JOURNAL OF COMPLEXITY 2, 60–77 (1986)

Structuring Causal Trees*

JUDEA PEARL

Computer Science Department, University of California, Los Angeles, California 90024

AND

MICHAEL TARSI

Computer Science Department, Tel Aviv University, Ramat Aviv, Israel

Models of complex phenomena often consist of hypothetical entities called "hidden causes," which cannot be observed directly and yet play a major role in understanding those phenomena. This paper examines the computational roles of these constructs, and addresses the question of whether they can be discovered from empirical observations. Causal models are treated as trees of binary random variables where the leaves are accessible to direct observation, and the internal nodes—representing hidden causes—account for interleaf dependencies. In probabilistic terms, every two leaves are conditionally independent given the value of some internal node between them. We show that if the mechanism which drives the visible variables is indeed tree structured, then it is possible to uncover the topology of the tree uniquely by observing pairwise dependencies among the leaves. The entire tree structure, including the strengths of all internal relationships, can be reconstructed in time proportional to $n \log n$, where n is the number of leaves.

1. INTRODUCTION: CAUSALITY, CONDITIONAL INDEPENDENCE, AND TREE STRUCTURES

This study is motivated by the observation that human beings, facing complex phenomena, exhibit an almost obsesssive urge to conceptually mold these phenomena into structures of cause-and-effect relationships. This tendency is, in fact, so compulsive that it sometimes comes at the expense of precision and often requires the invention of hypothetical, unobservable entities such as "ego," "elementary particles," and "supreme beings" to make theories fit the mold of causal schema. When we try to explain the actions of another person, for example, we invariably invoke abstract notions of mental

*This work was supported in part by National Foundation Grant IST-81-19405.

0743-7315/86 $3.00

states, social attitudes, beliefs, goals, plans, and intentions. Medical knowledge, likewise, is organized into causal hierarchies of invading organisms, physical disorders, complications, pathological states, and only finally, the visible symptoms.

A first step toward mechanizing the process of learning causal models would be to give causality an operational definition that will permit an algorithm to discover it from empirical data. This paper takes the position that human obsession with causation is computationally motivated. Causal models are attractive only because they provide effective data structures for representing empirical knowledge, and their effectiveness is a result of the high degree of decomposition they induce. More specifically, causes are viewed as names given to auxiliary variables which summarize interactions between the visible variables and, once calculated, would permit us to treat visible variables as if they were mutually independent.

If you ask n persons in the street what time it is, the answers will undoubtedly be very similar. Yet instead of suggesting that somehow the answers evoked or the persons surveyed influence each other, we postulate the existence of a central cause, the standard time, and the commitment of each person to adhere to that standard. Thus, instead of a complex n-ary relation, the causal model in this example consists of a network of n binary relations, all connected star-like to one central node which serves to dispatch information to and from the connecting variables. Psychologically, this architecture is much more pleasing. Since the activity of each variable is constrained by only one source of information (i.e., the central cause), no conflict in activity arises: any assignment of values consistent with the central constraints will also be globally consistent, and moreover, a change in any of the variables can communicate its impact to all other variables in only two steps.

In probabilistic formalisms, this decomposition is embodied by the notion of *conditional independence*. In our preceding example, the answers to the question "What time is it?" would be viewed as random variables that are bound together by a *spurious correlation* (Simon, 1954; Suppes, 1970) and become independent of each other once we know the state of the mechanism causing the correlation, i.e., the standard time.

The most familiar connection between causality and conditional independence is reflected in the notion of a *state*. It was devised to break up the influence that the past exerts on the future by providing a sufficiently detailed description of the present, and came to be known as a Markov property—future events are conditionally independent of past events, given the current state of affairs.

Conditional independence, however, is not limited to separating the past from the future but is often induced on events which occur at the same time. A distinctive characteristic of causality is that it generally gives rise to independent outcomes; i.e., knowing the cause C of an outcome X renders X independent of other possible consequences of C. In medical diagnosis, for

example, a group of co-occurring symptoms often become independent of each other once we know the disease that caused them. When some of the symptoms directly influence each other, the medical profession *invents* a name for that interaction (e.g., complication, clinical state, etc.) and treats it as a new auxiliary variable that decouples others; knowing the exact state of the auxiliary variable renders the interacting symptoms independent of each other.

On the basis of these observations we chose to represent causal models as trees of binary random variables, where the leaves are directly accessible to empirical observations and the internal nodes represent hidden causes; any two leaves become conditionally independent once we know the value of some internal variable on the path connecting them. The propagation of updated probabilities in such trees was analyzed by Pearl (1982) and Kim and Pearl (1983). It was shown that the propagation can be accomplished by a network of parallel processors working autonomously, and that the impact of new information can be imparted to all variables in time proportional to the longest path in the tree. These computational advantages, we conjecture, may account for the satisfying sensation called "in-depth understanding" that people experience upon discovering causal models consistent with observations.

Given that tree dependence captures the main feature of causation and that it provides a convenient computational medium for performing updating and predictions, we now ask whether it is possible to configure every set of random variables as a tree and, if so, how. Our first task would be to assume that there exist dummy variables which decompose the set into a tree, and then ask whether the internal structure of such a tree can be determined from observations made solely on the leaves. If it can, then the structure found will constitute an operational definition for the hidden causes often found in causal models. Additionally, if we take the view that "learning" entails the acquisition of computationally effective representations of nature's regularities, then procedures for configuring causal trees may reflect an important component of human learning.

