Fusion, Propagation, and Structuring in Belief Networks*

Judea Pearl
Cognitive Systems Laboratory, Computer Science Department, University of California, Los Angeles, CA 90024, U.S.A.

Recommended by Patrick Hayes

ABSTRACT

Belief networks are directed acyclic graphs in which the nodes represent propositions (or variables), the arcs signify direct dependencies between the linked propositions, and the strengths of these dependencies are quantified by conditional probabilities. A network of this sort can be used to represent the generic knowledge of a domain expert, and it turns into a computational architecture if the links are used not merely for storing factual knowledge but also for directing and activating the data flow in the computations which manipulate this knowledge.

The first part of the paper deals with the task of fusing and propagating the impacts of new information through the networks in such a way that, when equilibrium is reached, each proposition will be assigned a measure of belief consistent with the axioms of probability theory. It is shown that if the network is singly connected (e.g. tree-structured), then probabilities can be updated by local propagation in an isomorphic network of parallel and autonomous processors and that the impact of new information can be imparted to all propositions in time proportional to the longest path in the network.

The second part of the paper deals with the problem of finding a tree-structured representation for a collection of probabilistically coupled propositions using auxiliary (dummy) variables, colloquially called "hidden causes." It is shown that if such a tree-structured representation exists, then it is possible to uniquely uncover the topology of the tree by observing pairwise dependencies among the available propositions (i.e., the leaves of the tree). 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

This study was motivated by attempts to devise a computational model for humans' inferential reasoning, namely, the mechanism by which people integrate data from multiple sources and generate a coherent interpretation of that data. Since the knowledge from which inferences are drawn is mostly judg-

* This work was supported in part by the National Science Foundation, Grant#DSR 83-13875.

mental—subjective, uncertain and incomplete—a natural place to start would be to cast the reasoning process in the framework of probability theory. However, the mathematician who approaches this task from the vantage point of probability theory may dismiss it as a rather prosaic exercise. For, if one assumes that human knowledge is represented by a joint probability distribution, $P(x_1, \ldots, x_n)$, on a set of propositional variables, $x_1, \ldots, x_n$, the task of drawing inferences from observations amounts to simply computing the probabilities of a small subset, $H_1, \ldots, H_k$, of variables called hypotheses, conditioned upon a group of instantiated variables, $e_1, \ldots, e_m$, called evidence. Indeed, computing $P(H_1, \ldots, H_k|e_1, \ldots, e_m)$ from a given joint distribution on all propositions is merely arithmetic tedium, void of theoretical or conceptual interest.

It is not hard to see that this textbook view of probability theory presents a rather distorted picture of human reasoning and misses its most interesting aspects. Consider, for example, the problem of encoding an arbitrary joint distribution, $P(x_1, \ldots, x_n)$, on a computer. If we need to deal with n propositions, then to store $P(x_1, \ldots, x_n)$ explicitly would require a table with 2^n entries—an unthinkably large number, by any standard. Moreover, even if we found some economical way of storing $P(x_1, \ldots, x_n)$ (or rules for generating it), there would still remain the problem of manipulating it to compute the probabilities of propositions which people consider interesting. For example, computing the marginal probability $P(x_i)$ would require summing $P(x_1, \ldots, x_n)$ over all 2^{n-1} combinations of the remaining $n-1$ variables. Similarly, computing the conditional probability $P(x_1|x_j)$ from its textbook definition $P(x_1|x_j) = P(x_1, x_j)/P(x_j)$ would involve dividing two marginal probabilities, each resulting from summation over an exponentially large number of variable combinations. Human performance, by contrast, exhibits a different complexity ordering: probabilistic judgments on a small number of propositions (especially two-place conditional statements such as the likelihood that a patient suffering from a given disease will develop a certain type of complication) are issued swiftly and reliably, while judging the likelihood of a conjunction of many propositions entails a great degree of difficulty and hesitancy. This suggests that the elementary building blocks which make up human knowledge are not the entries of a joint-distribution table but, rather, the low-order marginal and conditional probabilities defined over small clusters of propositions.

Further light on the structure of probabilistic knowledge can be shed by observing how people handle the notion of independence. Whereas a person may show reluctance to giving a numerical estimate for a conditional probability $P(x_i|x_j)$, that person can usually state with ease whether x_i and x_j are dependent or independent, namely, whether or not knowing the truth of x_j will alter the belief in x_i. Likewise, people tend to judge the three-place relationships of conditional dependency (i.e., x_i influences x_j given x_k) with clarity, conviction, and consistency.

This suggests that the notions of dependence and conditional dependence are more basic to human reasoning than are the numerical values attached to probability judgments. (This is contrary to the picture painted in most textbooks on probability theory, where the latter is presumed to provide the criterion for testing the former.) Moreover, the nature of probabilistic dependency between propositions is similar in many respects to that of connectivity in graphs. For instance, we find it plausible to say that a proposition q affects proposition r *directly*, while s influences r *indirectly*, via q. Similarly, we find it natural to identify a set of direct justifications for q to sufficiently shield it (q) from all other influences and to describe them as the direct neighbors of q [5]. These graphical metaphors suggest that the fundamental structure of human knowledge can be represented by dependency graphs and that mental tracing of links in these graphs are the basic steps in querying and updating that knowledge.

1.1. Belief networks

Assume that we decide to represent our perception of a certain problem domain by sketching a graph in which the nodes represent propositions and the links connect those propositions that we judge to be *directly* related. We now wish to quantify the links with weights that signify the strength and type of dependencies between the connected propositions. If these weights are to reflect summaries of actual experiences, we must first attend to two problems: *consistency* and *completeness*. Consistency guarantees that we do not overload the graph with an excessive number of parameters; overspecification may lead to contradictory conclusions, depending on which parameter is consulted first. Completeness protects us from underspecifying the graph dependencies and guarantees that our conclusion-generating routine will not get deadlocked for lack of information.

One of the attractive features of the traditional joint-distribution representation of probabilities is the transparency by which one can synthesize consistent probability models or detect inconsistencies therein. In this representation, all we need to do to create a complete model, free of inconsistencies, is to assign nonnegative weights to the atomic compartments in the space (i.e., conjunctions of propositions), just making sure the sum of the weights equals one. By contrast, the synthesis process in the graph representation is more hazardous. For example, assume you have three propositional variables, x_1, x_2, x_3, and you want to express their dependencies by specifying the three pairwise probabilities $P(x_1, x_2)$, $P(x_2, x_3)$, $P(x_3, x_1)$. It turns out that this will normally lead to inconsistencies; unless the parameters given satisfy some nonobvious relationship, there exists no probability model that will support all three inputs. By contrast, if we specify the probabilities on only two pairs, incompleteness results; many models exist which conform to the input specification, and we will not be able to provide answers to all probabilistic queries.

Fortunately, the consistency-completeness issue has a simple solution stemming from the chain-rule representation of joint distributions. Choosing an arbitrary order d on the variables $x_1, \ldots, x_n$, we can write[1]:

$$\begin{aligned}&P(x_1, x_2, \ldots, x_n)\\&\quad = P(x_n|x_{n-1}, \ldots, x_1) \cdots P(x_3|x_2, x_1)P(x_2|x_1)P(x_1) .\end{aligned}$$

In this formula, each factor contains only one variable on the left side of the conditioning bar and, in this way, the formula can be used as a prescription for consistently quantifying the dependencies among the nodes of an arbitrary graph. Suppose we are given a directed acyclic graph G in which the arrows pointing at each node x_i emanate from a set S_i of parent nodes judged to be directly influencing x_i, and we wish to quantify the strengths of these influences in a complete and consistent way. If, by direct parents we mean a set of variables which, once we fix their values, would shield x_i from the influence of all other predecessors of x_i (i.e., $P(x_i|S_i) = P(x_i|x_1, \ldots, x_{i-1})$), then the chain-rule formula states that a separate assessment of each child-parents relationship should suffice. We need only assess the conditional probabilities, $P(x_i|S_i)$, by some functions, $F_i(x_i, S_i)$, and make sure these assessments satisfy

$$\sum_{x_i} F_i(x_i, S_i) = 1 , \quad 0 \leq F_i(x_i, S_i) \leq 1 ,$$

where the summation ranges over all values of x_i. This specification is complete and consistent because the product form

$$P(x_1, \ldots, x_n) = \prod_i F_i(x_i, S_i)$$

constitutes a joint probability distribution that supports the assessed quantities. In other words, if we compute the conditional probabilities $P(x_i|S_i)$ dictated by $P(x_1, \ldots, x_n)$, the original assessments $F_i(x_i, S_i)$ will be recovered:

[1] Probabilistic formulae of this kind are shorthand notation for the statement that for any instantiation i of the variables $x_1, x_2, \ldots, x_n$, the probability of the joint event $(x_1 = i_1)$ & $(x_2 = i_2)$ & $\cdots$ & $(x_n = i_n)$ is equal to the product of the probabilities of the corresponding conditional events $(x_1 = i_1)$, $(x_2 = i_2$ if $x_1 = i_1)$, $(x_3 = i_3$ if $(x_2 = i_2$ & $x_1 = i_1))$, ... For this expansion to be valid, we must require that $P(E) > 0$ for all conditioning events E.

$$P(x_i|S_i) = \frac{P(x_i, S_i)}{P(S_i)} = \frac{\sum_{x_j \notin (x_i \cup S_i)} P(x_1, \ldots, x_n)}{\sum_{x_j \notin S_i} P(x_1, \ldots, x_n)} = F_i(x_i, S_i) .$$

So, for example, the distribution corresponding to the graph of Fig. 1 can be written by inspection:

$$P(x_1, x_2, x_3, x_4, x_5, x_6)$$
$$= P(x_6|x_5)P(x_5|x_2, x_3)P(x_4|x_1, x_2)P(x_3|x_1)P(x_2|x_1)P(x_1) .$$

This also leads to a simple method of constructing a dependency-graph representation for any given joint distribution $P(x_1, \ldots, x_n)$. We start by imposing an arbitrary order d on the set of variables, $x_1, \ldots, x_n$, then choose x_1 as a root of the graph and assign to it the marginal probability $P(x_1)$ dictated by $P(x_1, \ldots, x_n)$. Next, we form a node to represent x_2; if x_2 is dependent on x_1, a link from x_1 to x_2 is established and quantified by $P(x_2|x_1)$. Otherwise, we leave x_1 and x_2 unconnected and assign the prior $P(x_2)$ to node x_2. At the ith stage, we form the node x_i and establish a group of directed links to x_i from the smallest subset of nodes $S_i \subseteq \{x_1, \ldots, x_{i-1}\}$ satisfying the condition

$$P(x_i|S_i) = P(x_i|x_{i-1}, \ldots, x_1) .$$

It can be shown that the set of subsets satisfying this condition is closed under intersection; therefore, the minimal subset S_i is unique. Thus, the distribution, $P(x_1, \ldots, x_n)$, together with the order d uniquely identify a set of parent nodes for each variable x_i, and that constitutes a full specification of a directed

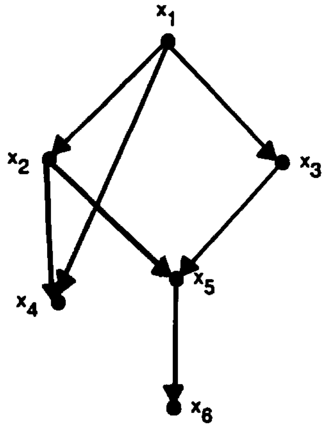


FIG. 1. A typical Bayesian network representing the distribution $P(x_1, \ldots, x_6) = P(x_6|x_5)P(x_5|x_2, x_3)P(x_4|x_1, x_2)P(x_3|x_1)P(x_2|x_1)P(x_1)$.

acyclic graph which represents many of the independencies imbedded in $P(x_1, \ldots, x_n)$.

