Causal Independence for Probability Assessment and Inference Using Bayesian Networks

David Heckerman and John S. Breese

Abstract— **A Bayesian network is a probabilistic representation for uncertain relationships, which has proven to be useful for modeling real-world problems. When there are many potential causes of a given effect, however, both probability assessment and inference using a Bayesian network can be difficult. In this paper, we describe causal independence, a collection of conditional independence assertions and functional relationships that are often appropriate to apply to the representation of the uncertain interactions between causes and effect. We show how the use of causal independence in a Bayesian network can greatly simplify probability assessment as well as probabilistic inference.**

I. Introduction

A Bayesian network is a modeling and inference tool for problems involving uncertainty [1], [2]. The representation rigorously describes probabilistic relationships, yet includes a human-oriented qualitative structure that facilitates communication between the user and the probabilistic model. Consequently, the representation has proven to be useful for modeling many real-world problems including diagnosis, forecasting, automated vision, sensor fusion, manufacturing control, and information retrieval [3].

To be more technical, a Bayesian network encodes a joint probability distribution over a set of random variables. A variable may be discrete, having a finite or countable number of states, or it may be continuous. Unless otherwise noted, we discuss Bayesian networks for discrete variables having a finite number of states. In describing a Bayesian network, we use lower-case letters to represent single variables and upper-case letters to represent sets of variables. We write $x = k$ to denote that variable x is in state k. When we observe the state for every variable in set X, we call this set of observations a *state* of X. The *joint space* of a set of variables U is the set of all states of U. The *joint probability distribution* over U is the probability distribution over the joint space of U. We use $p(X \mid Y)$ to denote the set of joint probability distributions over X, each one conditional on every state in the joint space of Y.

A *problem domain* is a set of variables. A Bayesian network for the domain $U = \{x_1, \ldots, x_n\}$ consists of a set of *local* conditional probability distributions, combined with a set of assertions of conditional independence that allow us to construct the global joint distribution over U from the local distributions. The decomposition is based on the chain rule of probability, which dictates that

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

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

$$p(x_i \mid x_1, \ldots, x_{i-1}) = p(x_i \mid \Pi_i). \tag{2}$$

The idea is that the distribution of x_i can often be described conditional on a set Π_i that is substantially smaller than the set $\{x_1, \ldots, x_{i-1}\}$. Given these sets, a Bayesian network can be described in part as a directed acyclic graph such that each variable $x_1, \ldots, x_n$ corresponds to a node in that graph, and 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.) Note that, because the parents in the graph coincide with the conditioning sets Π_i, the Bayesian network structure directly encodes the assertions of conditional independence in (2).

In a Bayesian network, each node x_i is associated with the conditional probability distributions $p(x_i \mid \Pi_i)$—one distribution for each state of Π_i. These distributions may be directly assessed, learned from data, or determined from a combination of prior knowledge and data [4]. From (1) and (2), we see that any Bayesian network for $\{x_1, \ldots, x_n\}$ uniquely determines a joint probability distribution for those variables. That is,

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i \mid \Pi_i). \tag{3}$$

The structure of a Bayesian network will depend on how the variables are ordered in the expansion of (1). If the order is chosen carelessly, the resulting network structure may fail to reveal many conditional independencies in the domain. In practice, however, domain experts often can readily assert causal relationships among variables in a domain; and we can use these assertions to construct a Bayesian-network structure without preordering the variables. Namely, to construct a Bayesian network for a given set of variables, we draw arcs from cause variables to their immediate effects. In almost all cases, doing so results in a Bayesian network whose conditional-independence implications are accurate. For example, we used cause-and-effect considerations to construct the Bayesian-network structure shown in Fig. 1. The network is used for troubleshooting printing problems within the Windows$^{\text{TM}}$ operating system. The connection between causation and conditional independence is discussed in detail in (e.g. [5]–[7]. Statistical techniques for learning Bayesian-network structure from data or a combination of data and expert knowledge are also available [8]–[12].