A related structuring task was treated by Chow and Liu (1968), who also used tree-dependent random variables to approximate an arbitrary joint distribution. However, in Chow's trees all nodes denote observed variables, so the conditional probabilities for any pair of variables are assumed given. By contrast, the internal nodes in our trees denote dummy variables, artificially concocted to make the representation tree-like. Only the leaves are accessible to empirical observations; namely, we do not know any of the conditional probabilities that link the internal nodes to the leaves, nor the structure of the tree—these would have to be learned. A similar problem of configuring probabilistic models with hidden variables is mentioned by Hinton *et al.* (1984) as one of the tasks that a Boltzmann machine should be able to solve. However, it is not clear whether the relaxation techniques employed by the Boltzmann machine can readily accept the restriction that the resulting struc-

ture be a tree. The method described in the following sections offers a solution to this problem, but it assumes some restrictive conditions: all variables are bivalued, a solution tree is assumed to exist, and all interleaf correlations are known precisely.

The paper is organized as follows: Section 2 presents nomenclature and precise definitions for the notions of star decomposability and tree decomposability. In Section 3 we treat triplets of random variables and ask under what conditions one is justified in attributing the observed dependencies to one central cause. We show that these conditions are readily testable and, when the conditions are satisfied, that the parameters specifying the relations between the visible variables and the central cause can be determined uniquely. In Section 4 we extend these results to the case of a tree with n leaves. We show that if a joint distribution of n variables has a tree-dependent representation, then the uniqueness of the triplets' decomposition enables us to configure that tree from pairwise dependencies among the variables. Moreover, the configuration procedure takes only $O(n \log n)$ steps. In Section 5 we evaluate the merits of this method and address the difficult issues of estimation and approximations.

2. Problem Definition and Nomenclature

Consider a set of n binary-valued random variables $x_1, \ldots, x_n$ with a given probability mass function $P(x_1, \ldots, x_n)$. We address the problem of representing P as a marginal of an $(n + 1)$-variable distribution $P_S(x_1, \ldots, x_n, w)$, that renders $x_1, \ldots, x_n$ conditionally independent given w, i.e.,

$$P_s(x_1, \ldots, x_n, w) = \prod_{i=1}^{n} P_s(x_i \mid w) P_s(w) \tag{1}$$

$$P(x_1, \ldots, x_n) = \alpha \prod_{i=1}^{n} P_s(x_i \mid w = 1) + (1 - \alpha) \prod_{i=1}^{n} P_s(x_i \mid w = 0). \tag{2}$$

The functions $P_S(x_i \mid w)$, $w = 0, 1$, $i = 1, \ldots, n$, can be viewed as 2×2 stochastic matrices relating each x_i to the central hidden variable w (see Fig. 1a); hence we name P_S a *star distribution* and call P *star decomposable*. Each matrix contains two independent parameters, f_i and g_i, where

$$\begin{aligned} f_i &= P_s(x_i = 1 \mid w = 1) \\ g_i &= P_s(x_i = 1 \mid w = 0) \end{aligned} \tag{3}$$

and the central variable w is characterized by its prior probability $P_s(w = 1) = \alpha$ (see Fig. 1b).

The advantages of having star-decomposable distributions are several.

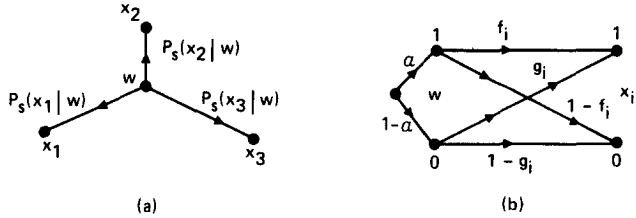


FIG. 1. (a) Three random variables, x_1, x_2, x_3 connected to a central variable w by a star network. (b) Illustrating the three parameters, α, f_i, g_i, associated with each link.

First, the product form of P_s in (1) makes it extremely easy to compute the probability of any combination of variables. More importantly, it is also convenient for calculating the conditional probabilities $P(x_i \mid x_j)$, describing the impact of an observation x_j on the probabilities of unobserved variables. The computation requires only two vector multiplications.

Unfortunately, when the number of variables exceeds 3, the conditions for star decomposability become very stringent, and are not likely to be met in practice. Indeed, a star-decomposable distribution for n variables has $2n + 1$ independent parameters, while the specification of a general distribution requires $2^n - 1$ parameters. Lazarfeld (1966) considered star-decomposable distributions where the hidden variable w is permitted to range over λ values, $\lambda > 2$. Such an extension requires the solution of $\lambda n + \lambda - 1$ nonlinear equations to find the values of its $\lambda n + \lambda - 1$ independent parameters. In this paper, we pursue a different approach, allowing a larger number of binary hidden variables, but insisting that they form a tree-like structure (see Fig. 2); i.e., each triplet forms a star but the central variables may differ from triplet to triplet. Trees often portray meaningful conceptual hierarchies and are computationally almost as convenient as stars.