In expert-systems applications where, instead of a numerical representation for $P(x_i, \ldots, x_n)$, we have only intuitive understanding of the major constraints in the domain, the graph can still be configured by the same modular method as before, except that the parent set S_i must be selected judgmentally. The addition of any new node x_i to the network requires only that the expert identify a set S_i of variables which "directly influence" x_i, locally assess the strength of this relation and make no commitment regarding the effect of x_i on other variables, outside S_i. Even though each judgment is performed locally, their sum total is guaranteed to be consistent. This model-building process permits people to express qualitative relationships perceived to be essential, and the network preserves these qualities, despite sloppy assignments of numerical estimates. In Fig. 1, for example, the fact that x_6 can tell us nothing new about x_3 once we know x_5, will remain part of the model, no matter how carelessly the numbers are assigned.

Graphs constructed by this method will be called *belief networks*, *Bayesian networks*, or *influence networks* interchangeably, the former two to emphasize the judgmental origin and the probabilistic nature of the quantifiers, the latter to reflect the directionality of the links. When the nature of the interactions is perceived to be causal, then the term, *causal network*, may also be appropriate. In general, however, an influence network may also represent associative or inferential dependencies, in which case the directionality of the arrows mainly provides computational convenience [10]. An alternative graphical representation, using undirected graphs, is provided by the so-called Markov fields approach [12] and will not be discussed here. For comparison of properties and applications, see [15, 24, 32].

In the strictest sense, these networks are not graphs but hypergraphs because to describe the dependency of a given node on its k parents requires a function of $k + 1$ arguments which, in general, could not be specified by k two-place functions on the individual links. This, however, does not diminish the advantages of the network representation because the essential interactions between the variables are still displayed by the connecting links. If the number of parents k is large, estimating $P(x_i|S_i)$ may be troublesome because, in principle, it requires a table of size 2^k. In practice, however, people conceptualize causal relationships by forming hierarchies of small clusters of variables (see Section 3.1) and, moreover, the interactions among the factors in each cluster are normally perceived to fall into one of a few prestored, prototypical structures, each requiring about k parameters. Common examples of such prototypical structures are: noisy OR gates (i.e., any one of the factors is likely to trigger the effect), noisy AND gates and various enabling mechanisms (i.e., factors identified as having no influence of their own except enabling other influences to become effective).

Note that the topology of a Bayes network can be extremely sensitive to the node ordering d; a network with a tree structure in one ordering may turn into a complete graph if that ordering is reversed. For example, if $x_1, \ldots, x_n$ stands for the outcomes of n independent coins, and x_{n+1} represents the output of a detector triggered if any of the coins comes up head, then the influence network will be an inverted tree of n arrows pointing from each of the variables $x_1, \ldots, x_n$ toward x_{n+1}. On the other hand, if the detector's outcome is chosen to be the first variable, say x_0, then the underlying influence network will be a complete graph.

This order sensitivity may at first seem paradoxical; d can be chosen arbitrarily, whereas people have fairly uniform conceptual structures, e.g., they agree on whether a pair of propositions are directly or indirectly related. The answer to this apparent paradox lies in the fact that the consensus about the structure of influence networks stems from the dominant role *causality* plays in the formation of these networks. In other words, the standard ordering imposed by the direction of causation indirectly induces identical topologies on the networks that people adopt for encoding experiential knowledge. It is tempting to speculate that, were it not for the social convention of adopting a standard ordering of events conforming to the flow of time and causation, human communication (as we now know it) would be impossible.

1.2. Conditional independence and graph separability

To facilitate the verification of dependencies among the variables in a Bayes network, we need to establish a clear correspondence between the topology of the network and various types of independence. Normally, independence between variables connotes lack of connectivity between their corresponding nodes. Thus, it would be ideal to require that, should the removal of some subset S of nodes from the network render nodes x_i and x_j disconnected, then such separation indicates genuine independence between x_i and x_j, conditioned on S:

$$P(x_i|x_j, S) = P(x_i|S) .$$

This would provide a clear graphical representation for the notion that x_j does not affect x_i directly but, rather, its influence is mediated by the variables in S. Unfortunately, a network constructed to satisfy this correspondence for any arbitrary S would normally fail to display an important class of independencies [24]. For example, in such a network, two variables which are marginally independent will appear directly connected, merely because there exists some other variable that depends on both.

Bayes' networks, on the other hand, allow representation of this class of independencies, but only at the cost of a slightly more complex criterion of separability, one which takes into consideration the directionality of the arrows in the graph. Consider a triplet of variables, x_1, x_2, x_3, where x_1 is connected to x_3 via x_2. The two links, connecting the pairs (x_1, x_2) and (x_2, x_3), can join at the midpoint, x_2, in one of three possible ways:

(1) tail-to-tail, $x_1 \leftarrow x_2 \rightarrow x_3$,

(2) head-to-tail, $x_1 \rightarrow x_2 \rightarrow x_3$ or $x_1 \leftarrow x_2 \leftarrow x_3$,

(3) head-to-head, $x_1 \rightarrow x_2 \leftarrow x_3$.

If we assume that x_1, x_2, x_3 are the only variables involved, it is clear from the method of constructing the network that, in cases (1) and (2), x_1 and x_3 are conditionally independent, given x_2, while in case (3), x_1 and x_3 are marginally independent (i.e., $P(x_3|x_1) = P(x_3)$) but may become dependent, given the value of x_2. Moreover, if x_2 in case (3) has descendants $x_4, x_5, \ldots$, then x_1 and x_3 may also become dependent if any one of those descendant variables is instantiated. These considerations motivate the definition of a qualified version of path connectivity, applicable to paths with directed links and sensitive to all the variables for which values are known at a given time.

Definition 1.1. (a) A subset of variables S_e is said to *separate* x_i from x_j if all paths between x_i and x_j are *separated* by S_e.

(b) A path P is *separated* by a subset S_e of variables if at least one pair of successive links along P is *blocked* by S_e.

We next introduce a nonconventional criterion under which a pair of converging arrows is said to be *blocked* by S_e.

Definition 1.2. (a) Two links meeting head-to-tail or tail-to-tail at node X are *blocked by* S_e if X is in S_e.

(b) Two links meeting head-to-head at node X are *blocked by* S_e if neither X nor any of its descendants is in S_e.

This modified definition of separation provides a graphical criterion for testing conditional independence: if S_e separates x_i from x_j, then x_i is conditionally independent of x_j, given S_e. The procedure involved in testing this modified criterion is slightly more complicated than the conventional test for deciding whether S_e is a separating cutset and can be handled by visual inspection. In Fig. 1, for example, one can easily verify that variables x_2 and x_3 are separated by $S_e = \{x_1\}$ or $S_e = \{x_1, x_4\}$ because the two paths between x_2 and x_3 are blocked by either one of these subsets. However, x_2 and x_3 are not separated by $S_e = \{x_1, x_6\}$ because x_6, as a descendant of x_5, "unblocks" the head-to-head connection at x_5, thus opening a pathway between x_2 and x_3.

Although the structure of Bayes' networks, together with the directionality of its links, depends strongly on the node ordering used in the network construction, conditional independence is a property of the underlying distribution and is, therefore, order-invariant. Thus, if we succeed in finding an ordering d in which a given conditional independence relationship becomes graphically transparent, that relationship remains valid even though it may not induce a graph-separation pattern in networks corresponding to other orderings. This permits the use of Bayes' networks for identifying by inspection a *screening neighborhood* for any given node, namely, a set S of variables that renders a given variable independent of every variable not in S. The separation criterion for Bayes' networks guarantees that the union of the following three types of neighbors is sufficient for forming a screening neighborhood: direct parents, direct successors and all direct parents of the latter. Thus, in a Markov chain, the screening neighborhood of any nonterminal node consists of its two immediate neighbors while, in trees, the screening neighborhood consists of the (unique) father and the immediate successors. In Fig. 1, however, the screening neighborhood of x_3 is $\{x_1, x_5, x_2\}$.

1.3. An outline and summary of results

The first part of this paper (Section 2) deals with the task of fusing and propagating the impacts of new evidence and beliefs through Bayesian networks in such a way that, when equilibrium is reached, each proposition will be assigned a certainty measure consistent with the axioms of probability theory. We first argue (Section 2.1) that any viable model of human reasoning should be able to perform this task by a self-activated propagation mechanism, i.e., by an array of simple and autonomous processors, communicating locally via the links provided by the belief network itself. In Section 2.2 we then show that these objectives can be fully realized in tree-structured networks, where each node has only one father. In Section 2.3 we extend the result to networks with multiple parents that are singly connected, i.e., there exists only one (undirected) path between any pair of nodes. In both cases, we identify belief parameters, communication messages and updating rules which guarantee that equilibrium is reached in time proportional to the longest path in the network and that, at equilibrium, each proposition will be accorded a belief measure consistent with probabilty theory. Several approaches toward achieving autonomous propagation in multiply connected networks are discussed in Section 2.4.

