Learning Causal Trees from Dependence Information *

Dan Geiger
dgeiger@nrtc.northrop.com
Northrop Research and Technology Center
One Research Park
Palos Verdes, CA 90274

Azaria Paz
paz@techsel.bitnet
Computer Science Department
Israel Institute of Technology
Haifa, Israel 32000

Judea Pearl
judea@cs.ucla.edu
Cognitive Systems Lab.
University of California
Los Angeles, CA 90024

Abstract

In constructing probabilistic networks from human judgments, we use causal relationships to convey useful patterns of dependencies. The converse task, that of inferring causal relationships from patterns of dependencies, is far less understood. This paper establishes conditions under which the directionality of some interactions can be determined from non-temporal probabilistic information — an essential prerequisite for attributing a causal interpretation to these interactions. An efficient algorithm is developed that, given data generated by an undisclosed causal polytree, recovers the structure of the underlying polytree, as well as the directionality of all its identifiable links.

1 Introduction

The study of causation, because of its pervasive usage in human communication and problem solving, is central to the understanding of human reasoning. All reasoning tasks dealing with changing environments rely heavily on the distinction between cause and effect. For example, a central task in applications such as diagnosis, qualitative physics, plan recognition and language understanding, is that of abduction, i.e., finding a satisfactory explanation to a given set of observations, and the meaning of explanation is intimately related to the notion of causation.

Most AI works have given the term "cause" a procedural semantics, attempting to match the way people use it in inference tasks, but were not concerned with what makes people believe that "a causes b", as opposed to, say, "b causes a" or "c causes both a and b." [de Kleer & Brown 78,Simon 54]. An empirical semantics for causation is important for several reasons. First, by formulating the empirical components of causation we gain a better understanding of the meaning conveyed by causal utterances, such as "a explains b", "a suggests b", "a tends to cause b", and "a actually caused b". These utterances are the basic building blocks from which knowledge bases are assembled. Second, any autonomous learning system attempting to build a causal model of its environment cannot rely exclusively on procedural semantics but must be able to translate direct observations to cause and effect relationships.

Temporal precedence is normally assumed essential for defining causation. Suppes [Suppes 70], for example, introduces a probabilistic definition of causation with an explicit requirement that a cause precedes its effect in time. Shoham makes an identical assumption [Shoham 87]. In this paper we propose a non-temporal semantics, one that determines the directionality of causal influence without resorting to temporal information, in the spirit of [Simon 54] and [Glymour at al. 87]. Such semantics should be applicable, therefore, to the organization of concurrent events or events whose chronological precedence cannot be determined empirically. Such situations are common in the behavioral and medical sciences where we say, for example, that old age explains a certain disability, not the other way around, even though the two occur together (in many cases it is the disability that precedes old age).

Another feature of our formulation is the appeal to probabilistic dependence, as opposed to functional or deterministic dependence. This is motivated by the observation that most causal connections found in natural discourse, for example "reckless driving causes accidents" are probabilistic in nature [Spohn 90]. Given that statistical analysis cannot distinguish causation from covariation, we must still identify the asymmetries that prompt people to perceive causal structures in empirical data, and we must find a computational model for such perception.

Our attack on the problem is as follows; first, we pretend that Nature possesses "true" cause and effect relationships and that these relationships can be represented by a *causal network*, namely, a directed acyclic graph where each node represents a variable in the domain and the parents of that node correspond to its

*This work was supported in part by the National Science Foundation Grant # IRI-8821444

direct causes, as designated by Nature. Next, we assume that Nature selects a joint distribution over the variables in such a way that direct causes of a variable render this variable conditionally independent of all other variables except its consequences. Nature permits scientists to observe the distribution, ask questions about its properties, but hides the underlying causal network. We investigate the feasibility of recovering the network's topology efficiently and uniquely from the joint distribution.

This formulation contains several simplifications of the actual task of scientific discovery. It assumes, for example, that scientists obtain the distribution, rather than events sampled from the distribution. This assumption is justified when a large sample is available, sufficient to reveal all the dependencies embedded in the distribution. Additionally, it assumes that all relevant variables are measurable, and this prevents us from distinguishing between *spurious correlations* [Simon 54] and genuine causes, a distinction that is impossible within the confines of a closed world assumption. Computationally, however, solving this simplified problem is an essential component in any attempt to deduce causal relationships from measurements, and that is the main concern of this paper.