Because a Bayesian network for any domain determines a joint probability distribution for that domain, we can—in principle—use a Bayesian network to compute any probability of interest. For example, suppose we have the simple Bayesian network for discrete variables with structure $w \rightarrow x \rightarrow y \rightarrow z$, and we want to know $p(w \mid z)$. From the rules of probability we have

$$p(w \mid z) = \frac{p(w, z)}{p(z)} = \frac{\sum_{x,y} p(w, x, y, z)}{\sum_{w,x,y} p(w, x, y, z)} \tag{4}$$

where $p(w, x, y, z)$ is the joint distribution determined from the Bayesian network. In practice, this approach is not feasible, because it entails summing over an exponential number of terms. Fortunately, we can exploit the conditional independencies encoded in a Bayesian network to make this computation more efficient. In this case, given the network structure, (4) becomes

$$\begin{aligned} p(w \mid z) &= \frac{\sum_{x,y} p(w, x, y, z)}{\sum_{w,x,y} p(w, x, y, z)} \\ &= \frac{p(w) \sum_x p(x \mid w) \sum_y p(y \mid x) p(z \mid y)}{\sum_w p(w) \sum_x p(w) p(x \mid w) \sum_y p(y \mid x) p(z \mid y)}. \end{aligned} \tag{5}$$

That is, using conditional independence, we can often reduce the dimensionality of the problem by rewriting the sums over multiple

Manuscript received August 28, 1994; revised November 1, 1995.

The authors are with Microsoft Research, Redmond, WA, 98052-6399 USA (e-mails: heckerma@microsoft.com, breese@microsoft.com).

Publisher Item Identifier S 1083-4427(96)05368-4.

1083–4427/96$05.00 © 1996 IEEE

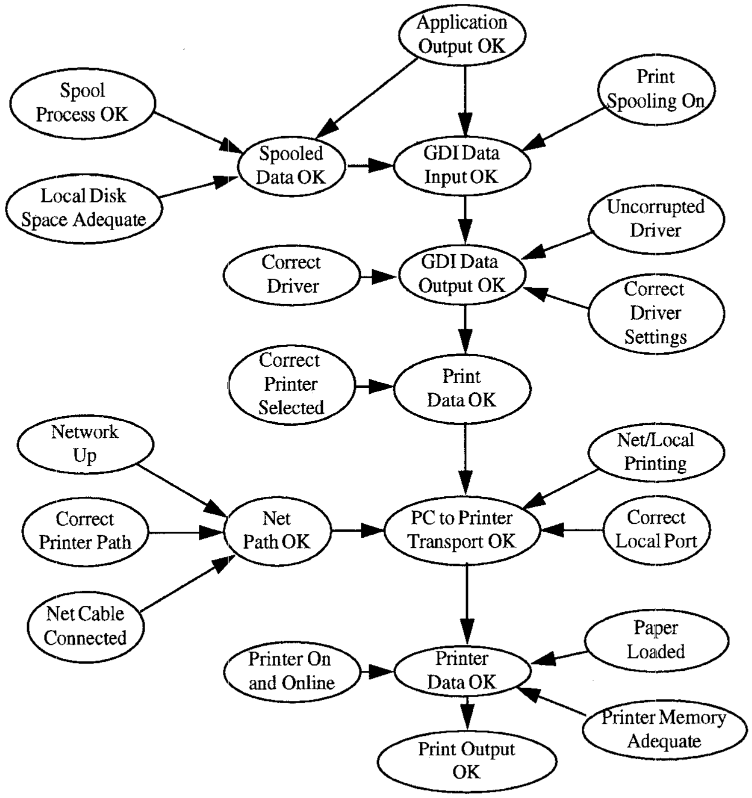
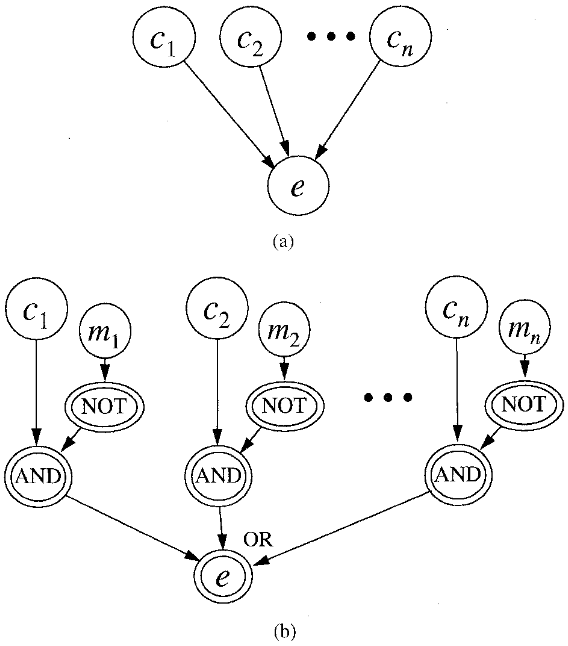


Fig. 1. A Bayesian-network structure for troubleshooting a printing problem. Arcs are drawn from cause to effect.

variables as the product of sums over a single variable (or at least smaller numbers of variables). The general problem of computing probabilities of interest from a (possibly implicit) joint probability distribution is called *probabilistic inference.* Several researchers have developed algorithms for exact probabilistic inference that make use of the conditional independencies represented in a Bayesian network for discrete variables [2], [13], [14]–[16].