The second part of the paper (Section 3) expands on one of these approaches by examining the feasibility of preprocessing a belief network and turning it permanently into a tree by introducing dummy variables. In Section 3.1 we argue that such a technique mimics the way people develop causal models, that dummy variables correspond to the mental constructs known as "hidden causes" and that humans' relentless search for causal models is motivated by their desire to achieve computational advantages similar to those offered by tree-structured belief networks. After defining (in Section 3.2) the notions of star-decomposability and tree-decomposability, Section 3.3 treats triplets of propositional variables and asks under what conditions one is justified in attributing the observed dependencies to one central cause represented by a fourth variable. We show that these conditions are readily testable and that, when the conditions are satisfied, the parameters specifying the relations between the visible variables and the central cause can be uniquely determined. In Section 3.4 we extend these results to the case of a tree with n leaves. We show that, if there exists a set of dummy variables which decompose a given Bayes network into a tree, then the uniqueness of the triplets' decomposition enables us to configure that tree from pairwise dependencies among the variables. Moreover, the configuration procedure involves only $O(n \log n)$ steps. In Section 3.5 we evaluate the merits of this method and address the difficult issues of estimation and approximation.

2. Fusion and Propagation

2.1. Autonomous propagation as a computational paradigm

Once a belief network is constructed, it can be used to represent the generic knowledge of a given domain and can be consulted to reason about the interpretation of specific input data. The interpretation process involves instantiating a set of variables corresponding to the input data, calculating its impact on the probabilities of a set of variables designated as hypotheses and, finally, selecting the most likely combinations of these hypotheses. In general, this process can be carried out by an external interpreter which may have access to all parts of the network, may use its own computational facilities and may schedule its computational steps so as to take full advantage of the network topology with respect to the incoming data. However, the use of such an interpreter appears foreign to the reasoning process normally exhibited by humans [30]. Our limited short-term memory and narrow focus of attention, combined with our inability to shift rapidly between alternative lines of reasoning, suggests that our reasoning process is fairly local, progressing incrementally along pre-established pathways. Moreover, the speed and ease with which we perform some of the low-level interpretive functions, such as recognizing scenes, reading text and even understanding stories, strongly suggest that these processes involve a significant amount of parallelism, and that most of the processing is done *at the knowledge level* itself, not external to it.

A paradigm for modeling such phenomena would be to view an influence network not merely as a passive parsimonious code for storing factual knowledge but also as a computational architecture for reasoning about that knowledge. That means that the links in the network should be treated as the only pathways and activation centers that direct and propel the flow of data in the process of querying and updating beliefs. Accordingly, we assume that each node in the network is designated a separate processor, which both maintains the parameters of belief for the host variable and manages the communication links to and from the set of neighboring, conceptually related, variables. The communication lines are assumed to be open at all times, i.e., each processor may, at any time, interrogate the belief parameters associated with its neighbors and compare them to its own parameters. If the compared quantities satisfy some local constraints, no activity takes place. However, if any of these constraints are violated, the responsible node is activated to set its violating parameter straight. This, of course, will activate similar revisions at the neighboring nodes and will set up a multidirectional propagation process, until equilibrium is reached.

The main reason for this distributed message-passing paradigm is that it leads to a "transparent" revision process, in which the intermediate steps can be given an intuitively meaningful interpretation. Since a distributed process restricts each computational step to obtain inputs only from neighboring, semantically related variables, and since the activation of these steps proceeds along semantically familiar pathways, people find it easy to give meaningful interpretation to the individual steps, thus establishing confidence in the final result. Additionally, it is possible to generate qualitative justifications mechanically by tracing the sequence of operations along the activated pathways and giving them causal or diagnostic interpretations using appropriate verbal expressions.

The ability to update beliefs by an autonomous propagation mechanism also has a profound effect on sequential implementations of evidential reasoning. Of course, when this architecture is simulated on sequential machines, the notion of autonomous processors working simultaneously in time is only a metaphor; however, it signifies the complete separation of the stored knowledge from the control mechanism—the proclaimed, yet rarely achieved, goal of rule-based architectures. This separation guarantees the ultimate flexibility for a sequential controller; the computations can be performed in any order, without the need to remember or verify which parts of the network have or have not already been updated. Thus, for example, belief updating may be activated by changes occurring in logically related propositions, by requests for evidence arriving from a central supervisor, by a predetermined schedule or entirely at random. The communication and interaction among individual processors can be simulated using a blackboard architecture [17], where each proposition is designated specific areas of memory to access and modify. Additionally, the uniformity of this propagation scheme renders it natural for formulation in object-oriented languages: each node is an object of the same generic type, and the belief parameters are the messages by which interacting objects communicate.

In AI, constraint-propagation mechanisms have been found essential in several applications, e.g., vision [27, 35] and truth maintenance [20]. However, their use in evidential reasoning has been limited to non-Bayesian formalisms (e.g. [19, 30]). There have been several reasons for this.

First, the conditional probabilities characterizing the links in the network do not seem to impose definitive constraints on the probabilities that can be assigned to the nodes. The quantifier $P(A|B)$ only restricts the belief accorded to A in a very special set of circumstances, namely, when B is known to be true with absolute certainty and when no other evidential data is available. Under normal circumstances, all internal nodes in the network will be subject to some uncertainty and, more seriously, after the arrival of evidence e, the posterior beliefs in A and B are no longer related by $P(A|B)$ but by $P(A|B, e)$, which may be totally different. The result is that any arbitrary assignment of beliefs to propositions A and B can be consistent with the value of $P(A|B)$ initially assigned to the link connecting them; in other words, among these parameters, no violation of constraint can be detected locally.

Next, the difference between $P(A|B, e)$ and $P(A|B)$ suggests that the weights on the links should not remain fixed but should undergo constant adjustment as new evidence arrives. Not only would this entail enormous computational overhead, but it would also obliterate the advantages normally associated with propagation through fixed networks of constraints.

Finally, the fact that evidential reasoning involves both top-down (predictive) and bottom-up (diagnostic) inferences has caused apprehensions that, once we allow the propagation process to run its course unsupervised, pathological cases of instability, deadlock, and circular reasoning will develop [19]. Indeed, if a stronger belief in a given hypothesis means greater expectation for the occurrence of its various manifestations and if, in turn, a greater certainty in the occurrence of these manifestations adds further credence to the hypothesis, how can one avoid infinite updating loops when the processors responsible for these propositions begin to communicate with one another? Such apprehensions are not unique to probabilistic reasoning but should be considered in any hierarchical model of cognition where mutual reinforcement takes place between lower and higher levels of processing, e.g., connectionist models of reading [29] and language production [4].

This paper demonstrates that coherent and stable probabilistic reasoning *can* be accomplished by local propagation mechanisms while keeping the weights on the links constant throughout the process. This is made possible by characterizing the belief in each proposition by a *list* of parameters, each representing the degree of support the host proposition obtains from one of its neighbors. In the next two subsections we show that maintaining such a breakdown record of the sources of belief facilitates local updating of beliefs and that the network relaxes to a stable equilibrium, consistent with the axioms of probability theory, in time proportional to the network diameter. This record of parameters is also postulated as the mechanism which permits people to retrace reasoned assumptions for the purposes of modifying the model and generating explanatory arguments.

2.2. Belief propagation in trees

We shall first consider tree-structured influence networks, i.e., one in which every node, except one called "root," has only one incoming link. We allow each node to represent a multivalued variable which may represent a collection of mutually exclusive hypotheses (e.g., identity of organism: ORG_1, $\text{ORG}_2, \ldots$) or a collection of possible observations (e.g. patient's temperature: high, medium, low). Let a variable be labeled by a capital letter, e.g., A, B, $C, \ldots$, and its possible values subscripted, e.g., $A_1, A_2, \ldots, A_n$. Each directed link $A \rightarrow B$ is quantified by a fixed conditional probability matrix, $M(B|A)$, with entries: $M(B|A)_{ij} = P(B_j|A_i)$. Normally, the directionality of the arrow designates A as the set of causal hypotheses and B as the set of consequences or manifestations for these hypotheses.

Example 2.1. Assume that in a certain trial there are three suspects, one of whom has definitely committed a murder, and that the murder weapon, showing some fingerprints, was later found by the police. Let A stand for the identity of the last user of the weapon, namely, the killer. Let B stand for the identity of the last holder of the weapon, i.e., the person whose fingerprints were left on the weapon, and let C represent the possible readings that may be obtained in a fingerprint-testing laboratory.

The relations between these three variables would normally be conceptualized by the chain $A \rightarrow B \rightarrow C$; A generates expectations about B, and B generates expectations about C, but A has no influence on C once we know the value of B.

To represent the common-sense knowledge that, under normal circumstances, the killer is expected to be the last to hold the weapon, we may use the 3×3 conditional probability matrix:

$$P(B_j|A_i) = \begin{cases} 0.80, & \text{if } A_i = B_j,\ i, j = 1, 2, 3, \\ 0.10, & \text{if } A_i \neq B_j,\ i, j = 1, 2, 3. \end{cases}$$

To represent the reliability of the laboratory test, we use a matrix $P(C_k|B_j)$, satisfying

$$\sum_k P(C_k|B_j) = 1 \quad \text{for all } j\,.$$

Each entry in this matrix represents an if-then rule of the type:

> *If* the fingerprint is of suspect B_j *then* expect reading of the type C_k, with certainty $P(C_k|B_j)$

Note that this rule convention is at variance with that used in many expert systems (e.g., MYCIN), where rules point from evidence to hypothesis (e.g., if symptom, then disease), thus denoting a flow of mental inference. By contrast, the arrows in Bayes' networks point from causes to effects or from conditions to consequence, thus denoting a flow of constraints in the physical world. The reason for this choice is that people often prefer to encode experiential knowledge in causal schemata [34] and, as a consequence, rules expressed in causal format are assessed more reliably.[2]

Incoming information may be of two types: *specific evidence* and *virtual evidence*. Specific evidence corresponds to direct observations which validate, with certainty, the values of some variables in the network. Virtual evidence corresponds to judgments based on undisclosed observations which affect the belief in some variables in the network. Such evidence is modeled by dummy nodes, representing the undisclosed observations, connected by unquantified (dummy) links to the variables affected by the observations. These links will carry only one-way information, from the evidence to the variables affected by it, but not vice versa. For example, if it is impractical for the fingerprint laboratory to disclose all possible readings (in variable C) or if the laboratory chose to base its finding on human judgment, C will be represented by a dummy node, and the link $B \rightarrow C$ will specify the relative degree to which each suspect is believed to be the owner of the fingerprint pattern examined. For example, the laboratory examiner may issue a report in the form of a list,

$$P(C_{\text{observed}} | B) = (0.80, 0.60, 0.50) ,$$

stating that he/she is 80% sure that the fingerprint belongs to suspect B_1, 60% sure that it belongs to B_2 and 50% sure that it belongs to B_3. Note that these numbers need not sum up to unity, thus permitting each judgment to be formed independently of the other, separately matching each suspect's fingerprints to those found on the weapon.