It is not hard to see that if Nature were to assign totally arbitrary probabilities to the links, then some distributions would not enable us to uncover the structure of the network. However, by employing additional restrictions on the available distributions, expressing properties we normally attribute to causal relationships, some structure could be recovered. The basic requirement is that two independent causes should become dependent once their effect is known [Pearl 88]. For example, two independent inputs for an AND gate become dependent once the output is measured. This observation is phrased axiomatically by a property called *Marginal Weak Transitivity* (Eq. 9 below). This property tells us that if two variables x and y are mutually independent, and each is dependent on their effect c, then x and y are conditionally dependent for at least one instance of c. Two additional properties of independence, intersection and composition (Eqs. 7, and 8 below), are found useful. Intersection is guaranteed if the distributions are strictly positive and is justified by the assumption that, to some extent, all observations are corrupted by noise. Composition is a property enforced, for example, by multivariate normal distributions, stating that two sets of variables X and Y are independent iff every $x \in X$ and $y \in Y$ are independent. In common discourse, this property is often associated with the notion of "independence", yet it is not enforced by all distributions.

The theory to be developed in the rest of the paper addresses the following problem. We are given a distribution P and we know that P is represented as a *singly-connected* dag D whose structure is unknown (such a dag is also called a *Polytree* [Pearl 88]). What properties of P allow the recovery of D ? It is shown that intersection composition and marginal weak transitivity are sufficient properties to ensure that the dag is uniquely recoverable (up to *isomorphism*) in polynomial time. The recovery algorithm developed considerably generalizes the method of Rebane and Pearl for the same task, as it does not assume the distribution to be *dag-isomorph* [Pearl 88, Chapter 8]. The algorithm implies, for example, that the assumption of a multivariate normal distribution is sufficient for a complete recovery of singly-connected dags.

2 Probabilistic Dependence: Background and Definitions

Our model of an environment consists of a finite set of variables U and a distribution P over these variables. Variables in a medical domain, for example, represent entities such as "cold", "headache", "fever". Each variable has a *domain* which is a set of permissible values. The sample space of the distribution is the Cartesian product of all domains of the variables in U. An environment can be represented graphically by an acyclic directed graph (dag) as follows: We select a linear order on all variables in U. Each variable is represented by a node. The parents of a node v correspond to a minimal set of variables that make v conditionally independent of all lesser variables in the selected order. Each ordering may produce a different graph, for example, one representation of the three variables above is the chain *headache* $\leftarrow$ *cold* $\rightarrow$ *fever* which is produced by the order *cold*, *headache* and *fever* (assuming *fever* and *headache* are independent symptoms of a *cold*). Another ordering of these variables: *fever*, *cold*, and *headache* would yield the dag *headache* $\leftarrow$ *cold* $\leftarrow$ *fever* with an additional arc between *fever* and *headache*. Notice that the directionality of links may differ between alternative representations. In the first graph directionality matches our perception of cause-effect relationships while in the second it does not, being merely a spurious by-product of the ordering chosen for the construction. We shall see that, despite the arbitrariness in choosing the construction ordering, some directions will be preferred to others, and these can be determined mechanically.

The basis for differentiating alternative representations are the dependence relationships encoded in the distribution describing the environment. We regard a distribution as a *dependency model*, capable of answering queries of the form "Are X and Y independent given Z ?" and use these answers to select among possible representations. The following definitions and theorems provide the ground for a precise formulation of the problem.