We shall say that a distribution $P(x_1, x_2, \ldots, x_n)$ is *tree decomposable* if it is the marginal of a distribution

$$P_T(x_1, x_2, \ldots, x_n, w_1, w_2, \ldots, w_m), \qquad m \leq n - 2,$$

where $w_1, w_2, \ldots, w_m$ correspond to the internal nodes of a tree T, x_1, $x_2, \ldots, x_n$, to its leaves and any two leaves are conditionally independent given the value of any internal node on the path connecting them.

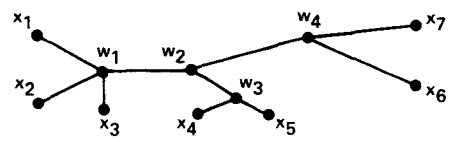


FIG. 2. A tree containing four dummy variables and seven visible variables.

Given an unrooted tree T and an assignment of variables to its nodes, the form of the corresponding distribution can be written by the following procedure. We first choose an arbitrary node as a root. This, in turn, defines a unique father $F(y_i)$ for each node $y_i \in \{x_1, \ldots, x_n, w_1, \ldots, w_m\}$ in T, except the chosen root, y_1. The joint distribution is simply given by the product form

$$P_T(x_1 \cdots x_n, w_1 \cdots w_m) = P(y_1) \prod_{i=2}^{m+n} P[y_i \mid F(y_i)]. \tag{4}$$

For example, if in Fig. 2 we choose w_2 as the root we obtain

$$P_T(x_1, \ldots, x_7, w_1, \ldots, w_4) = P(x_7 \mid w_4)P(x_6 \mid w_4)P(x_5 \mid w_3)P(x_4 \mid w_3) \times P(x_3 \mid w_1)P(x_2 \mid w_1)P(x_1 \mid w_1)P(w_1 \mid w_2)P(w_3 \mid w_2)P(w_4 \mid w_2)P(w_2).$$

Throughout this discussion we shall assume that each w has at least three neighbors; otherwise it is superfluous. In other words, an internal node with two neighbors can simply be replaced by an equivalent direct link between the two.

If we are given $P_T(x_1, \ldots, x_n, w_1, \ldots, w_m)$ then, clearly, we can obtain $P(x_1, \ldots, x_n)$ by summing over the w's. We now ask whether the inverse transformation is possible; i.e., given a tree-decomposable distribution $P(x_1, \ldots, x_n)$, can we recover its underlying extension $P_T(x_1, \ldots, x_n, w_1, \ldots, w_m)$? We shall show that (1) the tree distribution P_T is unique, (2) it can be recovered from P using $n \log n$ computations, and (3) the structure of T is uniquely determined by the second-order probabilities of P. The construction method depends on the analysis of star decomposability for triplets which is presented next.

3. Star-Decomposable Triplets

In order to test whether a given three-variable distribution $P(x_1, x_2, x_3)$ is star decomposable, we first solve Eq. (2) and express the parameters α, f_i, g_i as functions of the parameters specifying P. This task was carried out by Lazarfeld (1966) in terms of the seven joint-occurrence probabilities

$$\begin{aligned} p_i &= P(x_i = 1) \\ p_{ij} &= P(x_i = 1, x_j = 1) \\ p_{ijk} &= P(x_i = 1, x_j = 1, x_k = 1), \end{aligned} \tag{5}$$

yielding the following solution:

Define the quantities

$$[ij] = p_{ij} - p_i p_j \tag{6}$$

$$S_i = \left(\frac{[ij][ik]}{[jk]}\right)^{1/2} \tag{7}$$

$$\mu_i = \frac{p_i p_{ijk} - p_{ij} p_{ik}}{[jk]} \tag{8}$$

$$K = \frac{S_i}{p_i} - \frac{p_i}{S_i} + \frac{\mu_i}{S_i p_i} \tag{9}$$

and let t be the solution of

$$t^2 + Kt - 1 = 0. \tag{10}$$

The parameters α, f_i, g_i are given by

$$\alpha = \frac{t^2}{1 + t^2} \tag{11}$$

$$f_i = p_i + S_i\left(\frac{1 - \alpha}{\alpha}\right)^{1/2} \tag{12}$$

$$g_i = p_i - S_i\left(\frac{\alpha}{1 - \alpha}\right)^{1/2}. \tag{13}$$

Moreover, the differences $f_i - g_i$ are independent of p_{ijk},

$$f_i - g_i = S_i = \left(\frac{[ij][ik]}{[jk]}\right)^{1/2}. \tag{14}$$

The conditions for star decomposability are obtained by requiring that the preceding solutions satisfy:

(a) S_i is real,

(b) $0 \leq f_i \leq 1$,

(c) $0 \leq g_i \leq 1$.

Using the variances

$$\sigma_i = [p_i(1 - p_i)]^{1/2} \tag{15}$$

and the correlation coefficients

$$\rho_{ij} = \frac{p_{ij} - p_i p_j}{\sigma_i \sigma_j}, \tag{16}$$

requirement (a) is equivalent to the condition that all three correlation coefficients are nonnegative. (If two of them are negative, we can rename two variables by their complements; the newly defined triplet will have all its pairs positively correlated.) We shall call triplets with this property *positively correlated.*

This, together with requirements (b) and (c), gives (see Appendix I):

THEOREM 1. *A necessary and sufficient condition for three dichotomous random variables to be star decomposable is that they are positively correlated, and that the inequality*

$$\frac{p_{ijk}p_{ij}}{p_i} \leq p_{ijk} \leq \frac{p_{ik}p_{ij}}{p_i} + \sigma_j\sigma_k(\rho_{jk} - \rho_{ij}\rho_{ik}) \tag{17}$$

is satisfied for all $i \in \{1, 2, 3\}$. When this condition is satisfied, the parameters of the star-decomposed distribution can be determined uniquely, up to a complementation of the hidden variable w, i.e., $w \to (1 - w)$, $f_i \to g_i$, $\alpha \to (1 - \alpha)$.