All incoming evidence, both specific and virtual, will be denoted by D to connote *data*, and will be treated by instantiating the variables corresponding to the evidence. For the sake of clarity, we will distinguish between the fixed conditional probabilities that label the links, e.g., $P(A|B)$, and the dynamic values of the updated node probabilities. The latter will be denoted by $\text{BEL}(A_i)$, which reflects the overall belief accorded to proposition $A = A_i$ by all data so far received. Thus,

[2] It appears that, by and large, frames used to index human memory are organized to evoke *expectations* rather than *explanations*. The reason could, perhaps, be attributed to the fact that expectation-evoking frames normally consist of more stable relationships. For example, $P(B_j|C_k)$ in Example 2.1 would vary drastically with the proportion of people who have type B_j fingerprints. $P(C_k|B_j)$, on the other hand, depends merely on the similarity between the type of fingerprint that suspect B_j has and the readings observed in the lab; it is perceived to be a stable local property of the laboratory procedure, independent of other information regarding suspect B_j.

$$\mathrm{BEL}(A_i) \stackrel{\Delta}{=} P(A_i|D)$$

where D is the value combination of all instantiated variables.

Consider the fragment of a tree, as depicted in Fig. 2. The belief in the various values of B depends on three distinct sets of data: i.e., data from the tree rooted at B, from the tree rooted at C and from the tree above A. However, since A separates B from all variables except B's decendants (see Section 1.2), the influence of the latter two sources of information on B are completely summarized by their combined effect on A. More formally: let D_B^- stand for the data contained in the tree rooted at B and D_B^+ for the data contained in the rest of the network. We have

$$P(B_j|A_i, D_B^+) = P(B_j|A_i) \tag{1}$$

which also leads to the usual "intersiblings" conditional independence:

$$P(B_j, C_k|A_i) = P(B_j|A_i) \cdot P(C_k|A_i)\,, \tag{2}$$

since the proposition $C = C_k$ is part of D_B^+.

2.2.1. *Data fusion*

Assume we wish to find the belief induced on B by some data $D = D_B^- \cup D_B^+$. Bayes' theorem, together with (1), yields the product rule

$$\mathrm{BEL}(B_i) = P(B_i|D_B^+, D_B^-) = \alpha P[D_B^-|B_i] \cdot P[B_i|D_B^+]\,, \tag{3}$$

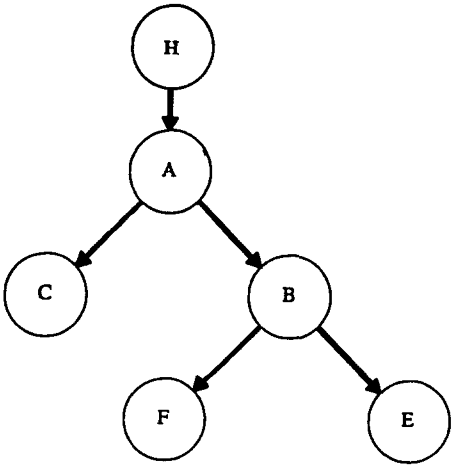


FIG. 2. A segment of a tree illustrating data partitioning.

where α is a normalizing constant. This is a generalization of the celebrated Bayes formula for binary variables

$$O(H|E) = \lambda(E)O(H) , \tag{4}$$

where $\lambda(E) = P(E|H)/P(E|\bar{H})$ is known as the *likelihood ratio* and $O(H) = P(H)/P(\bar{H})$ as the *prior odds* [6].

As an example, let D_B^- represent the experience of examining the fingerprints left on the murder weapon, and let D_B^+ stand for all other testimonies heard in the trial. $P(B_i|D_B^+)$ would then stand for our prior (before examining the fingerprints) belief that the ith suspect was the last to hold the weapon, and $P(D_B^-|B_i)$ would represent the report issued by the fingerprint laboratory. Taking, as before, $P(D_B^-|B) = (0.80, 0.60, 0.50)$, and assuming we have $P(B|D_B^+) = (0.60, 0.30, 0.10)$, our total belief in the assertions $B = B_i$ is given by

$$\begin{aligned}\mathrm{BEL}(B) &= \alpha P(D_B^-|B)P(B|D_B^+)\\ &= \alpha(0.80, 0.60, 0.50)(0.60, 0.30, 0.10)\\ &= \alpha(0.48, 0.18, 0.05)\end{aligned}$$

and, to properly normalize $\mathrm{BEL}(B)$, we set $\alpha = (0.48 + 0.18 + 0.05)^{-1}$ and obtain $\mathrm{BEL}(B) = (0.676, 0.254, 0.07)$.

Equation (3) generalizes (4) in two ways. First, it permits the treatment of nonbinary variables where the mental task of estimating $P(E|\bar{H})$ is often unnatural and where conditional independence with respect to the negations of the hypotheses is normally violated (i.e., $P(E_1, E_2|\bar{H}) \neq P(E_1|\bar{H})P(E_2|\bar{H})$). Second, it identifies a surrogate to the prior probability term for every intermediate node in the tree, even *after* obtaining some evidential data.

In ordinary Bayesian updating of sequential data, it is often possible to recursively use the posterior odd as a new prior for computing the impact of the next item of evidence. However, this method works only when the items of evidence are mutually independent conditioned on the updated hypothesis, H, and will not be applicable to network updating because only variables which are separated from each other by H are guaranteed to be conditionally independent, given H. In general, therefore, it is not permissible to use the total posterior belief, updated by (3), as a new multiplicative prior for the calculation. Thus, the significance of (3) lies in showing that a product rule analogous to (4) can be applied to any node in the network without requiring a separate prior probability assessment. However, the multiplicative role of the prior probability has been taken over by that portion of belief contributed by evidence from the subtree *above* the updated variable, i.e., excluding the data collected from its descendants. The root is the only node which requires a prior probability estimation, and since it has no network above, D^+_{root} should be interpreted as the background knowledge which remains unexplicated.

Equation (3) suggests that the probability distribution of every variable in the network can be computed if the node corresponding to that variable contains the parameters

$$\lambda(B_i) = P(D^-_B|B_i) \tag{5}$$

and

$$\pi(B_i) = P(B_i|D^+_B)\,. \tag{6}$$

$\pi(B_i)$ represents the causal or *anticipatory* support attributed to B_i by the ancestors of B, and $\lambda(B_i)$ represents the diagnostic or *retrospective* support B_i receives from B's descendants. The total strength of belief in B_i would be obtained by *fusing* these two supports via the product

$$\text{BEL}(B_i) = \alpha\lambda(B_i)\pi(B_i)\,. \tag{7}$$

While two parameters, $\lambda(E)$ and $O(H)$, were sufficient for binary variables, an n-valued variable needs to be characterized by two n-tuples:

$$\lambda(B) = \lambda(B_1), \lambda(B_2), \ldots, \lambda(B_n)\,, \tag{8}$$

$$\pi(B) = \pi(B_1), \pi(B_2), \ldots, \pi(B_n)\,. \tag{9}$$

To see how information from several decendants fuse at node B, note that the data D^-_B in (5) can be partitioned into disjoint subsets, D^{1-}, $D^{2-}, \ldots, D^{m-}$, one for each subtree emanating from (the m children of) B. Since B "separates" these subtrees, conditional independence holds:

$$\lambda(B_i) = P(D^-_B|B_i) = \prod_k P(D^{k-}|B_i)\,, \tag{10}$$

so $\lambda(B_i)$ can be formed as a product of the terms $P(D^{k-}|B_i)$ if these are delivered to processor B as messages from its children. For instance if in our fingerprint example $P(D^{1-}|B) = (0.80, 0.60, 0.50)$ and $P(D^{2-}|B) = (0.30, 0.50, 0.90)$ represent two reports issued by two independent laboratories, then the overall diagnostic support $\lambda(B)$ attributable to the three possible states of B is

$$\lambda(B) = (0.80, 0.60, 0.50) \cdot (0.30, 0.50, 0.90) = (0.24, 0.30, 0.45)\,.$$

This, combined with the previous causal support $\pi(B) = (0.60, 0.30, 0.10)$, yields an overall belief of

$$\mathrm{BEL}(B) = \alpha(0.24, 0.30, 0.45)(0.60, 0.30, 0.10)$$

$$= (0.516, 0.322, 0.161)\,.$$

Thus, we see that, at each node of a Bayes tree, the fusion of all incoming data is purely multiplicative.

2.2.2. *Propagation mechanism*

Assuming that the vectors λ andπ are stored with each node of the network, our task is now to determine how the influence of new information will spread through the network, namely, how the parameters π and λ of a given node can be determined from the π's and λ's of its neighbors. This is done easily by conditioning (5) and (6) on all the values that the neighbors can assume. For example, suppose E is the kth son of B. To compute the kth multiplicand in the product of (10) from the value of $\lambda(E)$, we write

$$P(D^{k-}|B_i) = \sum_j P(D_E^-|B_i, E_j)P(E_j|B_i)$$

and obtain (using (1) and (5))

$$P(D^{k-}|B_i) = \sum_j \lambda(E_j)P(E_j|B_i)\,.$$

Thus, $P(D^{k-}|B_i)$ is obtained by taking the λ-vector stored at the kth son of B and multiplying it by the fixed conditional-probability matrix that quantifies the link between B and E. Thus, the λ-vector of each node can be computed from the λ's of its children by multiplying the latter by their respective link matrices and then multiplying the resultant vectors together, term-by-term, as shown in (10). Each multiplicand $P(D^{k-}|B)$ would be treated as a *message* sent by the kth son of B and, if the sending variable is named E, the message will be denoted by $\lambda_E(B)$,

$$\lambda_E(B_i) = \sum_j P(E_j|B_i)\lambda(E_j)\,.$$

A similar analysis, applied to the vector π, shows that the π of any node can be computed from the π of its father and the λ's of its siblings, again after multiplication by the corresponding link matrices. No direct communication with the siblings is necessary since the information required of them already resides at the father's site (for the purpose of calculating its λ, as in (10)) and can be sent down to the requesting son. This can be shown by conditioning $\pi(B)$ over the values of the parent A:

$$\begin{aligned}\pi(B_i) &= P(B_i|D^+(B))\\ &= \sum_j P(B_i|A_j, D^+(B))P(A_j|D^+(B))\\ &= \sum_j P(B_i|A_j), P(A_j|\text{all data excluding } D^-(B))\\ &= \sum_j P(B_i|A_j)\left[\alpha\pi(A_j)\prod_m \lambda_m(A_j)\right]\end{aligned}$$

with m ranging over the siblings of B. The expression in the brackets contains parameters available to processor A, and it can be chosen, therefore, as the message $\pi_B(A)$ that A transmits to B.