In this paper, we examine an important weakness of the Bayesian-network representation. When modeling the real world, we often encounter situations in which an effect has many potential causes. In these situations, probability specification and inference can be impractical if not impossible. For example, suppose we have n binary (two-state) causes $c_1, \ldots, c_n$ bearing on a single binary effect e, as shown in Fig. 2(a).[1] According to the definition of Bayesian networks, we must specify the probability distribution of e conditional on every state of its parents. Thus, in this example, we must specify 2^n probability distributions for the node. Furthermore, probabilistic inference (e.g., the computation of $p(c_1 \mid e)$) has time complexity $O(2^n)$.

To overcome this limitation of the representation, Kim and Pearl [17] introduced the *noisy-OR* model. The model, which assumes that causes and effect are binary with states true and false, is shown in Fig. 2(b). The nodes m_i represent inhibitory causal mechanisms, each of which have two states—true and false. Nodes in the figure with double borders, called *deterministic nodes,* are deterministic functions of their parents. In particular, a node labeled *AND* assumes a state given by the conjunction of the node's parents. Nodes labeled *NOT* and *OR* have corresponding relationships. Thus, in the noisy-OR model, each inhibitory causal mechanism m_i prevents its corresponding cause c_i from producing the effect; and the effect will be false only if all the inhibitory causal mechanisms associated with present causes are active. In addition, as is indicated by the lack of arcs between the nodes m_i, causal mechanisms are mutually independent.



(a)

(b)

Fig. 2. (a) A Bayesian network structure for multiple causes and a single effect. (b) A noisy-OR model for the multiple-cause interaction in (a).

[1] To avoid clutter, we do not show the probability distributions in this or other Bayesian networks in this paper.

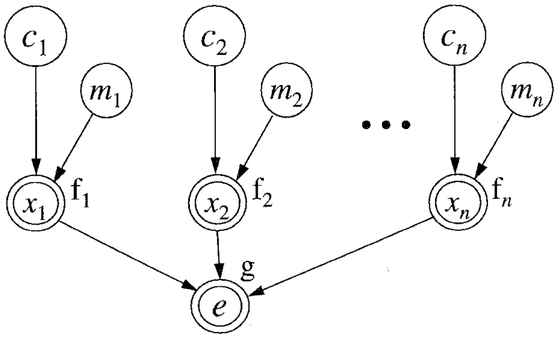


Fig. 3. A Bayesian network depicting general causal independence.

As an example of the noisy-OR model, consider the interactions among the node *Spooled Data OK* and its parents in the print troubleshooter model. Although the spool process may be bad for a given font due to a programming bug, this cause of bad spooled output will be inhibited if the document being printed does not use that font. Also, local disk space may be inadequate, but this cause of bad spooled output will be inhibited if the print job is small. Thus, we can use the noisy-OR model to capture these relationships.

Henrion [18] extended the noisy-OR model to include situations where the effect can be true even when all described causes are false. In this extension, we include a dummy or *leak cause*, which is always set to true. This single cause represents other causes not described that may be contributing to the effect.

Because mechanisms are independent in the noisy-OR model, use of the model leads to a significant reduction in the number of probabilities required to quantify the cause-effect interaction. Namely, whereas the unrestricted model requires 2^n probabilities, the noisy-OR model requires only n probabilities: one probability for each causal mechanism. Consequently, probability assessment is simplified, and learning algorithms are more accurate (assuming the model is correct).

The noisy-OR model has been generalized in several ways [19]–[22]. In this paper, we describe these generalizations, which collectively we call *causal independence*, and show how these models are related to one another. In addition, we show how the use of causal independence leads to simplifications in probability assessment and probabilistic inference. Use of the noisy-OR model to improve the learning of probabilities is discussed in [23].

II. General Causal Independence

Causal independence is a straightforward generalization of the noisy-OR model, and is depicted in the Bayesian network of Fig. 3. In this model, the causes, effect, causal mechanisms, and intermediate nodes (x_i) may be discrete or continuous. In addition, each function f_i and the function g are unrestricted. Also, as in the case of the noisy-OR model, we assume that the causal mechanisms are mutually independent—hence the name for the model. Note that, because causes and the intermediate nodes are no longer restricted to two states, we should not interpret the causal mechanisms as necessarily inhibitory. Rather, the mechanisms represent a more general mapping from cause to effect. A slightly less general form of causal independence is described by Srinivas [19].