Obviously, in order to satisfy (17), the term $(\rho_{jk} - \rho_{ij}\rho_{ik})$ must be nonnegative. This introduces a simple necessary condition for star decomposability that may be used to quickly rule out many likely candidates.

COROLLARY *A necessary condition for a distribution $P(x_1, x_2, x_3)$ to be star decomposable is that all correlation coefficients obey the triangle inequality*

$$\rho_{jk} \geq \rho_{ji}\rho_{ik}. \tag{18}$$

Equation (18) is satisfied with equality if w coincides with x_i, i.e., when x_j and x_k are independent given x_i. Thus, an intuitive interpretation of this corollary is that the correlation between any two variables must be stronger than that induced by their dependencies on the third variable; a mechanism accounting for direct dependencies must be present.

Having established the criterion for star decomposability, we may address a related problem: Suppose P is not star decomposable; can it be approximated by a star-decomposable distribution $\hat{P}$ that has the same second-order probabilities?

The preceding analysis contains the answer to this question. Note that the third-order statistics are represented only by the term p_{ijk}, and this term is confined by Eq. (7) to a region whose boundaries are determined by second-order parameters. Thus, if we insist on keeping all second-order dependencies of P intact and are willing to choose p_{ijk} so as to yield a star-decomposable distribution, we can only do so if the region circumscribed by (7) is nonempty. This leads to the statement:

THEOREM 2. *A necessary and sufficient condition for the second-order dependencies among the triplet x_1, x_2, x_3 to support a star-decomposable extension is that the six inequalities*

$$\frac{p_{ij}p_{ik}}{p_i} \leq x \leq \frac{p_{ij}p_{ik}}{p_i} + \sigma_j\sigma_k(\rho_{jk} - \rho_{ij}\rho_{ik}), \qquad i = 1, 2, 3 \tag{19}$$

possess a solution for x.

4. A TREE-RECONSTRUCTION PROCEDURE

We are now ready to confront the central problem of this paper: Given a tree-decomposable distribution $P(x_1, \ldots, x_n)$, can we uncover its underlying topology and the underlying tree distribution $P_T(x_1, \ldots, x_n, w_1, \ldots, w_m)$?

The construction method is based on the observation that any three leaves in a tree have one and only one internal node that can be considered their *center;* i.e., it lies on all the paths connecting the leaves to each other. If one removes the center, the three leaves become disconnected from each other. This means that if P is tree decomposable then the joint distribution of any triplet of variables x_i, x_j, x_k is star decomposable; i.e., $P(x_i, x_j, x_k)$ uniquely determines the parameters α, f_i, g_i as in Eq. (11), (12), and (13), where α is the marginal probability of the central variable. Moreover, if we compute the star decompositions of two triplets of leaves, both having the same central node w, the two distributions should have the same value for $\alpha = P_T(w = 1)$. This provides us with a basic test for verifying whether two arbitrary triplets of leaves share a common center; a successive application of this test is sufficient for determining the structure of the entire tree.

Consider a 4-tuple x_1, x_2, x_3, x_4 of leaves in T. These leaves are interconnected through one of the four possible topologies shown in Fig. 3. The topologies differ in the identity of the triplets which share a common center. For example, in the topology of Fig. 3a, the pair [(1, 2, 3), (1, 2, 4)] shares a common center and so does the pair [(1, 3, 4), (2, 3, 4)]. In Fig. 3b, on the other hand, the sharing pairs are [(1, 2, 4,) (2, 4, 3)] and [(1, 3, 4), (2, 1, 3)], and in Fig. 3d all triplets share the same center. Thus, the basic test for

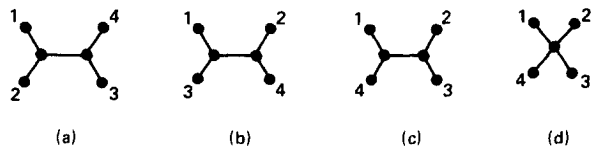


FIG. 3. The four possible topologies by which four leaves can be related.

center-sharing triplets enables us to decide the topology of any 4-tuple and, eventually, to configure the entire tree.

We start with any three variables x_1, x_2, and x_3, form their star decomposition, choose a fourth variable x_4, and ask to which leg of the star x_4 should be joined. We can answer this question easily by testing which pairs of triplets share centers, deciding on the appropriate topology, and connecting x_4 accordingly. Similarly, if we already have a tree structure T_i, with i leaves, and wish to know where to join the $(i + 1)$th leaf, we can choose any triplet of leaves from T_i with central variable w, and test to which leg of w the variable x_{i+1} should be joined. This, in turn, identifies a subtree T_i' of T_i that should receive x_{i+1} and permits us to remove from further consideration the subtrees emanating from the unselected legs of w. Repeating this operation on the selected subtree T_i' will eventually reduce it to a single branch, to which x_{i+1} is joined.