Thus,

$$\pi(B_i) = \sum_j P(B_i|A_j)\pi_B(A_j)\,, \tag{11}$$

where

$$\pi_B(A_j) = \alpha\pi(A_j) \prod_{m:\text{ sibling of } B} \lambda_m(A_j)\,, \tag{12}$$

or, alternatively,

$$\pi_B(A_j) = \alpha' \frac{\text{BEL}(A_j)}{\lambda_B(A_j)}\,. \tag{13}$$

The division by $\lambda_B(A)$ amounts to removing from BEL(A) the contribution of D_B^- as dictated by the definition of π in (6).

These results lead to the following propagation scheme:

Step 1. When processor B is activated to update its parameters, it simultaneously inspects the $\pi_B(A)$ message communicated by the father A and the messages $\lambda_1(B)$ $\lambda_2(B), \ldots,$ communicated by each of its sons. Using these inputs, it then updates its λ and π as follows:

Step 2. λ is computed using a term-by-term multiplication of the vectors λ_1, $\lambda_2, \ldots,$ (as in (10)):

$$\lambda(B_i) = \lambda_1(B_i) \times \lambda_2(B_i) \times \cdots = \prod_k \lambda_k(B_i)\,.$$

Step 3. π is computed using:

$$\pi(B_i) = \beta \sum_j P(B_i|A_j)\pi_B(A_j)\,,$$

where β is a normalizing constant and $\pi_B(A)$ is the last message sent to B from the father A.

Step 4. Using the messages received, together with the updated values of λ and π, each processor then computes new π- and λ-messages to be posted on the message boards reserved for its sons and its father, respectively. These are computed as follows:

Step 5. *Bottom-up propagation.* The new message $\lambda_B(A)$ that B sends to its father (A) is computed by

$$\lambda_B(A_j) = \sum_i P(B_i | A_j)\lambda(B_i) .$$

Step 6. *Top-down propagation.* The new message $\pi_E(B)$ that B sends to its kth child E is computed by

$$\pi_E(B_i) = \alpha\pi(B_i) \prod_{m \neq k} \lambda_m(B_i) ,$$

or, alternatively,

$$\pi_E(B_i) = \alpha' \frac{\mathrm{BEL}(B_i)}{\lambda_E(B_i)} .$$

This updating scheme is shown schematically in Fig. 3, where multiplications of any two vectors stand for term-by-term operations. There is no need, of course, to normalize the π-messages prior to transmission (only the BEL($\cdot$) expressions actually require normalization). This is done solely for the purpose of retaining the probabilistic meaning of these messages. Additional economy can be achieved by having each node B transmit a single message BEL(B) to all its children and letting each child use (13) to uncover its appropriated π-message.

Terminal and data nodes in the tree require special treatments. Here we have to distinguish several cases:

(1) *Anticipatory node*, a leaf node that has not been instantiated yet: For such variables, BEL should be equal to π and, therefore, we should set $\lambda = (1, 1, \ldots, 1)$.

(2) *Data node*, a variable with instantiated value: Following (5) and (6), if the jth state of B were observed to be true, we set $\lambda = \pi = (0, \ldots, 0, 1, 0, \ldots, 0)$ with 1 at the jth position.

(3) *Dummy node*, a node B representing virtual or judgmental evidence bearing on A: We do not specify $\lambda(B)$ or $\pi(B)$ but, instead, post a $\lambda_B(A)$ message to A, where $\lambda_B(A_i) = K \cdot P(\text{observation}|A_i)$, and K is any convenient constant.

(4) *Root node*: The boundary condition for the root node is established by setting $\pi(\text{root})$ = prior probability of the root variable.

Example 2.2. To illustrate these computations let us return to Example 2.1, and let us assume that based on all testimonies heard so far, our belief in the

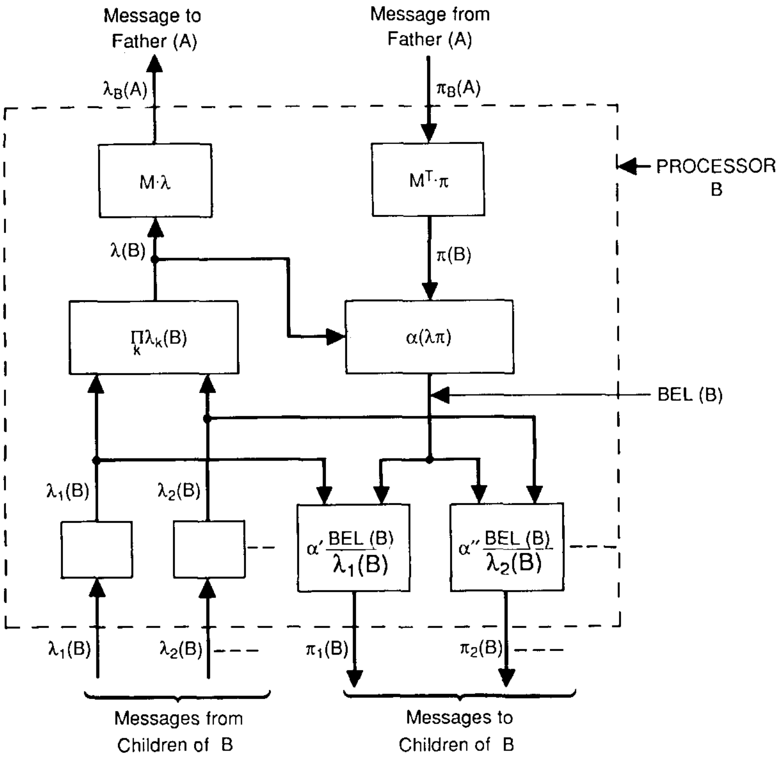


FIG. 3. The internal structure of a single processor performing belief updating for variable B.

identity of the killer amounts to $\pi(A) = (0.8, 0.1, 0.1)$. Before obtaining any fingerprint information, B is an anticipatory node with $\lambda(B) = (1, 1, 1)$, which also yields $\lambda_B(A) = \lambda(A) = (1, 1, 1)$ and $\mathrm{BEL}(A) = \pi(A)$. $\pi(B)$ can be calculated from (13) (using $\pi_B(A) = \pi(A)$ and $P(B_i|A_j) = 0.8$ if $i = j$), yielding

$$\pi(B) = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \begin{bmatrix} 0.8 \\ 0.1 \\ 0.1 \end{bmatrix} = (0.66, 0.17, 0.17) = \mathrm{BEL}(B)\,.$$

Now assume that a laboratory report arrives, summarizing the test results (a virtual evidence C) by the message $\lambda_C(B) = \lambda(B) = (0.80, 0.60, 0.50)$. Node B updates its belief to read:

$$\begin{aligned} \mathrm{BEL}(B) = \alpha\lambda(B)\pi(B) &= \alpha(0.80, 0.60, 0.50)(0.66, 0.17, 0.17) \\ &= (0.738, 0.142, 0.119) \end{aligned}$$

and computes a new message, $\lambda_B(A)$, for A:

$$\lambda_B(A) = M \cdot \lambda = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \begin{bmatrix} 0.8 \\ 0.6 \\ 0.5 \end{bmatrix} = (0.75, 0.61, 0.54)\,.$$

Upon receiving this message, node A sets $\lambda(A) = \lambda_B(A)$ and recomputes its belief to

$$\begin{aligned} \mathrm{BEL}(A) &= \alpha\lambda(A)\pi(A) = \alpha(0.75, 0.61, 0.54)(0.8, 0.1, 0.1) \\ &= (0.84, 0.085, 0.076)\,. \end{aligned}$$

Now assume that suspect A_1 produces a very strong alibi in his favor, suggesting that there are only 1 : 10 odds that he could have committed the crime. To fuse this information with all previous evidence, we link a new virtual-evidence node E directly to A and post the message $\lambda_E(A) = (0.10, 1.0, 1.0)$ on the link. $\lambda_E(A)$ combines with $\lambda_B(A)$ to yield

$$\begin{aligned} \lambda(A) &= \lambda_E(A)\lambda_B(A) = (0.075, 0.61, 0.54)\,, \\ \mathrm{BEL}(A) &= \alpha(A)\pi(A) \\ &= \alpha(0.075, 0.061, 0.54)(0.84, 0.85, 0.076) \\ &= (0.404, 0.333, 0.263) \end{aligned}$$

and generates the message $\pi_B(A) = \alpha\lambda_E(A)\pi(A) = \alpha(0.08, 0.1, 0.1)$ to B. Upon receiving $\pi_B(A)$, processor B updates its causal support $\pi(B)$ to read:

$$\pi(B) = \alpha' \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \begin{bmatrix} 0.08 \\ 0.10 \\ 0.10 \end{bmatrix} = (0.30.\ 0.35, 0.35)$$

and $\mathrm{BEL}(B)$ becomes

$$\begin{aligned} \mathrm{BEL}(B) &= \alpha\lambda(B)\pi(B) \\ &= \alpha(0.8, 0.6, 0.5)(0.334, 0.343, 0.317) \\ &= (0.423, 0.326, 0.251)\,. \end{aligned}$$

The purpose of propagating beliefs top-down to sensory nodes such as B is two-fold—to guide data-acquisition strategies toward the most informative sensory nodes and to facilitate explanations which justify the system's inference steps.

Note that $\mathrm{BEL}(A)$ cannot be taken as an updated prior of A for the purpose of calculating $\mathrm{BEL}(B)$. In other words, it is wrong to update $\mathrm{BEL}(B)$ via the textbook formula

$$\mathrm{BEL}(B_i) = \sum_j P(B_i|A_j)\mathrm{BEL}(A_j) ,$$

also known as Jeffrey's rule [11], because BEL(A) itself was affected by information transmitted from B, and reflecting this information back to B would amount to counting the same evidence twice.