As is true for the noisy-OR model, use of the causal-independence model simplifies the quantification of the cause-effect interaction, because the causal mechanisms are mutually independent. For example, assuming all variables are discrete with a finite number of states, we can quantify the interaction by specifying $n + 1$ functions and a number of probabilities that is linear in n. In contrast, to quantify the unrestricted model where e has parents $c_1, \ldots, c_n$, we require a number of probabilities that is exponential in n.

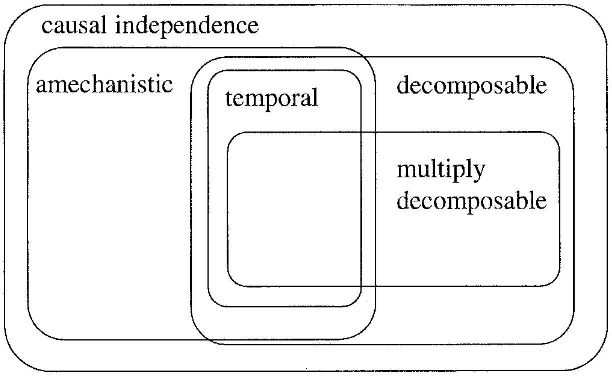


Fig. 4. A Venn diagram showing the various specializations of causal independence and their relationships.

Let us consider some examples of causal independence that have been used in real-world applications. In the *noisy-MAX model*, each cause has a distinguished state "absent" or "off," each intermediate node x_i takes on consecutive integer values between 0 and an upper bound (possibly infinity), and the effect e takes on consecutive integer values between 0 and a bound equal to the largest bound on the intermediate nodes. The function g is the MAX function $g(x_{ij}, \ldots, x_n) = \max(x_{ij}, \ldots, x_n)$; and each function f_i has the restriction that $f_i(c_i = \text{absent}, m_i) = 0$. Note that the noisy-MAX model is equivalent to the noisy-OR model when each intermediate node has states $\{0, 1\}$.

In the *noisy-addition model*, each cause has a distinguished state "absent" or "off,"[2] each intermediate node x_i takes on consecutive integer values between some lower and upper bound (possibly ± infinity), and e takes on integer values. The function g is addition; and each function f_i has the restriction that $f_i(c_i = \text{absent}, m_i) = 0$.

The most commonly used example of causal independence where causes and effect are continuous is the linear-Gaussian model, given by

$$e = a + \sum_{i=1}^{n} b_i \cdot c_i + \varepsilon$$

where a and $b_{ij}, \ldots, b_n$ are constants and ε has a normal distribution with mean zero and variance v (written $N(0, v)$). We can describe this model in terms of causal independence as follows

$$x_i = b_i \cdot c_i, \quad i = 1, \ldots, n$$
$$x_{n+1} = m_{n+1} \sim N(0, v)$$

and

$$g(x_1, \ldots, x_{n+1}) = x_1 + \cdots + x_{n+1}.$$

III. Specific Forms of Causal Independence

In this section, we examine several specializations of the causal-independence model. The names of the models along with their relationships are depicted in Fig. 4. After describing each model, we identify its benefits for probability assessment and/or probabilistic inference. As is to be expected, the more specific models have added benefits, but are less generally applicable.

[2] Heckerman [21] describes special cases of the noisy-MAX and noisy-addition models where all causes are binary.

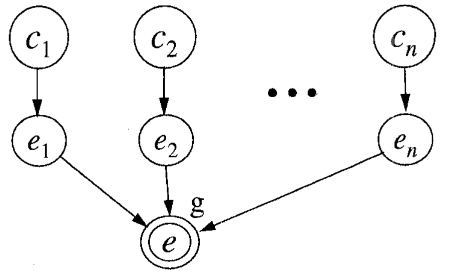


Fig. 5. A Bayesian network for amechanistic causal independence.

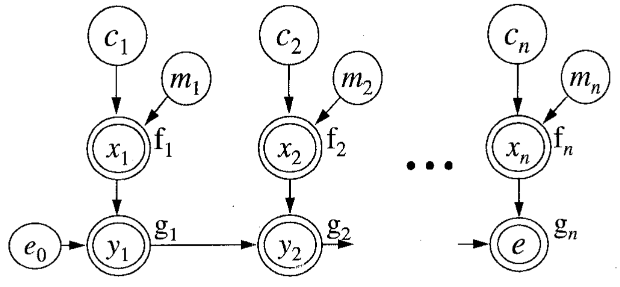


Fig. 6. A Bayesian network for decomposable causal independence.