Definition [Pearl & Paz 89] *A* dependency model M *over a finite set of elements U is any subset of triplets (X, Z, Y) where X, Y and Z are disjoint subsets of U.*

The interpretation of $(X, Z, Y) \in M$ is the sentence

"X is independent of Y, given Z", denoted also by $I(X, Z, Y)$. When speaking about dependency models, we use both set notations and logic notations. If $(X, Z, Y) \in M$, we say that the *independence statement* $I(X, Z, Y)$ holds for M. Similarly, we either say that M contains a triplet (X, Z, Y) or that M satisfies a statement $I(X, Z, Y)$. An independence statement $I(X, Z, Y)$ is called an *independency* and its negation is called a *dependency*. Every probability distribution defines a dependency model:

Definition [Pearl & Paz 89]*: Let U be a finite set of variables. A* Probabilistic Dependency Model M_P *is defined in terms of a probability distribution P with a sample space $\prod_{u_i \in U} d(u_i)$, the Cartesian product of $d(u_i)$, where $d(u_i)$ is the domain of u_i. If X, Y and Z are three disjoint subsets of U, and* $\mathbf{X}$, $\mathbf{Y}$ *and* $\mathbf{Z}$ *are any instances from the domains of the variables in these subsets, then by definition $(X, Z, Y) \in M_P$ iff*

$$P(\mathbf{X}, \mathbf{Y}|\mathbf{Z}) = P(\mathbf{X}|\mathbf{Z}) \cdot P(\mathbf{Y}|\mathbf{Z}) \qquad (1)$$

The definition above is suitable also for normal distributions, in which case the distribution function P in Eq. (1) is replaced by the normal density functions. The conditional density functions are well defined for normal distributions if all variances are finite.

Dependency models can also be encoded in graphical forms. The following graphical definition of dependency models is motivated by regarding directed acyclic graphs as a representation of causal relationships. Designating a node for every variable and assigning a link between every cause to each of its direct consequences defines a graphical representation of a causal hierarchy. For example, the propositions "It is raining" (r), "the pavement is wet" (w) and "John slipped on the pavement" (s) are well represented by a three node chain, from r through w to s ; it indicates that rain and wet pavement could cause slipping, yet wet pavement is designated as the *direct cause*; rain could cause someone to slip if it wets the pavement, but not if the pavement is covered. Moreover, knowing the condition of the pavement renders "slipping" and "raining" independent, and this is represented graphically by showing node r and s separated from each other by node w. Furthermore, if we assume that "broken pipe" (b) is another direct cause for wet pavement, as in Figure 1, then an induced dependency exists between the two events that may cause the pavement to get wet: "rain" and "broken pipe". Although they appear connected in Figure 1, these propositions are marginally independent and become dependent once we learn that the pavement is wet or that someone broke his leg. An increase in our belief in either cause would decrease our belief in the other as it would "explain away" the observation.

The following definition of d-separation permits us to graphically identify such induced dependencies from the network. A preliminary definition is needed.

Definition *A* trail *in a dag is a sequence of links that form a path in the underlying undirected graph. A trail is said to contain the nodes adjacent to its links. A node b is called a head-to-head node with respect to a trail t if there are two consecutive links $a \rightarrow b$ and $b \leftarrow c$ on t. A node that starts or ends a trail t is not a head-to-head node with respect to t*[1] .

Definition [Pearl 88] *If X, Y, and Z are three disjoint subsets of nodes in a dag D, then Z is said to d-separate X from Y, denoted $I(X, Z, Y)_D$, iff there exists no trail t between a node in X and a node in Y along which (1) every head-to-head node (wrt t) either is or has a descendent in Z and (2) every node that delivers an arrow along t is outside Z. A trail satisfying the two conditions above is said to be* active. *Otherwise, it is said to be* blocked *(by Z).*

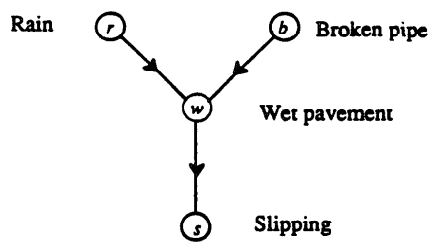

Figure 1

Definition *A* Dag Dependency Model M_D *is defined in terms of a directed acyclic graph D. If X, Y and Z are three disjoint sets of nodes in D, then, by definition, $(X, Z, Y) \in M_D$ iff there is no active trail by Z between nodes in X and Y.*

For example, in Figure 1, $(r, \emptyset, b) \in M_D$, $(r, s, b) \notin M_D$, $(r, \{w, s\}, b) \notin M_D$, and $(r, w, s) \in M_D$.