2.2.3. *Illustrating the flow of belief*

Figure 4 shows six successive stages of belief propagation through a simple binary tree, assuming that updating is triggered by changes in the belief parameters of neighboring processors. Initially (Fig. 4(a)), the tree is in equilibrium, and all terminal nodes are anticipatory. As soon as two data nodes are activated (Fig. 4(b)), white tokens are placed on their links, directed towards their fathers. In the next phase, the fathers, activated by these tokens, absorb them and manufacture the appropriate number of tokens for their neighbors (Fig. 4(c)): white tokens for their fathers and black ones for the children. (The links through which the absorbed tokens have entered do not receive new tokens, thus reflecting the feature that a π-message is not affected by a λ-message crossing the same link.) The root node now receives two white tokens, one from each of its descendants. That triggers the production of two black tokens for top-down delivery (Fig. 4(d)). The process continues in this fashion until, after six cycles, all tokens are absorbed, and the network reaches a new equilibrium.

As soon as a leaf node posts a token for its parent, it is ready to receive new data and, when this occurs, a new token is posted on the link, replacing the old

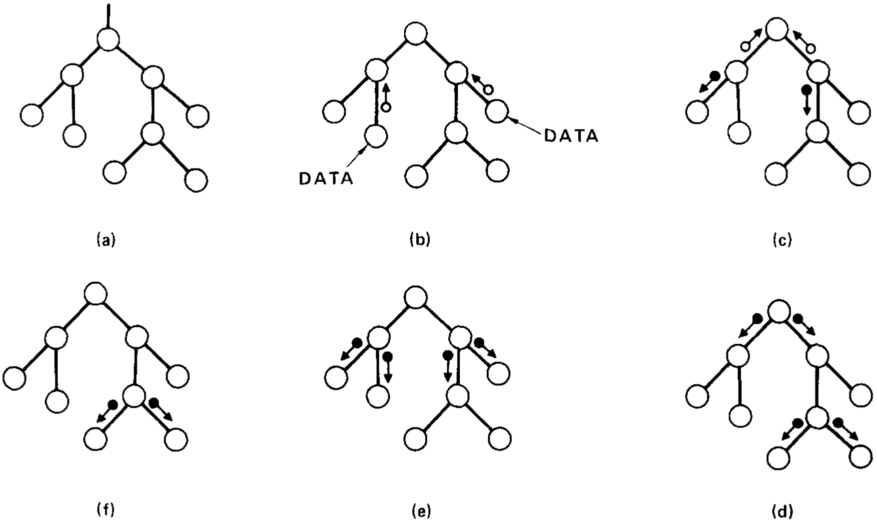


FIG. 4. The impact of new data propagates through a tree by a message-passing process.

one. In this fashion the inference network can also track a changing environment and provide coherent interpretation of signals emanating simultaneously from multiple sources.

2.2.4. *Properties of the updating scheme*

(1) The local computations required by the updating scheme are efficient in both storage and time. For an m-ary tree with n values per node, each processor should store $n^2 + mn + 2n$ real numbers and perform $2n^2 + mn + 2n$ multiplications per update.

(2) The local computations and the final belief distribution are entirely independent of the control mechanism that activates the individual operations. They can be activated by either data-driven or goal-driven (e.g., requests for evidence) control strategies, by a clock or at random.

(3) New information diffuses through the network in a single pass. Instabilities and indefinite relaxations have been eliminated by maintaining a two-parameter system (π and λ) to decouple causal support from diagnostic support. The time required for completing the diffusion (in parallel) is proportional to the diameter of the network.

2.3. Propagation in singly connected networks

The tree structures treated in the preceding section require that exactly one variable be considered a cause of any other variable. This restriction simplifies computations, but its representational power is rather limited since it forces us to group together all causal factors sharing a common consequence into a single node. By contrast, when people associate a given observation with multiple potential causes, they weigh one causal factor against another as independent variables, each pointing to a specialized area of knowledge. As an illustration, consider the following situation:

> Mr. Holmes received a phone call at work from his neighbor notifying him that she heard a burglar alarm sound from the direction of his home. As he is preparing to rush home, Mr. Holmes recalls that recently the alarm had been triggered by an earthquake. Driving home, he hears a radio newscast reporting an earthquake 200 miles away. [14]

Mr. Holmes perceives two episodes which may be potential causes for the alarm sound, an attempted burglary and an earthquake. Even though burglaries can safely be assumed independent of earthquakes, the radio announcement still reduces the likelihood of a burglary, as it "explains away" the alarm sound. Moreover, the causal events are perceived as individual variables each pointing to a separate frame of knowledge.

This nonmonotonic interaction among multiple causes is a prevailing pattern of human reasoning. When a physician discovers evidence in favor of one disease, it reduces the likelihood of other diseases, although the patient might well be suffering from two or more disorders simultaneously. The same maxim also governs the interplay of other frame-like explanations (not necessarily causal). For example, it is essential for comprehending sentences such as "John could not walk straight, and I thought he got drunk again. However, seeing the blood on his shirt, I knew it was a different matter."

This section extends the propagation scheme to graph structures which permit a node to have multiple parents and thus capture "sideways" interactions via common successors. However, the graphs are required to be *singly connected*, namely, one (undirected) path, at most, exists between any two nodes.

2.3.1. *Fusion equations*

Consider a fragment of a singly connected network, depicted in Fig. 5. The link $B \rightarrow A$ partitions the graph into two parts: an upper subgraph, G^+_{BA}, and a lower subgraph G^-_{BA}. These two graphs contain two sets of *data*, which we shall call D^+_{BA} and D^-_{BA}, respectively. Likewise, the links $C \rightarrow A$, $A \rightarrow X$, and $A \rightarrow Y$ define the subgraphs G^+_{CA}, G^-_{AX}, and G^-_{AY}, which contain the data sets D^+_{CA}, D^-_{AX} and D^-_{AY}, respectively. Since A is a common child of B and C, it does not separate G^+_{BA} from G^+_{CA}. However, it does separate the following three subgraphs: $G^+_{BA} \cup G^+_{CA}$, G^-_{AX} and G^-_{AY}, and we can write

$$P(D^-_{AX}, D^-_{AY} | A_i, D^+_{BA}, D^+_{CA}) = P(D^-_{AX}|A_i)P(D^-_{AY}|A_i)\,. \qquad (14)$$

Thus, using Bayes' rule, the overall strength of belief in A_i can be written:

$$\begin{aligned} \mathrm{BEL}(A_i) &= P(A_i|D^+_{BA}, D^+_{CA}, D^-_{AX}, D^-_{AY}) \\ &= \alpha P(A_i|D^+_{BA}, D^+_{CA})P(D^-_{AX}|A_i)P(D^-_{AY}|A_i)\,, \end{aligned} \qquad (15)$$

where α is a normalizing constant. By further conditioning over the values of B and C (see Appendix A), we get:

$$\begin{aligned} \mathrm{BEL}(A_i) = {} & \alpha P(D^-_{AX}|A_i)P(D^-_{AY}|A_i) \\ & \cdot \Big[\sum_{jk} P(A_i|B_j, C_k)P(B_j|D^+_{BA})P(C_k|D^+_{CA})\Big]\,. \end{aligned} \qquad (16)$$

Equation (16) shows that the probability distribution of each variable A in the network can be computed if three types of parameters are made available: (1) the current strength of the causal support, π, contributed by each incoming link to A:

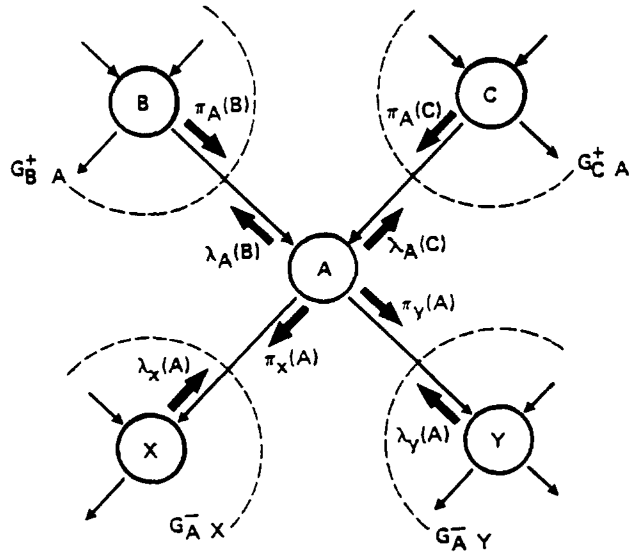


FIG. 5. Fragment of a singly connected network with multiple parents, illustrating data partitioning and belief parameters.

$$\pi_A(B_j) = P(B_j|D_{BA}^+)\,, \tag{17}$$

(2) the current strength of the diagnostic support, λ, contributed by each outgoing link from A:

$$\lambda_X(A_i) = P(D_{AX}^-|A_i)\,, \tag{18}$$

and (3) the fixed conditional-probability matrix, $P(A|B, C)$, which relates the variable A to its immediate causes. Accordingly, we let each link carry two dynamic parameters, π and λ, and let each node store an encoding of $P(A|B, C)$.

With these parameters at hand, the fusion equation (16) becomes

$$\mathrm{BEL}(A_i) = \alpha\lambda_X(A_i)\lambda_Y(A_i)\sum_{jk} P(A_i|B_j, C_k)\pi_A(B_j)\pi_A(C_k)\,. \tag{19}$$

Alternatively, from two parameters, π and λ, residing on the same link, we can compute the belief distribution of the parent node by the product

$$\mathrm{BEL}(B_j) = \alpha\pi_A(B_j)\lambda_A(B_j)\,. \tag{20}$$

2.3.2. *Propagation equation*

Assuming that the vectors π and λ are stored with each link, our task is now to prescribe how the influence of new information should spread through the network.

Updating λ

Starting from the definition of $\lambda_A(B_i) = P(D^-_{BA}|B_i)$, we partition the data D^-_{BA} into its components: A, D^-_{AX}, D^-_{AY}, and D^+_{CA}, and summing over all values of A and C (see Appendix A), we get:

$$\lambda_A(B_i) = \alpha \sum_j \left[\pi_A(C_j) \sum_k \lambda_X(A_k)\lambda_Y(A_k)P(A_k|B_i, C_j) \right] . \tag{21}$$

Equation (21) shows that only three parameters (in addition to the conditional probabilities $P(A|B, C)$) are needed for updating the diagnostic parameter vector $\lambda_A(B)$: $\pi_A(C)$, $\lambda_X(A)$, and $\lambda_Y(A)$. This is expected since D^-_{BA} is completely summarized by X, Y, and C.