A. Amechanistic Causal Independence

One problem with general causal independence is that it is sometimes difficult to identify specific causal mechanisms and intermediate nodes. This problem is avoided in *amechanistic causal independence*, first described by Heckerman and Breese [22] under the name atemporal causal independence.

In using this model, we designate some state of every cause to be *distinguished*. For most real-world models, this state will be the one that has no bearing on the effect—that is, the "absent" or "off" state—but we do not require this association. We use $*$ to denote the distinguished state for each cause. Also, for every cause c_i, we introduce an intermediate node e_i that corresponds to effect e had all causes but c_i been in their distinguished states. Finally, we assume that the e_i are mutually independent, and that e is deterministic function of $e_1, \ldots, e_n$, as is shown in Fig. 5.

The noisy-OR, noisy-MAX, noisy-addition, and linear-Gaussian models are examples of amechanistic causal independence. For example, to transform the noisy-OR model as described in Fig. 2(b) to the amechanistic form of Fig. 5, we (1) remove the causal-mechanism nodes m_i from the Bayesian network of Fig. 2(b),[3] (2) identify each node x_i in Fig. 2(b) with e_i in Fig. 5, and (3) take g in Fig. 5 to be the OR function.

Amechanistic causal independence has an interesting semantics. In particular, by definition of the intermediate nodes e_i, these nodes can not be simultaneously observed. Nonetheless, the model includes the assumption that these nodes are mutually independent. Philosophers call such an assumption a *counterfactual* [25], [26]—a statement that can not be verified by observation. Although this assumption may seem unusual, this and other counterfactual assumptions can be made rigorous in the context of a causal model [27], [6], [7].

We note that amechanistic causal independence has several model restrictions. Namely, each intermediate node e_i must have the same number of states as e. Also, let e_0 denote the state of e when all causes are in their distinguished state. Then, by definition of e_i, it follows that $e_i = e_0$ when $c_i = *$, and that $g(e_0, \ldots, e_0) = e_0$.

B. Decomposable Causal Independence

In many domains, the function g in the general causal-independence model can be decomposed into a series of two-argument functions, as shown in Fig. 6,[4] such that the number of states in each y_i is less than exponential in n. When this restriction is met, we say that the causal-independence model is *decomposable* [22]. An example of this form of causal independence is the noisy-OR model, where each $g_i(x, y) = \mathrm{OR}(x, y)$, and $e_0 = \text{false}$. The noisy-MAX, noisy-addition, and linear-Gaussian models are also examples of decomposable causal independence. A function g that does not yield decomposable causal independence is the r-of-n function, $0 < r < n, n > 2$, which takes binary inputs and returns 1 if and only if exactly r of its inputs are 1.

Unlike the forms of causal independence described in the previous sections, this form of has advantages for probabilistic inference. For domains where variables are discrete with a finite number of states, the computational complexity of exact inference is at least exponential in the number of parents of the node with the most parents, and complexity is often dominated by this factor. Thus, although decomposition increases the number of nodes in the Bayesian network, it decreases the number of parents of node e, thereby often leading to a reduction in inference complexity. For example, when causes and effect are binary variables, the computation of $p(c_1 \,|\, e)$ using the Bayesian network in Fig. 3 has complexity $O(2^n)$, whereas the same computation in the Bayesian network of Fig. 6 has complexity $O(n)$.

We can obtain even greater inference speedups when the function g can be decomposed for different orderings of the causes. For example, in the noisy-OR, noisy-MAX, noisy-addition, and linear-Gaussian models, we can change the ordering of the causes, and still obtain a model of the form shown in Fig. 6, because the functions OR, MAX, and addition are associative and commutative. We call this form of causal independence *multiply decomposable.*

To illustrate how this form of causal independence can further simplify probabilistic inference, consider the multiply-connected Bayesian network in Fig. 7(a). If we represent the cause-effect relationships in the form of Fig. 6 using the ordering (c_1, c_2, c_3), then we obtain the Bayesian network in Fig. 7(b).[5] In contrast, if we use the ordering (c_2, c_3, c_1), then we obtain the Bayesian network in Fig. 7(c). Inference using exact Bayesian-network algorithms typically will be less efficient in the Bayesian network of Fig. 7(b) than that in the Bayesian network of Fig. 7(c), because there is a larger undirected cycle in the former network. In Section IV, we examine inference speedups in more detail.