These two distinct types of dependency models: graphical and probabilistic provide different formalisms for the notion of "independent". The similarity between these models is summarized axiomatically by the following definition of graphoids.

Definition [Pearl & Paz 89] *A* graphoid *is any dependency model M which is closed under the following inference rules, considered as* axioms[2]*:*

Trivial Independence

$$I(X, Z, \emptyset) \qquad (2)$$

Symmetry

$$I(X, Z, Y) \Rightarrow I(Y, Z, X) \qquad (3)$$

[1]The definitions of undirected graphs, acyclic graphs, trees, paths, adjacent links and nodes can be found in any text on graph algorithms (e.g., [Even 79]).

[2]This definition differs slightly from that given in [Pearl & Paz 89] where axioms (3) through (6) define semigraphoid and dependency models obeying also (7) are called graphoids. Axiom (2) is added for future clarity.

Decomposition

$$I(X, Z, Y \cup W) \Rightarrow I(X, Z, Y) \quad (4)$$

Weak union

$$I(X, Z, Y \cup W) \Rightarrow I(X, Z \cup W, Y) \quad (5)$$

Contraction

$$I(X, Z, Y) \,\&\, I(X, Z \cup Y, W) \Rightarrow I(X, Z, Y \cup W) \quad (6)$$

Intuitively, the essence of these axioms lies in Eqs. (5) and (6). If we associate dependency with informational *relevance*, these equations assert that when we learn an irrelevant fact, all relevance relationships among other variables in the system should remain unaltered; any information that was relevant remains relevant and that which was irrelevant remains irrelevant. These axioms are very similar to those assembled by Dawid [Dawid 79] for probabilistic conditional independence, those proposed by Smith [Smith 89] for *Generalized Conditional Independence* and those used by Spohn [Spohn 80] in his exploration of *causal independence*. We shall henceforth call axioms (2) through (6) *graphoid axioms*. It can readily be shown that the two dependency models presented thus far, the probabilistic and the graphical, are both graphoids. Several additional graphoids are discussed in [Pearl & Paz 89,Pearl & Verma 87].

Definition *A dag is an* independence-map (I-map) *of a graphoid M if whenever X and Y are* d-*separated by Z in D, then $I(X, Z, Y)$ holds for M. In other words, $M_D \subseteq M$, where M_D is the dependency model defined by D. A dag D is a* minimal-edge I-map *of M if deleting any edge of D would make D cease to be an* I-*map of M.*

Definition [Pearl 88] *A dag D is called a* Causal network *of a dependency model M, if D is a minimal-edge* I-*map of M.*

The task of finding a dag which is a minimal-edge I-map of a given distribution P was solved in [Pearl & Verma 87,Verma & Pearl 88]. The algorithm consists of the following steps: assign a total ordering d to the variables of P. For each variable a_i of P, identify a minimal set of predecessors $\pi(a_i)$ that renders a_i independent of all its other predecessors in the ordering of the first step. Assign a direct link from every variable in $\pi(a_i)$ to a_i. The resulting dag is an I-map of P, and is minimal in the sense that no edge can be deleted without destroying its I-mapness. The input L for this construction consists of n conditional independence statements, one for each variable, all of the form $I(a_i, \pi(a_i), U(a_i) - \pi(a_i))$ where $U(a_i)$ is the set of predecessors of a_i and $\pi(a_i)$ is a subset of $U(a_i)$ that renders a_i conditionally independent of all its other predecessors. This set of conditional independence statements is said to *generate* a dag and is called a *recursive basis* drawn from P.

The theorem below summarizes the discussion above.

Theorem 1 [Verma & Pearl 88] *If M is a graphoid, and L is any recursive basis drawn from M, then the dag generated by L is an* I-*map of M.*

Note that a probability model may possess many causal networks each corresponding to a different ordering of its variables in the recursive basis. If temporal information is available, one could order the variables chronologically and this would dictate an almost-unique dag representation (except for the choice of $\pi(a_i)$). However, in the lack of temporal information the directionality of links must be extracted from additional requirements about the graphical representation. Such requirements are identified below.