Updating π

Similar manipulation on (17) (see Appendix A) yields the following rule for updating the causal parameter $\pi_X(A)$:

$$\pi_X(A_i) = \alpha\lambda_Y(A_i)\left[\sum_{jk} P(A_i|B_j, C_k)\pi_A(B_j)\pi_A(C_k) \right] . \tag{22}$$

Thus, $\pi_X(A)$, like $\lambda_A(B)$, is also determined by three neighboring parameters: $\lambda_Y(A)$, $\pi_A(B)$, and $\pi_A(C)$.

Equations (21) and (22) demonstrate that a perturbation of the causal parameter π will not affect the diagnostic parameter λ on the same link, and vice versa. The two are orthogonal to each other since they depend on two disjoint sets of data. Therefore, any perturbation of beliefs due to new evidence propagates through the network and is absorbed at the boundary without reflection. A new state of equilibrium will be reached after a finite number of updates which, in the worst case, would be equal to the diameter of the network.

Equation (21) also reveals that if no data are observed below A (i.e., all λ's pointing to A are unit vectors), then all λ's emanating from A are unit vectors. This means that evidence gathered at a particular node does not influence its spouses until their common son gathers diagnostic support. This reflects the special connectivity conditions established in Section 1.2 and matches our intuition regarding multiple causes. In Mr. Holmes' case, for example, prior to the neighbor's telephone call, seismic data indicating an earthquake would not have influenced the likelihood of a burglary.

Although the treatment in this paper is restricted to discrete variables, (21) and (22) can be readily extended to handle continuous variables as well. The case of additive Gaussian variables is particularly attractive because all belief distributions and all the π- and λ-messages can be characterized by only two parameters each, the mean and the variance. Thus, the computations required are simpler, and matrix manipulations are avoided [23]. Distributed updating of noncausal, object-class hierarchies is described in [25].

2.4. Summary and extensions for multiply connected networks

The preceding two sections show that the architectural objectives of propagating beliefs coherently through an active network of primitive, identical, and autonomous processors can be fully realized in singly connected graphs. Instabilities due to bidirectional inferences are avoided by using multiple, source-identified belief parameters, and equilibrium is guaranteed to be reached in time proportional to the network diameter.

The primitive processors are simple and repetitive, and they require no working memory except that used in matrix multiplications. Thus, this architecture lends itself naturally to hardware implementation, capable of real-time interpretation of rapidly changing data. It also provides a reasonable model of neural nets involved in such cognitive tasks as visual recognition, reading comprehension [28] and associative retrieval [1], where unsupervised parallelism is an uncontested mechanism.

It is also interesting to note that the marginal conditional probabilities on the links of the network remain constant and retain their viability throughout the updating process. This is important because having to adjust the weights each time new data arrives would be computationally prohibitive. The stable viability of the marginal conditional probabilities may explain why people can assess the magnitude of these relationships better than those of any other probabilistic quantity. Apparently, these relationships have been chosen as the standard primitives for organizing and quantifying probabilistic knowledge in our long-term memory.

The efficacy of singly connected networks in supporting autonomous propagation raises the question of whether similar propagation mechanisms can operate in less restrictive networks (like the one in Fig. 1), where multiple parents of common children may possess common ancestors, thus forming loops in the underlying network. If we ignore the existence of loops and permit the nodes to continue communicating with each other as if the network were singly connected, messages may circulate indefinitely around these loops, and the process will not converge to the correct state of equilibrium.

A straightforward way of handling the network of Fig. 1 would be to appoint a local interpreter for the loop x_1, x_2, x_3, x_5 that will account for the interactions between x_2 and x_3. This amounts, basically, to collapsing nodes x_2 and x_3 into a single node representing the compound variable (x_2, x_3). This method works well on small loops [32], but as soon as the number of variables exceeds 3 or 4, compounding requires handling huge matrices and masks the natural conceptual structure embedded in the original network.

A second method of propagation is based on "stochastic relaxation" [8] similar to that used by Boltzman machines [9]. Each processor examines the states of the variables within its screening neighborhood, computes a belief distribution for the values of its host variable, then randomly selects one of these values with probability given by the computed distribution. The value chosen will subsequently be interrogated by the neighbors upon computing their beliefs, and so on. This scheme is guaranteed convergence, but it usually requires very long relaxation times before reaching a steady state.

A third method called *conditioning* [22] is based on our ability to change the connectivity of a network and render it singly connected by instantiating a selected group of variables. In Fig. 1, for example, instantiating x_1 to some value would block the pathway x_2, x_1, x_3, and would render the rest of the network singly connected, so that the propagation techniques of the preceding section would be applicable. Thus, if we wish to propagate the impact of an observed datum, say at x_6, to the entire network, we first assume $x_1 = 0$, propagate the impact of x_6 to the variables $x_2, \ldots, x_5$, repeat the propagation under the assumption $x_1 = 1$ and, finally, sum the two results weighted by the posterior probability $P(x_1|x_6)$. It can also be executed in parallel by letting each node receive, compute, and transmit several sets of parameters, one for each value of the conditioning variable(s). Conditioning provides a working solution in most practical cases, but it occasionally suffers from the inevitable combinatorial explosion—the number of messages may grow exponentially with the number of nodes required for breaking up all loops in the network.

The use of conditioning to facilitate propagation is not foreign to human reasoning. When we find it hard to estimate the likelihood of a given outcome, we often make hypothetical *assumptions* that render the estimation simpler and then negate the assumptions to see if the results do not vary substantially. One of the most pervasive patterns of plausible reasoning is the maxim that, if two diametrically opposed assumptions impart two different degrees of confidence onto a proposition Q, then the unconditional degree of confidence merited by Q should be somewhere between the two. The terms "hypothetical" or "assumption-based" reasoning, "reasoning by cases," and "envisioning" all refer to the same basic mechanism of selecting a key variable, binding it to some of its values, deriving the consequences of each binding separately, and integrating those consequences together.

Finally, a preprocessing approach, which is discussed more fully in Section 3, introduces auxiliary variables and permanently turns the network into a tree. To understand the basis of this method, consider, for example, the tree of Fig. 2. The variables C, H, E, F are tightly coupled in the sense that no two of them can be separated by the others; therefore, if we were to construct a Bayesian network based on these variables *alone*, a complete graph would ensue. Yet, together with the intermediate variables A and B the interactions among the leaf variables are tree-structured, clearly demonstrating that some multiply connected networks can inherit all the advantages of tree representations by the introduction of a few dummy variables. In some respects, this method is similar to that of appointing external interpreters to handle nonseparable components of the graph, because the processors assigned to the dummy variables, like the external interpreters, serve no other function but that of mediation among the real variables. However, the dummy-variables scheme enjoys the added advantage of uniformity: the processors representing the dummy variables can be idential to those representing the real variables, in full compliance with our architectural objectives. Moreover, there are strong reasons to believe that the process of reorganizing data structures by adding fictitious variables mimics an important component of conceptual development in human beings—the evolution of causal models. These considerations are discussed in the section that follows.

3. Structuring Causal Trees

3.1. Causality, conditional independence, and tree architecture

Human beings exhibit an almost obsessive urge to conceptually mold empirical 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 states, social attitudes, beliefs, goals, plans, and intentions. Medical knowledge, likewise, is organized into causal hierarchies of invading organisms, physical disorders, complications, clinical states and, only finally, the visible symptoms.

We take the position that human obsession with causation, like many other psychological compulsions, is computationally motivated. Causal models are attractive only because they provide effective data structures for representing empirical knowledge—they can be queried and updated at high speed with minimal external supervision; so, it behooves us to take a closer look at the structure of causal models and determine what it is that makes them so effective. In other words, what are the computational assets of those fictitious variables called "causes" that make them worthy of such relentless human pursuit, and what renders causal explanations so pleasing and comforting, once they are found?

The paradigm expounded in this paper is that the main ingredient responsible for the pervasive role of causal models is their *centrally organized architecture*, i.e., an architecture in which dependencies among variables are mediated by one central mechanism.

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 dealing with 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 than one which entails intervariable communication. 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 a change in any of the variables can communicate its impact to all other variables in only two steps.

Computationally speaking, such causes are merely names given to auxiliary variables which facilitate the efficient manipulation of the activities of the original variables in the system. They encode a summary of the interactions among the visible variables and, once calculated, permit us to treat the visible variables as if they were mutually independent.

The dual summarizing/decomposing role of a causal variable is analogous to that of an orchestra conductor: it achieves coordinated behavior through central communication and thereby relieves the players from having to communicate directly with one another. In the physical sciences, a classical example of such coordination is exhibited by the construct of a *field* (e.g., gravitational, electric, or magnetic). Although there is a one-to-one mathematical correspondence between the electric field and the electric charges in terms of which it is defined, nearly every physicist takes the next step and ascribes physical reality to the electric field, imagining that in every point of space there is some real physical phenomenon taking place which determines both the magnitude and direction which tag the point. This psychological construct offers an advantage vital to understanding the development of electrical sciences: It decomposes the complex phenomena associated with interacting electric charges into two independent processes: (1) the creation of the field at a given point by the surrounding charges, and (2) the conversion of the field into a physical force once another charge passes near that point.

The advantages of centrally coordinated architectures are not unique to star-structured networks but are also present in tree structures since every internal node in the tree centrally coordinates the activities of its neighbors. In a management hierarchy, for example, where employees can communicate with each other only through their immediate superiors, the passage of information is swift, economical, conflict-free, and highly parallel. Likewise, we know that, if the interactions among a set of variables can be represented by a tree of binary constraints, then a globally consistent solution can be found in linear time, using backtrack-free search [3, 7]. These computational advantages of trees also retain their power when the relationships constraining the variables are probabilistic in nature.

In probabilistic formalisms, the topological concept of central coordination is embodied in 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* [31, 33]; they become independent of each other once we know the state of the mechanism causing the correlation, i.e., the standard time. Thus, conditional independence captures both functions of our orchestra conductor: coordination and decomposition.

The most familiar connection between causality and conditional independence is reflected in the scientific notion of a *state*. It was devised to nullify the influence that the past exerts on the future by providing a sufficiently detailed description of the present. In probabilistic terms this came to be known as a Markov property; future events are conditionally independent of past events, given the current state of affairs. This is precisely the role played by the set of parents S_i in the construction of Bayesian networks (Section 1.1); they screen the variable x_i from the influence of all its other ancestors.

But conditional independence is not limited to separating the past from the future; it often applies to events occurring at the same time. Knowing the values of the parent set S_i not only decouples x_i from its other ancestors but renders x_i independent of *all* other variables except its descendants. In fact, this sort of independence constitutes the most universal and distinctive characteristic featured by the notion of causality. In medical diagnosis, for example, a group of cooccurring symptoms often become independent of each other once we know the disease that caused them. When some of the symptoms directly influenced 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, which again assumes the decompositional role characteristic of causal agents; knowing the exact state of the auxiliary variable renders the interacting symptoms independent of each other. In other words, the auxiliary variables constitute a sufficient summary for determining the likely development of each individual symptom in the group; thus, additional knowledge regarding the states of the other symptoms becomes superfluous.

