and BDu metrics as a function of η. For both metrics, search times increased as η increased, and learning times for the BDe metric were greater than those for the BDu metric. These behaviors are due to the fact that inference—that is the computation of N'_{ijk} (Equations 21 and 22)—dominate learning times. Indeed, when we did not use a prior network, but instead assumed all probabilities $p(U|B^e_{S_C},\xi)$ were uniform and used a maximum spanning tree to initialize greedy search, learning times for the Alarm network dropped to approximately 45 seconds. Thus, when prior networks were used, run times increased when η increased, because the prior networks became more complex. Also, the run times associated with BDe were greater, because the computation of the metric included determinations of $p(x_i = k, \Pi_i = j|B^e_{S_C},\xi)$, whereas the computation of the BDu metric involved determinations of $p(x_i = k|\Pi_i = j, B^e_{S_C},\xi)$. The former computations are more complex using the Jensen inference algorithm [Jensen et al., 1990], which we employed in our initial implementation. In subsequent implementations, we plan to use a query-based inference method, such as Symbolic Probabilistic Inference [D'Ambrosio, 1991]. We expect that both the BDe and BDu learning times, as well as their differences, will decrease substantially.

We found our greedy hill-climbing algorithm to be the best of our algorithms for learning network structures in the Alarm domain. Table 2 shows cross entropy and learning times for each search algorithm. In this comparison, we used the BDe metric with $K = 8$ and $\kappa = 1$ (uniform priors $p(B^e_S|\xi)$), uniform probabilities $p(U|B^e_{S_C},\xi)$, and a database size of 8000. The algorithms CH opt is the greedy algorithm described by CH initialized with an ordering that is consistent with the Alarm network. The algorithm CH rev is the same algorithm initialized with the reversed ordering. We included this algorithm to gauge the sensitivity of the CH algorithm to ordering. Our greedy algorithm was initialized with a maximum spanning tree, as described in Section 3.1. The annealing algorithm used parameters $T_0 = 100, l = 70, m = 500, n = 1000$, and $\beta = 0.9$, which we determined to yield reasonable accuracy after some experimentation. For comparison, we computed the cross entropy of the Alarm network and a network whose structure was identical to the Alarm network, and whose probabilities were determined from the posterior means of its parameters, given the database (see row labeled "gold standard"). Our greedy algorithm obtained the lowest cross entropy of all algorithms, and was only slightly slower than was the CH algorithm. Also, the learning accuracy and execution time for the CH algorithm was sensitive to the variable ordering provided to the algorithm, whereas our greedy algorithm required no additional information for initialization. Our annealing algorithm did poorly both with respect to cross entropy and learning time.

Table 1: Average learning times for the Alarm domain using greedy search initialized with a prior network and databases of size 500.

η	BDe	BDu
0	17 min	7 min
1	46 min	15 min
2	70 min	20 min

Table 2: Cross entropy and learning times for various search algorithms.

	cross entropy	learning time
CH opt	0.036	3 min
CH rev	0.223	4.5 min
greedy search	0.035	5.5 min
annealing	0.098	150 min
gold standard	0.027	na

Acknowledgments

We thank Wray Buntine, Greg Cooper, and anonymous reviewers for useful suggestions. In addition, we thank Koos Rommelse for assisting with the implementation of the learning and evaluation algorithms.

References

[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.

[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, 1991] Cooper, G. and Herskovits, E. (1991). A Bayesian method for constructing Bayesian belief networks from databases. In *Proceedings of Seventh Conference on Uncertainty in Artificial Intelligence*, Los Angeles, CA, pages 86–94. Morgan Kaufmann.

[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.

[D'Ambrosio, 1991] D'Ambrosio, B. (1991). Local expression languages for probabilistic dependence. In *Proceedings of Seventh Conference on Uncertainty in Artificial Intelligence*, Los Angeles, CA, pages 95–102. Morgan Kaufmann.

[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.

[Druzdel and Simon, 1993] Druzdel, 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.

[Geiger and Heckerman, 1994] Geiger, D. and Heckerman, D. (1994). Learning Gaussian networks. Technical Report MSR-TR-94-10, Microsoft.

[Heckerman et al., 1994] Heckerman, D., Geiger, D., and Chickering, D. (1994). Learning Bayesian networks: The combination of knowledge and statistical data. Technical Report MSR-TR-94-09, Microsoft.

[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 timeless view of causality. In *Proceedings of Tenth Conference on Uncertainty in Artificial Intelligence*, Seattle, WA. Morgan Kaufmann.

[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.

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

[Jensen et al., 1990] Jensen, F., Lauritzen, S., and Olesen, K. (1990). Bayesian updating in recursive graphical models by local computations. *Computational Statisticals Quarterly*, 4:269–282.

[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.

[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*, To appear.

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

[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.

[Singh and Valtorta, 1993] Singh, M. and Valtorta, M. (1993). An algorithm for the construction of Bayesian network structures from data. In *Proceedings of Ninth Conference on Uncertainty in Artificial Intelligence*, Washington, DC, pages 259–265. Morgan Kaufmann.

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

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

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