3 Reconstructing Singly Connected Causal Networks

We shall restrict our discussion to singly connected causal networks, namely networks where every pair of nodes is connected via no more then one trail and to distributions that are close to normal (Gausian) in the sense that they adhere to axioms (7) through (9) below, as do all multivariate normal distributions with finite variances and non-zero means.

Lemma 2 *The following axioms are satisfied by normal distributions.*

Intersection

$$I(X, Z \cup Y, W) \,\&\, I(X, Z \cup W, Y) \Rightarrow I(X, Z, Y \cup W) \quad (7)$$

Composition

$$I(X, Z, Y) \,\&\, I(X, Z, W) \Rightarrow I(X, Z, Y \cup W) \quad (8)$$

Marginal Weak Transitivity

$$I(X, \emptyset, Y) \,\&\, I(X, c, W) \Rightarrow I(X, \emptyset, c) \text{ or } I(c, \emptyset, Y) \quad (9)$$

Definition *A graphoid (e.g., a distribution) is called* intersectional *if it satisfies (7),* semi-normal *if it satisfies (7) and (8), and* pseudo-normal *if it satisfies (7) through (9).*

Definition *A* singly-connected *dag (or a* polytree*) is a directed acyclic graph with at most one trail connecting any two nodes. A dag is* non-triangular *if any two parents of a common node are never parents of each other. Polytrees are examples of non-triangular dags. The skeleton of a dag D, denoted* skeleton(D)*, is the undirected graph obtained from D if the directionality of the links is ignored. The skeleton of a polytree is a tree.*

Definition *A Markov network G_0 of an intersectional graphoid M is the network formed by connecting two nodes, a and b, if and only if $(a, U \setminus \{a, b\}, b) \notin M$. A* reduced graph *$G_R$ of M is the graph obtained from G_0 by removing any edge $a - b$ for which $(a, \emptyset, b) \in M$.*

Definition *Two dags D_1 and D_2 are* isomorphic *if the corresponding dependency models are equal.*

Isomorphism draws the theoretical limitation of the ability to identify directionality of links using information about independence. For example, the two dags: $a \rightarrow b \rightarrow c$ and $a \leftarrow b \leftarrow c$, are indistinguishable in the sense that they portray the same set of independence assertions; these are isomorphic dags. On the other hand, the dag $a \rightarrow b \leftarrow c$ is distinguishable from the previous two because it portrays a new independence assertion, $I(a, \emptyset, c)$, which is not represented in either of the former dags. An immediate corollary of the definitions of d-separation and isomorphism is that any two polytrees sharing the same skeleton and the same head-to-head connections must be isomorphic.

Lemma 3 *Two polytrees T_1 and T_2 are* isomorphic *iff they share the same skeleton, and the same head-to-head connections.*

Sufficiency: *If T_1 and T_2 share the same skeleton and the same head-to-head connections then every active trail in T_1 is an active trail in T_2 and vice versa. Thus, M_{T_1} and M_{T_2}, the dependency models corresponding to T_1 and T_2 respectively, are equal.*

Necessity: *T_1 and T_2 must have the same set of nodes U, for otherwise their dependency models are not equal. If $a \rightarrow b$ is a link in T_1 and not in T_2, then the triplet $(a, U \setminus \{a, b\}, b)$ is in M_{T_1} but not in M_{T_2}. Thus, if M_{T_1} and M_{T_2} are equal, then T_1 and T_2 must have the same skeleton. Assume T_1 and T_2 have the same skeleton and that $a \rightarrow c \leftarrow b$ is a head-to-head connection in T_1 but not in T_2. The trail $a - c - b$ is the only trail connecting a and b in T_2 because T_2 is singly-connected and it has the same skeleton as T_1. Since c is not a head-to-head node wrt this trail, $(a, c, b) \in M_{T_2}$. However, $(a, c, b) \notin M_{T_1}$ because the trail $a \rightarrow c \leftarrow b$ is activated by c. Thus, if M_{T_1} and M_{T_2} are equal, then T_1 and T_2 must have the same head-to-head connections.* □

More generally, it can be shown that two dags are isomorphic iff they share the same skeleton and the same head-to-head nodes emanating from non adjacent sources [Pear, Geiger & Verma 89].