Appendix II describes the construction procedure in algorithmic detail and shows that by choosing, in each state, a central variable that splits the available tree into subtrees of roughly equal size, the joining branch of x_{i+1} can be identified in at most $\log_{k/(k-1)}(i)$ tests, where k is the maximal degree of the tree T_i. This amounts to $O(n \log n)$ tests for constructing an entire tree of n leaves.

So far we have shown that the structure of the tree T can be uncovered uniquely. Next we show that the distribution P_T, likewise, is uniquely determined from P, i.e., that we can determine all the functions $P(x_i \mid w_j)$ and $P(w_j \mid w_k)$ in (4), for $i = 1, \ldots, n$ and $j, k = 1, 2, \ldots, m$. The functions $P(x_i \mid w_j)$ assigned to the peripheral branches of the tree are determined directly from the star decomposition of triplets involving adjacent leaves. In Fig. 2, for example, the star decomposition of $P(x_1, x_2, x_5)$ yields $P(x_1 \mid w_1)$ and $P(x_2 \mid w_1)$. The conditional probabilities $P(w_i \mid w_k)$ assigned to interior branches are determined by solving matrix equations. For example, $P(x_1 \mid w_2)$ is obtained from the star decomposition of (x_1, x_5, x_7), and it is related to $P(x_1 \mid w_1)$ via

$$P(x_1 \mid w_2) = \sum_{w_1} P(x_1 \mid w_1)P(w_1 \mid w_2).$$

This matrix equation has a solution for $P(w_1 \mid w_2)$ because $P(x_1 \mid w_1)$ must be nonsingular. It is only singular when $f_1 = g_1$, i.e., when x_1 is independent of w_1 and is therefore independent of all other variables. Hence, we can determine the parameters of the branches next to the periphery, use those to determine more interior branches, and so on, until all the interior conditional probabilities $P(w_i \mid w_j)$ are determined.

Next, we shall show that the tree structure can be recovered without resorting to third-order probabilities; correlations among pairs of leaves suffice. This feature stems from the observation that when two triplets of a 4-tuple are star decomposable with respect to the same central variable w

(e.g., 1, 2, 3 and 1, 2, 4 in Fig. 3a), then not only are the values of α the same, but the f and g parameters associated with the two common variables (e.g., 1 and 2 in Fig. 3a) must also be the same. Whereas the value of α depends on a third-order probability, the difference $f_i - g_i$ depends only on second-order terms via Eq. (14). Thus, requiring that $f_1 - g_1$ in Fig. 3a obtain the same value in the star decomposition of (1, 2, 3) as in that of (1, 2, 4) leads to the equation

$$\frac{[12][13]}{[23]} = \frac{[12][14]}{[24]}, \tag{20}$$

and, using (6), this yields

$$\rho_{13}\rho_{42} = \rho_{14}\rho_{32}. \tag{21}$$

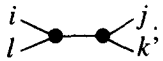
An identical equality will be obtained for each $f_i - g_i$, $i = 1, 2, 3, 4$, relative to the topology of Fig. 3a. Similarly, the topology of Fig. 3b dictates

$$\rho_{12}\rho_{43} = \rho_{14}\rho_{23}, \tag{22}$$

and that of Fig. 3c,

$$\rho_{12}\rho_{34} = \rho_{13}\rho_{24}. \tag{23}$$

Thus, we see that each of these three topologies is characterized by its own distinct equality, while the topology of Fig. 3d is characterized by having all three equalities hold simultaneously. This provides the necessary second-order criterion for deciding the topology of any 4-tuple tested: if the equality $\rho_{ij}\rho_{kl} = \rho_{ik}\rho_{jl}$ holds for some permutation of the indices, we decide on the topology



if it holds for two such permutations, the entire 4-tuple is star decomposable. Note that the equality $\rho_{ij}\rho_{kl} = \rho_{ik}\rho_{jl}$ must hold for at least one permutation of the variables, or else the 4-tuple would not be tree decomposable.

5. Conclusions and Open Questions

This paper provides an operational definition for entities called "hidden causes," which are not directly observable but facilitate the acquisition of effective causal models from empirical data. Hidden causes are viewed as dummy variables which, if held constant, induce probabilistic independence

among sets of visible variables. It is shown that if all variables are bivalued and if the activities of the visible variables are governed by a tree-decomposable probability distribution, then the topology of the tree can be uncovered uniquely from the observed correlations between pairs of variables. Moreover, the structuring algorithm requires only n log n steps.

The method introduced in this paper has two major shortcomings: It requires precise knowledge of the correlation coefficients, and it only works when there exists an underlying model that is tree structured. In practice, we often have only sample estimates of the correlation coefficients, and it is therefore unlikely that criteria based on equalities (as in Eq. (21)) will ever be satisfied exactly. It is possible, of course, to relax these criteria and make topological decisions by seeking proximities rather than equalities. For example, instead of searching for an equality $\rho_{ij}\rho_{kl} = \rho_{ik}\rho_{jl}$, we can decide the 4-tuple topology on the basis of the permutation of indices that minimizes the difference $\rho_{ij}\rho_{kl} - \rho_{ik}\rho_{jl}$. Experiments show, however, that the structure which evolves by such a method is very sensitive to inaccuracies in the estimates ρ_{ij}, because no mechanism is provided to retract erroneous decisions made in the early stages of the structuring process. Ideally, the topological membership of the $(i + 1)$th leaf should be decided not merely by its relations to a single triplet of leaves chosen to represent an internal node w, but also by its relations to all previously structured triplets which share w as a center. This, of course, will substantially increase the complexity of the algorithm.