C. Temporal Causal Independence

The last form of causal independence that we consider, called *temporal causal independence* [21], is a special case of both amechanistic and decomposable causal independence. The model is depicted in Fig. 8(a). As in the case of amechanistic causal independence, we designate some state of every cause to be distinguished. In addition, for each cause c_i, we introduce an intermediate node e_i that corresponds to effect e had all causes but c_i been set to their distinguished states. In addition, for every cause c_i, we include a variable e_{ti} that represents the state of e had causes $c_{i+1}, \ldots, c_n$ been

[3] In removing a node from a Bayesian network, we obtain a new Bayesian network whose joint probability distribution is consistent with that of the original Bayesian network. The process of node removal in Bayesian networks is discussed in [24].

[4] We introduce the constant e_0 so that all functions g_i have two arguments.

[5] We have removed the causal mechanisms, intermediate nodes, and constant e_0 from the Bayesian network for simplicity. This removal does not change our argument.

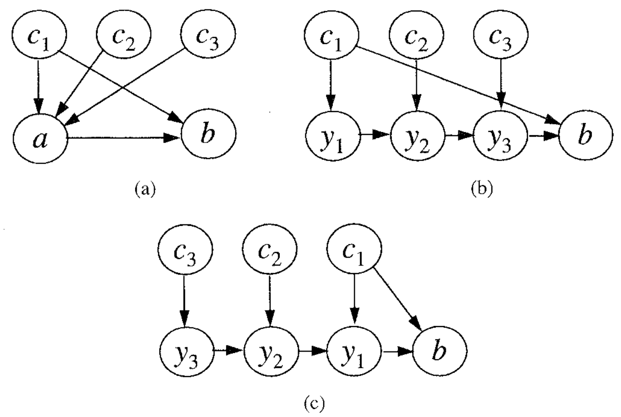


Fig. 7. (a) A multiply connected Bayesian network. (b)–(c) Two transformations of the Bayesian network in (a). The network in (c) has a smaller undirected cycle than that in (b).

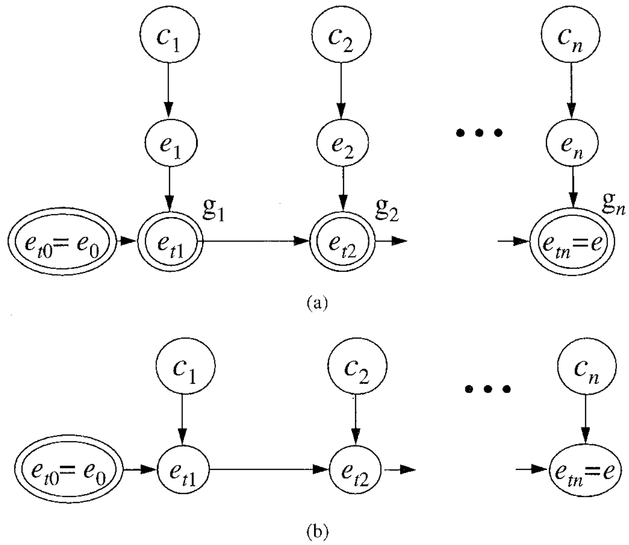


Fig. 8. (a) A Bayesian-network representation of temporal causal independence. (b) A reduced model for probability assessment.

set to their distinguished state. Finally, as in the case of decomposable causal independence, we assume that the intermediate variables are related by the functions $e_{ti} = g_i(e_{t(i-1)}, e_i)$, $i = 1, \ldots, n$.

The model takes on a temporal semantics when we remove the intermediate nodes e_i, yielding the Bayesian network of Fig. 8(b). In particular, assume that the causes are initially set to their distinguished state at time $t = 0$. In addition, assume that, at time $t = i$, cause c_i is *activated*—that is, allowed to vary from its distinguished state—and subsequently remains at this new state. Then, we can interpret Fig. 8(b) by associating node c_i with the ith cause after activation, and node e_{ti} with the effect at time $t = i$.

Under this interpretation, the effect at time $t = i$ is independent of previously activated causes, given the effect at time $t = i - 1$ and the ith cause after activation. In practice, experts often find it easy to confirm or deny conditional-independence assertions of this form. Also, the definitions of e_i and e_{ti} impose a constraint on each function g_i. Namely, $e_t = g_i(e_{t(i-1)}, e_0)$—that is, e_0 is the left identity of g_i, $i = 1, \ldots, n$.

In closing our discussion of the various forms of causal independence, we stress that the preferred form will depend on the specific causes and effects being modeled as well as the expert providing the model. In an application involving the effect of drugs on white blood cell counts, we found the temporal version of causal independence to be a more natural method for interacting with the expert [21]. In contrast, in a number of hardware troubleshooting applications [28], we found the amechanistic form to be more effective.