The algorithm below uses queries of the form $I(X, Z, Y)$ to decide whether a pseudo-normal graphoid M (e.g., a normal distribution) has a polytree I-map representation and if it does, it's topology is identified. Axioms (7) through (9) are then used to prove that if D exists, then it is *unique* up to isomorphism. The algorithm is remarkably efficient; it requires only polynomial time (in the number of independence assertions), while a brute force approach would require checking $n!$ possible dags, one for each ordering of M's variables. One should note, however, that validating each such assertion from empirical data may require extensive computation.

The Recovery Algorithm

Input: Independence assertions of the form $I(X, Z, Y)$ drawn from a pseudo-normal graphoid M.
Output: A polytree I-map of M if such exists, or acknowledgment that no such I-map exists.

1. Start with a complete graph.
2. Construct the Markov network G_0 by removing every edge $a - b$ for which $(a, U \setminus \{a, b\}, b)$ is in M.
3. Construct G_R by removing from G_0 any link $a - b$ for which $(a, \emptyset, b)$ is in M. If the resulting graph G_R has a cycle then answer "NO". Exit.
4. Orient every link $a - b$ in G_R towards b if b has a neighboring node c, such that $(a, \emptyset, c) \in M$ and $a - c$ is in G_0.
5. Orient the remaining links without introducing new head-to-head connections. If the resulting orientation is not feasible answer "NO". Exit.
6. If the resulting polytree is not an I-map, answer "NO". Otherwise, this polytree is a minimal-edge I-map of M.

The following sequence of claims establishes the correctness of the algorithm and the uniqueness of the recovered network; full proofs are given in [Geiger 90].

Theorem 4 *Let D be a non-triangular dag that is a minimal-edge* I*-map of an intersectional graphoid M. Then, for every link $a - b$ in D, $(a, U \setminus \{a, b\}, b) \notin M$.*

Theorem 4 ensures that every link in a minimal-edge polytree I-map (or more precisely, a link in a minimal-edge non-triangular dag I-map) must be a link in the Markov network G_0. Thus, we are guaranteed that Step 2 of the algorithm does not remove links that are needed for the construction of a minimal-edge polytree I-map.

Theorem 5 *Let M be a semi-normal graphoid that has a minimal-edge polytree* I*-map T. Then, the reduced graph G_R of M equals skeleton(T).*

Corollary 6 *All minimal-edge polytree* I*-maps of a semi-normal graphoid have the same skeleton (Since G_R is unique).*

Theorem 5 shows that by computing G_R, the algorithm identifies the skeleton of any minimal-edge polytree I-map T, if such exists. Thus, if G_R has a cycle, then M has no polytree I-map and if M does have a polytree I-map, then it must be one of the orientations of G_R. Hence by checking all possible orientations of the links of the reduced graph one can decide whether a semi-normal graphoid has a minimal-edge polytree I-map. The next two theorems justify a more efficient way of establishing the orientations of G_R. Note that composition and intersection, which are properties of semi-normal graphoids, are sufficient to ensure that the skeleton of a polytree I-map of M is uniquely recoverable. Marginal weak transitivity, which is a property

of pseudo-normal graphoids, is used to ensure that the algorithm orients the skeleton in a valid way. It is not clear, however, whether axioms (7) through (9) are indeed necessary for a unique recovery of polytrees.

Definition *Let M be a pseudo-normal graphoid for which the reduced graph G_R has no cycles. A* partially oriented polytree *P of M is a graph obtained form G_R by orienting a subset of the links of G_R using the following rule: A link $a \rightarrow b$ is in P if $a - b$ is a link in G_R, b has a neighboring node c, such that $(a, \emptyset, c) \in M$ and the link $a - c$ is in G_0. All other links in P are undirected.*

Theorem 7 *If M is a semi-normal graphoid that has a polytree* I-*map, then M defines a unique partially oriented polytree P.*

Theorem 8 *Let P be a partially oriented polytree of a semi-normal graphoid M. Then, every oriented link $a \rightarrow c$ of P is part of every minimal-edge polytree* I-*map of M.*

Theorem 7 guarantees that the rule by which a partially oriented polytree is constructed cannot yield a conflicting orientation when M is pseudo-normal. Theorem 8 guarantees that the links that are oriented in P are oriented correctly, thus justifying Step 4.