Similar difficulties plague the task of finding the best tree-structured *approximation* to a distribution which is not tree decomposable. Even though we argued that natural data which lend themselves to causal modeling should be representable as tree-decomposable distributions, these distributions may contain internal nodes with more than two values. The task of determining the parameters associated with such nodes is much more complicated and, in addition, rarely yields unique solutions. Unique solutions, as shown in Section 4, are essential for building large structures from smaller ones. We leave open the question of explaining how approximate causal modeling, an activity which humans seem to perform with relative ease, can be embodied in computational procedures that are both sound and efficient.

APPENDIX I: CONDITIONS FOR STAR DECOMPOSABILITY

Let

$$\begin{aligned} p_i &= P(x_i = 1) \\ p_{ij} &= P(x_i = 1, x_j = 1) \\ p_{ijk} &= P(x_i = 1, x_j = 1, x_k = 1). \end{aligned} \tag{I1}$$

The seven joint-occurrence probabilities, p_1, p_2, p_3, p_{12}, p_{13}, p_{23}, p_{123}, uniquely define the seven parameters necessary for specifying $P(x_1, x_2, x_3)$; for example,

$$P(x_1 = 1, x_2 = 1, x_3 = 0) = p_{12} - p_{123}$$
$$P(x_1 = 1, x_2 = 0) = p_1 - p_{12}, \qquad \text{etc.},$$

and will be used in the following analysis.

Assuming P is star decomposable (Eq. (1)), we can express the joint-occurrence probabilities in terms of α, f_i, g_i and obtain seven equations for these seven parameters.

$$p_i = \alpha f_i + (1 - \alpha) g_i \tag{I2}$$
$$p_{ij} = \alpha f_i f_j + (1 - \alpha) g_i g_j \tag{I3}$$
$$p_{ijk} = \alpha f_i f_j f_k + (1 - \alpha) g_i g_j g_k. \tag{I4}$$

These equations can be manipulated to yield product forms on the right-hand sides:

$$p_{ij} - p_i p_j = \alpha(1 - \alpha)(f_i - g_i)(f_j - g_j) \tag{I5}$$
$$p_i p_{ijk} - p_{ij} p_{ik} = \alpha(1 - \alpha) f_i g_i (f_j - g_j)(f_k - f_k). \tag{I6}$$

Equation (I5) comprises three equations which can be solved for the differences $f_i - g_i$, $i = 1, 2, 3$, giving

$$f_i - g_i = S_i = \pm\left(\frac{[ij][ik]}{[jk]}\right)^{1/2}, \tag{I7}$$

where the bracket $[ij]$ stands for the determinant

$$[ij] = p_{ij} - p_i p_j. \tag{I8}$$

These, together with (I2), determine f_i and g_i in terms of S_i and α (still unknown):

$$f_i = p_i + S_i\left(\frac{1 - \alpha}{\alpha}\right)^{1/2} \tag{I9}$$

$$g_i = p_i - S_i\left(\frac{\alpha}{1 - \alpha}\right)^{1/2} \tag{I10}$$

To determine α, we invoke Eq. (I6) and obtain

$$\left(\frac{\alpha}{1-\alpha}\right)^{1/2} = t \qquad \left(\text{or, } \alpha = \frac{t^2}{1+t^2}\right), \tag{I11}$$

where t is the solution to

$$t^2 + Kt - 1 = 0 \tag{I12}$$

and K is defined by

$$K = \frac{S_i}{p_i} - \frac{p_i}{S_i} + \frac{\mu_i}{S_i p_i} \tag{I13}$$

$$\mu_i = \frac{[jk, i]}{[jk]} = \frac{p_i p_{ijk} - p_{ij} p_{ik}}{[jk]}. \tag{I14}$$

It can be easily verified that K (and, therefore, α) obtains the same value regardless of which index i provides the parameters in (I13).

From Eq. (I13) we see that the parameters S_i and μ_i of P govern the solutions of (I12) which, in turn, determine whether P is star decomposable via the resulting values of α, f_i, g_i. These conditions are obtained by requiring that

(a) S_i is real,
(b) $0 \leq f_i \leq 1$,
(c) $0 \leq g_i \leq 1$.

Requirement (a) implies that, of the three brackets in (I7), either all three are nonnegative or exactly two are negative. These brackets are directly related to the correlation coefficient, via

$$p_{ij} = [ij][p_i(1-p_i)]^{-1/2}[p_j(1-p_j)]^{-1/2} = \frac{[ij]}{\sigma_i \sigma_j}, \tag{I15}$$

and so requirement (a) is equivalent to the condition that all three correlation coefficients be nonnegative. If two of them are negative, we can rename two variables by their complements; the newly defined triplet will have all its pairs positively correlated.