TABLE I
THE EFFECT OF DECOMPOSITION ON CLIQUE SIZE

	Clique Size Without Decomposition		Clique Size With Decomposition	
Network	Largest	Sum	Largest	Sum
Medical	8192	15068	1536	4966
Hardware	32	176	32	196
BN2(binary)	32	128	8	160
BN2(5)	3125	12500	125	1250

IV. INFERENCE IMPROVEMENT IN BAYESIAN NETWORKS

In the previous section, we saw that there are two potential sources for gains in inference efficiency: (1) reduction in the size of parent sets afforded by (singly) decomposable causal independence, and (2) rearrangement of decompositions afforded by multiply decomposable causal independence. Although the speedups are clear for the simple Bayesian networks that we have considered, the gains are not so transparent for more general Bayesian networks.

To better understand the general case, we performed several experiments, measuring increases in inference efficiency for several artificial and real-world Bayesian networks: Medical, Hardware, BN2(binary), and BN2(5). In each model, we used the noisy-MAX model to encode all parent–child relationships. The Medical network is a 32-node Bayesian network for medical diagnosis. Nodes have two or three states; and the node with the most parents has 11 parents. The Hardware network is a 27-node Bayesian network for hardware diagnosis. The network has very few undirected cycles and mostly binary nodes; and there are at most three causes for each effect. The BN2 networks are artificial networks consisting of ten causes and four effects. Each effect has four causes, and two of the causes are common causes of each effect. Each node in the BN2(binary) and BN2(5) models have two and five states, respectively.

In our experiments, we used Jensen's junction-tree inference algorithm [15], an adaptation of Lauritzen-Spiegelhalter's algorithm [14]. In using this algorithm, we transform a given Bayesian network to an annotated undirected tree, where each node in the tree—sometimes called a *clique*—corresponds to a set of nodes in the original Bayesian network. Associated with each clique is its joint probability distribution. The run time of the algorithm is roughly proportional to the sum of the clique sizes; and we use this sum as a surrogate for run time.

Table I shows the benefits of parent-size reduction due to single decomposition with decomposition orderings chosen at random. We see that decomposition produces a factor-of-three reduction in the sum of cliques sizes in the Medical network. Most of this improvement can be traced to the node with 11 parents. Without decomposition, this node-parent set produces a clique of size 8192. Whereas, with decomposition, this node-parent set produces cliques, the largest of which has size 1536. Decomposition actually worsens performance in the Hardware network. In particular, decomposition increases the number of cliques, but does little to reduce the size of cliques, because most nodes are binary and each node has at most three parents. Comparisons of BN2(binary) and BN2(5) show that decomposition

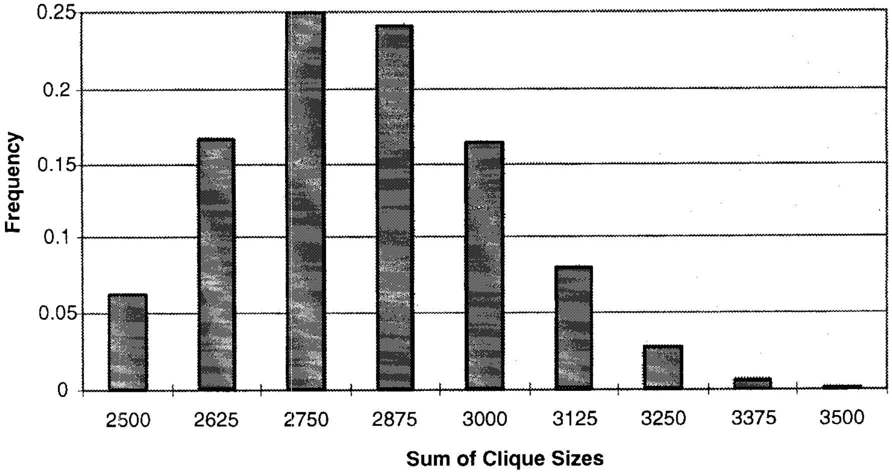


Fig. 9. Distribution of sums of clique sizes for the BN2(5) network.

becomes more effective as the number of node states is increased. Overall, these results indicate that use of decomposition can decrease inference complexity substantially when nodes have many states and many parents.

To measure improvements due to multiple decompositions, we determined the sum of clique sizes over many random orderings for the BN2(5) Bayesian network. The distribution of sums, shown in Fig. 9, indicates that gains are relatively modest. In particular, the sum of clique sizes associated with the best expansion is only slightly smaller than that of the average expansion. We have obtained similar results for other Bayesian networks.