We have thus shown that the algorithm identifies the right skeleton and that every link that is oriented must be oriented that way if a polytree I-map exists. It remains to orient the rest of the links.

Theorem 9 below shows that no polytree I-map of M introduces new head-to-head connections, hence, justifying Step 5. Lemma 3, further shows that all orientations that do not introduce a head-to-head connection yield isomorphic dags because these polytrees share the same skeleton and the same head-to-head connections. Thus, in order to decide whether or not M has a polytree I-map, it is sufficient to examine merely a single polytree for I-mapness, as performed by Step 6.

Theorem 9 *Let P be a partially oriented Polytree of a pseudo-normal graphoid M. Every orientation of the undirected links of P which introduces a new head-to-head connection to P yields a polytree that is not a minimal-edge* I-*map of M.*

4 Summary and Discussion

In the absence of temporal information, discovering directionality in interactions is essential for inferring causal relationships. This paper provides conditions under which the directionality of some links in a probabilistic network is uniquely determined by the dependencies that surround the link. It is shown that if a distribution is generated from a singly connected causal network (i.e., a polytree), then the topology of the network can be recovered uniquely, provided that the distribution satisfies three properties: composition, intersection and marginal weak transitivity. Although the assumption of singly-connectedness is somewhat restrictive, it may not be essential for the recovery algorithm, because Theorem 1, the basic step of the recovery, assumes only non-triangularity. Thus, an efficient recovery algorithm for non-triangular dags may exist as well. Additionally, the recovery of singly connected networks demonstrates the feasibility of extracting causal asymmetries from information about dependencies, which is inherently symmetric. It also highlights the nature of the asymmetries that need be detected for the task.

Another useful feature of our algorithm is that its input can be obtained either from empirical data or from expert judgments or a combination thereof. Traditional methods of data analysis rely exclusively on gathered statistics which domain experts find hard to provide. Independence assertions, on the other hand, are readily available from domain experts.

We are far from claiming that the method presented in this paper discovers genuine physical influences between causes and effects. First, a sensitivity analysis is needed to determine how vulnerable the algorithm is to errors associated with inferring conditional independencies from sampled data. Second, such a discovery requires breaking away from the confines of the closed world assumption, while we have assumed that the set of variables U adequately summarizes the domain, and remains fixed throughout the structuring process. This assumption does not enable us to distinguish between genuine causes and spurious correlations [Simon 54]; a link $a \rightarrow b$ that has been determined by our procedure may be represented by a chain $a \leftarrow c \rightarrow b$ where c is a variable not accounted for when the network is first constructed. Thus, the dependency between a and b which is marked as causal when $c \notin U$ is in fact spurious, and this can only be revealed when c becomes observable. Such transformations are commonplace in the development of scientific thought: What is currently perceived as a cause may turn into a spurious effect when more refined knowledge becomes available. The initial perception, nevertheless serves an important cognitive function in providing a tentative and expedient encoding of dependence patterns in that level of abstraction.

Future research should explore structuring techniques that incorporate variables outside U. The addition of these so called "hidden" variables often renders graphical representations more compact and more accurate. For example, a network representing a collection of interrelated medical symptoms would be highly connected and of little use, but when a disease variable is added, the interactions can often be represented by a singly connected network. Facilitating such decomposition is the main role of "hidden variables" in neural networks [Hinton 89] and is also incorporated in the program TETRAD citebk:glymour. Pearl and Tarsi provide an algorithm that generates tree representations with hidden variables, whenever such a representation exists [Pearl & Tarsi 86]. An extension of this

algorithm to polytrees would further enhance our understanding of causal structuring.