Now attend to requirement (b). Equation (I9) shows that f_i can be negative only if S_i is negative, i.e., if S_i is identified with the negative square root in (I7). However, the choice of negative S_i yields a solution (f_i', g_i', α') which is symmetrical to that stemming from a positive $S_i(f_i, g_i, \alpha)$, with $f_i' = g_i$, $g_i' = f_i$, $\alpha' = 1 - \alpha$. Thus, S_i and f_i can be assumed nonnegative, and it remains to examine the condition $f_i \leq 1$ or, equivalently, $t \geq S_i/(1 - p_i)$ (see (I9)) and (I11)). Imposing this condition in (I12) translates to

$$p_{ijk} \leq \frac{p_{ij}p_{ik}}{p_i} + \sigma_k\sigma_j[\rho_{jk} - \rho_{ij}\rho_{ik}]. \tag{I16}$$

Similarly, inserting requirement (c), $g_i \geq 0$, in Eq. (I12) yields the inequality

$$\frac{p_{ik}p_{ij}}{p_i} \leq p_{ijk}, \tag{I17}$$

which, together with (I16), leads to Theorem 1, Section 3.

APPENDIX II: DESCRIPTION AND ANALYSIS OF THE TREE-CONSTRUCTION ALGORITHM

Our primary task is to reconstruct an unrooted tree by a sequence of tests performed on its leaves. In each test we select a group of four leaves and identify the pairs that are connected by disjoint paths. Our procedure for accomplishing this task is best described in two stages. First we treat the construction of rooted trees where the tests are performed on triplets of leaves. Then we show that the rooted-tree procedure can be easily adapted to handle unrooted trees as well.

II.1. *Reconstructing Rooted Trees*

Let T be a rooted tree with n leaves $x_1, x_2, \ldots, x_n$. A leaf x_i is said to be the leader of the triple (x_i, x_j, x_k) if the path from the root to x_i does not contain the deepest common ancestor of x_j and x_k. If a triple does not have a leader then the deepest common ancestor of all three leaves is also a common ancestor of any two of them.

We next present an algorithm to reconstruct a tree where the available information is the leader (if there exists any) of every triple of leaves. We will try to minimize the number of triples for which we ask who the leader is.

In order to state the algorithm we first make the following observation.

LEMMA 1. *Let k be the maximum number of sons of a node in a rooted tree T with n leaves. There exists a node v of T such that $n/(k+1) < \mathrm{des}(v) \leq nk/(k+1)$, where $\mathrm{des}(v)$ is defined to be the number of leaves which are descendants of v, and $\mathrm{des}(v) = 1$ if v is a leaf.*

Proof. Let v_0 be the root of T. Define v_{t+1} to be that son of v_t which has the largest $\mathrm{des}(\cdot)$ value among all the sons of v_t. We thus defined a sequence $v_0, v_1, \ldots, v_t, v_{t+1}, \ldots, v_m$, where the last term (v_m) is a leaf. Let v_j be the first node in the sequence $v_1, v_2, \ldots, v_m$ with $\mathrm{des}(v_j) \leq kn/(k+1)$. v_j does exist, because $\mathrm{des}(v_o) = n$ and $\mathrm{des}(v_m) = 1$. Now, from

$$\mathrm{des}(v_{j-1}) > \frac{kn}{k+1} \tag{II1}$$

we obtain

$$\text{des}(v_j) \geq \frac{\text{des}(v_{j-1})}{k} > \frac{n}{k+1} \qquad \text{(II2)}$$

as required. ■

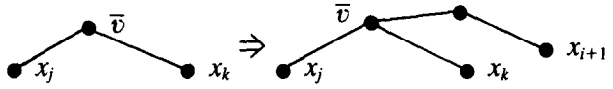
Let T be a rooted tree with leaves $x_1, x_2, \ldots, x_n$. Every node of T which is not a leaf has at least two and at most k sons. Our algorithm constructs a sequence of trees $T_2, T_3, \ldots, T_n$, where T_2 is the tree

$T_n = T$, and T_{i+1} is obtained by adding x_{i+1} as a new leaf to T_i. T_i would be the subtree of T containing only the leaves $x_1 \cdots x_i$ and the path connecting them; i.e., any nonleaf node of T which does not have any sons is removed and any node which remains with just one son is replaced by an edge joining the son directly to its father. The location where x_{i+1} should be added to T_i is found in the following "binary search-like" algorithm.

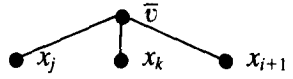
Procedure **add** (**integer** i) *Begin*

1. $T_c = T_i$ (T_c is a subtree of T_i to which x_{i+1} is to be added. It becomes progressively smaller by eliminating those sections of T_i known not to contain x_{i+1} (statements 8, 9, and 10).
2. $s :=$ the number of leaves in T_c.
3. If $s = 2$ let $\overline{v}$ be the root of T_c and x_j, x_k its two leaves.
4. If $s > 2$ select as $\overline{v}$ any node of T_c for which $s/(k+1) < \text{des}(\overline{v}) \leq sk/(k+1)$ (Lemma 1) and let x_j, x_k be two leaves whose common ancestor is $\overline{v}$.
5. Ask for the leader of the triple (x_{i+1}, x_j, x_k) (with respect to T).
6. If $s > 2$ then begin.
7. Define a partition of T_c into two subtrees: T_{c_1} rooted at $\overline{v}$ with all the descendants of $\overline{v}$ and $T_{c_2} = T_c - T_{c_1}$ in which $\overline{v}$ is considered a leaf.
8. If x_{i+1} is the leader of (x_{i+1}, x_j, x_k) then set $T_c = T_{c_2}$.
9. If there is no leader, set $T_c = T_{c_1}$ from which the two sons of $\overline{v}$ whose descendants are x_j and x_k are removed with all their descendants.
10. If x_j (or x_k) is the leader, set $T_c =$ the subtree of T_{c_1} rooted at that son of $\overline{v}$ which is the ancestor of x_k (or x_j, respectively).
11. GO TO 2 END
12. If $s = 2$ then begin.
13. If x_j (or x_k) is the leader add a new node on the edge of x_k or x_j, respectively, and make it the father of x_{i+1}.