References

[1] R. A. Howard and J. E. Matheson, "Influence diagrams," in *Readings on the Principles and Applications of Decision Analysis*, R. A. Howard and J. E. Matheson, Eds. Menlo Park, CA: Strategic Decisions Group, 1981, vol. II, pp. 721–762.

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

[3] D. Heckerman, A. Mamdani, and M. Wellman, "Real-world applications of Bayesian networks," *Commun. ACM*, vol. 38, 1995.

[4] D. Spiegelhalter and S. Lauritzen, "Sequential updating of conditional probabilities on directed graphical structures," *Networks*, vol. 20, pp. 579–605, 1990.

[5] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*. New York: Springer-Verlag, 1993.

[6] J. Pearl, "Causal diagrams for empirical research," *Biometrika*, vol. 82, pp. 669–710, 1995.

[7] D. Heckerman and R. Shachter, "A decision-based view of causality," *J. Artif. Intell. Res.*, vol. 3, pp. 405–430, 1995.

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

[9] D. Spiegelhalter, A. Dawid, S. Lauritzen, and R. Cowell, "Bayesian analysis in expert systems," *Statist. Sci.*, vol. 8, pp. 219–282, 1993.

[10] W. Buntine, "Operations for learning with graphical models," *J. Artif. Intell. Res.*, vol. 2, pp. 159–225, 1994.

[11] D. Madigan and A. Raftery, "Model selection and accounting for model uncertainty in graphical models using Occam's window," *J. Amer. Statist. Assoc.*, vol. 89, pp. 1535–1546, 1994.

[12] D. Heckerman, D. Geiger, and D. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," *Machine Learning*, vol. 20, pp. 197–243, 1995.

[13] R. D. Shachter, "Probabilistic inference and influence diagrams," *Oper. Res.*, vol. 36, pp. 589–604, 1988.

[14] S. L. Lauritzen and D. J. Spiegelhalter, "Local computations with probabilities on graphical structures and their application to expert systems," *J. Royal Statist. Soc. B*, vol. 50, pp. 157–224, 1988.

[15] F. V. Jensen, S. L. Lauritzen, and K. G. Olesen, "Bayesian updating in recursive graphical models by local computations," *Computat. Statist. Quart.*, vol. 4, pp. 269–282, 1990.

[16] B. D'Ambrosio, "Local expression languages for probabilistic dependence," in *Proc. 7th Conf. on Uncertainty in Artificial Intelligence*, Los Angeles, CA, July 1991, pp. 95–102.

[17] J. H. Kim and J. Pearl, "A computational model for causal and diagnostic reasoning in inference engines," in *Proc. 8th Int. Joint Conf. on Artificial Intelligence*, Karlsruhe, West Germany, 1983, pp. 190–193.

[18] M. Henrion, "Some practical issues in constructing belief networks," in *Proc. 3rd Workshop on Uncertainty in Artificial Intelligence*, Seattle, WA, July 1987, pp. 132–139, Assoc. for Uncertainty in Artificial Intelligence, Mountain View, CA,

[19] S. Srinivas, "A generalization of the noisy-Or model," in *Proc. 9th Conf. Uncertainty in Artificial Intelligence*, Washington, DC, July 1993, pp. 208–215.

[20] F. Diez, "Parameter adjustment in Bayes networks. The generalized noisy OR-gate," in *Proc. 9th Conf. on Uncertainty in Artificial Intelligence*, Washington, DC, July 1993, pp. 99–105.

[21] D. Heckerman, "Causal independence for knowledge acquisition and inference," in *Proc. Ninth Conf. on Uncertainty in Artificial Intelligence*, Washington, DC, Morgan Kaufmann, July 1993, pp. 122–127.

[22] D. Heckerman and J. Breese, "A new look at causal independence," in *Proc. 10th Conf. Uncertainty in Artificial Intelligence*, Seattle, WA, July 1994, pp. 286–292.

[23] R. Neal, "Connectionist learning of belief networks," *Artif. Intell.*, vol. 56, pp. 71–113, 1992.

[24] R. D. Shachter, "Evaluating influence diagrams," *Oper. Res.*, vol. 34, pp. 871–882, 1986.

[25] D. Lewis, "Causation," *J. Philosophy*, pp. 556–572, 1973.

[26] P. Holland, "Statistics and causal inference," *J. Amer. Statist. Assoc.*, vol. 81, pp. 945–968, 1986.

[27] D. Rubin, "Bayesian inference for causal effects: The role of randomization," *Ann. Statist.*, vol. 6, pp. 34–58, 1978.

[28] D. Heckerman, J. Breese, and K. Rommelse, "Decision-theoretic troubleshooting," *Commun. ACM*, vol. 38, pp. 49–57, 1995.