Another valuable extension would be an algorithm that recovers general dags. Such algorithms have been suggested for distributions that are *graph-isomorph* [Spirtes, Glymour & Scheines 89, Verma 90]. The basic idea is to identify with each pair of variables x and y a minimal subset S_{XY} of other variables[3] that shields x from y, to link by an edge any two variables for which no such subset exists, and to direct an edge from x to y if there is a variable z linked to y but not to x, such that $I(x, S_{XZ} \cup y, z)$ does not hold (see Pearl 1988, page 397, for motivation). The algorithm of Spirtes et al. (1989) requires an exhaustive search over all subsets of variables, while that of Verma (1990) prunes the search starting from the Markov net. It is not clear, however, whether the assumption of dag isomorphism is realistic in processing real-life data such as medical records or natural language texts.

References

[Dawid 79] Dawid, A. P. 1979. Conditional independence in statistical theory. *Journal Royal Statistical Society, Series B*, 41(1):1–31.

[de Kleer & Brown 78] de Kleer, J.; and Brown, J. S. 1978. Theories of causal ordering. *AI*, 29(1):33–62.

[Even 79] Even, S. 1979. *Graph Algorithm.* Computer Science Press, Potomac MD.

[Geiger 90] Geiger, D. 1990. *Graphoids: A Qualitative Framework for Probabilistic Inference.* PhD thesis, UCLA Computer Science Department. Also appears as a Technical Report (R-142) Cognitive Systems Laboratory, CS, UCLA.

[Glymour at al. 87] Glymour, C.; Scheines, R.; Spirtes, P.; and Kelly, K. 1987. *Discovering Causal Structre.* Academic Press, New York.

[Hinton 89] Hinton, G. E. 1989. Connectionist learning procedures. *AI*, 40(1-3):185–234.

[Pear, Geiger & Verma 89] Pearl, J.; Geiger, D.; and Verma, T. S. 1989. The logic of influence diagrams. In J. Q. Smith R. M. Oliver (ed.), *Influence Diagrams, Beliefnets and Decision analysis*, chapter 3. John Wiley & Sons Ltd. New York.

[Pearl & Paz 89] Pearl, J.; and Paz, A. 1989. Graphoids: A graph-based logic for reasoning about relevance relations. In B. Du Boulay et al. (ed.), *Advances in Artificial Intelligence-II*, pages 357–363. North Holland, Amsterdam.

[Pearl & Verma 87] Pearl, J.; and Verma, S. T. 1987. The logic of representing dependencies by directed acyclic graphs. In *AAAI*, pages 347–379, Seattle Washington.

[3] the set S_{XY} contains ancestors of x or y

[Pearl 88] Pearl, J. 1988. *Probabilistic reasoning in intelligent systems.* Morgan-Kaufman, San Mateo.

[Pearl & Tarsi 86] Pearl, J.; and Tarsi, M. 1986. Structuring causal trees. *Journal of Complexity*, 2:60–77.

[Shoham 87] Shoham, Y. 1987. *Reasoning About Change.* MIT Press, Boston MA.

[Simon 54] Simon, H. 1954. Spurious correlations: A causal interpretation. *Journal Amercan Statistical Association*, 49:469–492.

[Smith 89] Smith, J. Q. 1989. Influence diagrams for statistical modeling. *Annals of Statistics*, 17(2):654–672.

[Spirtes, Glymour & Scheines 89] Spirtes, P.; Glymour, C.; and Scheines, R. 1989. Causality from probability. Technical Report CMU-LCL-89-4, Department of Philosophy Carnegie-Mellon University.

[Spohn 80] Spohn, W. 1980. Stochastic independence, causal independence, and shieldability. *Journal of Philosophical Logic*, 9:73–99.

[Spohn 90] Spohn, W. 1990. Direct and indirect causes. *Topoi*, 9.

[Suppes 70] Suppes, P. 1970. *A probabilistic theory of causation.* North Holland, Amsterdam.

[Verma 90] Verma, S. T. 1990. Learning causal structure from independence information. in preparation.

[Verma & Pearl 88] Verma, T. S.; and Pearl J. 1988. Causal networks: Semantics and expressiveness. In *Forth Workshop on Uncertainty in AI*, pages 352–359, St. Paul Minnesota.