14. If x_{i+1} is the leader add a new root and make x_{i+1} and the old root $\bar{v}$ its sons.



15. If there is no leader, make x_{i+1} a son of $\bar{v}$.



16. END END **add**

Complexity Analysis

Whenever the procedure **add** is applied to construct T_{i+1} out of T_i it starts to search a tree with i leaves. After each leadership test (statement 5) the search proceeds on a subtree which might contain at most a fraction $k/(k+1)$ of the leaves of the previous subtree. Thus, the number of steps (leadership tests) can be at most $\log i$ to the base $(k+1)/k$.

The complexity of the entire algorithm is the sum of this amount over $i = 2, 3, \ldots, n = \log_{(k+1)/k}(n!) = O(n \log n)$ for every fixed number k. However, if the degree k is not bounded, the construction of T_{i+1} out of T_i might take up to i steps, which leads to a total complexity of $\sum_{i=1}^{n} i = n(n+1)/2 = O(n^2)$. This upper bound will actually be achieved in a star-like tree, where all n leaves are sons of the root.

The number of different binary (and thus any fixed $k \geq 2$) trees on n labeled leaves can be lower bounded by $n!$ using the following construction: Take a simple path $a_1, a_2, \ldots, a_n$, make a_1 the root, and for every permutation $P = X_{i1}, X_{i2}, \ldots, X_{in}$ construct a binary tree $T(P)$ making every x_{ij} the son of a_j. This shows that spending $O(n \log n)$ tests is the best possible for this kind of problem. (Every leadership test provides one of four possible answers which amounts to two bits of information.)

The number of trees possible in the case where k is not bounded can be estimated as follows: Since no node of T has just one son, the total number of nodes in T is less than twice the number of leaves—$2n$. On $2n$ labeled nodes there exist $2n^{2n-2}$ different spanning trees; thus $2n^{2n-2}$ is an upper bound to our tree-counting problem. To identify one of these spanning trees would require at least $\log(2n^{2n-2})$ tests, which is still $O(n \log n)$; thus, our algorithm, with $O(n^2)$, is not guaranteed optimality in this case.

II.2. *Constructing Unrooted Trees*

Let T be an unrooted tree in which the degree of every node is at least three, and let u, v, w, x be any quadruple of leaves. We say that *x pairs with*

u relative to (v, w) if the path from x to u is edge-disjoint relative to the path from v to w.

Remove a leaf x of T and examine the remaining tree T_1 as rooted at the node x_1 with which x is adjacent in T. The following observation is a direct consequence of the definitions:

LEMMA 2. *x pairs with u relative to (v, w) in the tree T if and only if u is the leader of u, v, w in the tree T_1 rooted at x_1.*

The algorithm of Section II.1 can be used for reconstructing unrooted trees out of questions of the form: "Which node pairs with x in the quadruple (x, u, v, w)?"

Choosing arbitrarily a fixed leaf x we use Lemma 2 and the algorithm of Section II.1 to reconstruct T_1 rooted at x_1 and finally add x as a son of x_1 to get the required tree T. The complexity analysis does not change, since an unrooted tree T with $n + 1$ leaves and maximal degree $k + 1$ will provide T_1 with n nodes and at most k sons for every node.

ACKNOWLEDGMENTS

Norman Dalkey has called our attention to the work of Lazarfeld (1966) and has provided a continuous stream of valuable advice. Thomas Ferguson made helpful comments on an early version of the manuscript. A summary of the main results was presented at the International Joint Conference on Artificial Intelligence, University of California, Los Angeles, August 19–23, 1985.

REFERENCES

CHOW, C. K., AND LIU, C. N. (1968), Approximating discrete probability distributions with dependence trees, *IEEE Trans. Inform. Theory* **IT-14,** 462–467.

HINTON, G. E., SEJNOWSKI, T. J., AND ACKLEY, D. H. (1984), "Boltzmann Machines: Constraint Satisfaction Networks That Learn," Technical Report CMU-CS-84-119, Department of Computer Science, Carnegie–Mellon University.

KIM, J., AND PEARL, J. (1983), A computational model for combined causal and diagnostic reasoning in inference systems, *in* "Proceedings, International Joint Conference on Artificial Intelligence—83, Karlsruhe, West Germany, August," pp. 190–193.

LAZARFELD, P. (1966), Latent structure analysis, *in* "Measurement and Prediction" (Stouffer, Guttman, Slachman, Lazarfeld, Star, and Claussen, Eds.), Wiley, New York.

PEARL, J. (1982), Reverend Bayes on inference engines: A distributed hierarchical approach, *in* "Proceedings, AAAI National Conference on Artificial Intelligence, Pittsburgh, Pa., August," pp. 133–136.

SIMON, H. A. (1954), Spurious correlations: A causal interpretation, *J. Amer. Statist. Assoc.* **49,** 467–492.

SUPPES, P. (1970), "A Probabilistic Theory of Causality," North-Holland, Amsterdam.