The continuous influx of such auxiliary concepts into our languages cast new light on the status of conditional independence in probabilistic modelling. Contrary to positions often found in the literature, conditional independence is not a "restrictive assumption" made for mathematical elegance; neither is it an occasional grace of nature for which we must passively wait. Rather, it is a mental construct that we actively create and a psychological necessity which our culture labors to satisfy.

The decompositional role of causal variables attains its ultimate realization in tree-structured networks, where every pair of nonadjacent variables becomes independent given a third variable on the path connecting the pair. Indeed, the speed, stability and autonomy of the updating scheme described in Section 2.2 draws its power from the high degree of decomposition provided by the tree structure. These computational advantages, we postulate, give rise to the satisfying sensation called "in-depth understanding," which people experience when they discover 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 interpretations and predictions, we now ask whether it is possible to reconfigure every belief network as a tree and, if so, how. First we assume that there exist dummy variables which decompose the network 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 such trees may reflect an important component of human learning.

A related structuring task was treated by Chow and Liu [2], 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 probability for any pair of variables is assumed to be given. By contrast, the internal nodes in our trees denote dummy variables, artificially concocted to make the representation tree-like. Since only the leaves are accessible to empirical observations, we know neither the conditional probabilities that link the internal nodes to the leaves nor the structure of the tree—these we would have to learn. A similar problem of configuring probabilistic models with hidden variables is mentioned by Hinton et al. [9] as one of the tasks that a Boltzman machine should be able to solve. However, it is not clear whether the relaxation techniques employed by the Boltzman machine can easily escape local minima and whether they can readily accept the constraint that the resulting structure 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 the value of each interleaf correlation is precisely known.

3.2. 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.,

$$P_s(x_1, \ldots, x_n, w) = \prod_{i=1}^{n} P_s(x_i|w)P_s(w) , \tag{23}$$

$$P(x_1, \ldots, x_n) = \alpha \prod_{i=1}^{n} P_s(x_i|w=1) + (1-\alpha) \prod_{i=1}^{n} P_s(x_i|w=0) . \quad (24)$$

The functions $P_s(x_i|w)$, $w = 0, 1, i = 1, \ldots, n$, can be viewed as 2×2 stochastic matrices relating each x_i to the central hidden variable w (see Fig. 6(a)); 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

$$f_i = P_s(x_i = 1|w=1) , \qquad g_i = P_s(x_i = 1|w=0) \quad (25)$$

and the central variable w is characterized by its prior probability $P_s(w-1) = \alpha$ (see Fig. 6(b)).

The advantages of having star-decomposable distributions are several. First, the product form of P_s in (23) makes it very easy to compute the probability of any combination of variables. More importantly, the product form is also convenient for calculating the conditional probabilities, $P(x_i|x_j)$, 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 [16] 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. 7), 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.

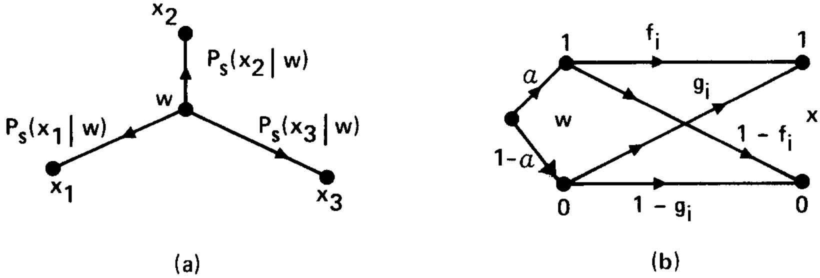


FIG. 6. (a) Three random variables, x_1, x_2, x_3 connected to a central variable w by a star network. (b) Illustration of the three parameters, α, f_i, g_i, associated with each link.

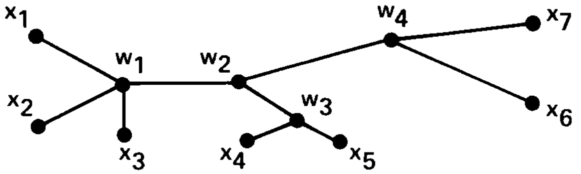


FIG. 7. A tree containing four dummy variables and seven visible variables.

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

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

that supports a tree-structured network, such that $w_1, w_2, \ldots, w_m$ correspond to the internal nodes of a tree T and $x_1, x_2, \ldots, x_n$ to its leaves.

Note that if P_T supports a rooted tree T, then any two leaves are conditionally independent, given the value of any internal node on the path connecting them. These relationships between leaves and internal nodes are a property of the undirected tree, independent of the choice of root. Now, since a choice of a new root for T will create a tree T' which is also supported by P_T, we are permitted to treat T as an unrooted tree. Conversely, 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:

$$P_T(x_1, \ldots, x_n, w_1, \ldots, w_m) = P(y_1) \prod_{i=2}^{m+n} P[y_i | F(y_i)] . \tag{26}$$

For example, if in Fig. 7 we choose w_2 as the root, we obtain:

$$\begin{aligned} &P_T(x_1, \ldots, x_7, w_1, \ldots, w_4) \\ &\quad = P(x_7|w_4)P(x_6|w_4)P(x_5|w_3)P(x_4|w_3) \\ &\quad\quad \cdot P(x_3|w_1)P(x_2|w_1)P(x_1|w_1)P(w_1|w_2) \cdot P(w_3|w_2)P(w_4|w_2)P(w_2) . \end{aligned}$$

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. Similarly, we shall assume that all link matrices are nonsingular, conveying genuine dependencies between the linked variables; otherwise, the tree can be decomposed into disconnected components, i.e., a forest.

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 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. (Impatient readers may skip this analysis and go directly to Theorem 3.1.)

3.3. 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 (24) and express the parameters α, f_i, g_i as a function of the parameters specifying P. This task was carried out by Lazarfeld [16] in terms of the seven joint-occurrence probabilities.

$$\begin{aligned} p_i &= P(x_i = 1)\,, \\ p_{ij} &= P(x_i = 1, x_j = 1)\,, \\ p_{ijk} &= P(x_i = 1, x_j = 1, x_k = 1)\,, \end{aligned} \tag{27}$$

and led to the following solution:

Define the quantities,

$$[ij] = p_{ij} - p_i p_j\,, \tag{28}$$

$$S_i = \left[\frac{[ij][ik]}{[jk]} \right]^{1/2}, \tag{29}$$

$$\mu_i = \frac{(p_i p_{ijk} - p_{ij} p_{ik})}{[jk]}\,, \tag{30}$$

$$K = \frac{S_i}{p_i} - \frac{p_i}{s_i} + \frac{\mu_i}{S_i p_i}\,, \tag{31}$$

and let t be the solution of

$$t^2 + Kt - 1 = 0\,. \tag{32}$$

The parameters α, f_i, g_i are given by:

$$\alpha = t^2/(1 + t^2)\,, \tag{33}$$

$$f_i = p_i + S_i[(1-\alpha)/\alpha]^{1/2}\,, \tag{34}$$

$$g_i = p_i - S_i[\alpha/(1-\alpha)]^{1/2}\,. \tag{35}$$

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

$$f_i - g_i = S_i = \Big[[ij][ik]/[jk]\Big]^{1/2}\,. \tag{36}$$

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

(a) S_i should be real,
(b) $0 \leqslant f_i \leqslant 1$,
(c) $0 \leqslant g_i \leqslant 1$.

Using the variances

$$\sigma_i = [p_i(1-p_i)]^{1/2} \tag{37}$$

and the correlation coefficients

$$\rho_{ij} = (p_{ij} - p_i p_j)/\sigma_i\sigma_j\,, \tag{38}$$

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), yields (see Appendix B):

Theorem 3.1. *A necessary and sufficient condition for three dichotomous random variables to be star-decomposable is that they are positively correlated, and that the inequality,*

$$\frac{p_{ik}p_{ij}}{p_i} \leqslant p_{ijk} \leqslant \frac{p_{ik}p_{ij}}{p_i} + \sigma_j\sigma_k(\rho_{jk} - \rho_{ij}\rho_{ik})\,, \tag{39}$$

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 (39), 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 3.2. *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*:

$$\rho_{jk} \geqslant \rho_{jk}\rho_{ik} \ . \tag{40}$$

Inequality (40) 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 (39) 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 (39) is nonempty. This leads to the statement:

Theorem 3.3. *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*,

$$\frac{p_{ij}p_{ik}}{p_i} \leqslant x \leqslant \frac{p_{ij}p_{ik}}{p_i} + \sigma_j\sigma_k(\rho_{jk} - \rho_{ij}\rho_{ik}) \ , \quad i = 1, 2, 3 \ , \tag{41}$$

possess a solution for x.

3.4. A tree-reconstruction procedure

We are now ready to confront the central problem of this section—given a tree-decomposable distribution $P(x_i, \ldots, x_n)$, 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 (33), (34), and (35), 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, and 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. 8. The topologies differ in the identity of the triplets which share a common center. For example, in the topology of Fig. 8(a) the pair [(1, 2, 3), (1, 2, 4)] share a common center, and so does the pair [(1, 3, 4), (2, 3, 4)]. In Fig. 8(b), on the other hand, the sharing pairs are [(1, 2, 4), (2, 4, 3)] and [(1, 3, 4), (2, 1, 3)], and in Fig. 8(d) all triplets share the same center. Thus, the basic test for 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 should x_4 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 we 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 should x_{i+1} 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.

It is possible to show [26] that, if we choose, 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 T_i. This amounts to $O(n \log n)$ test 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 is, likewise, uniquely determined from P, i.e., that we can determine all the functions $P(x_i|w_j)$ and

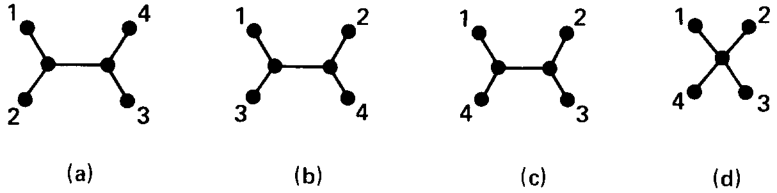


FIG. 8. The four possible topologies by which four leaves can be related.

$P(w_j|w_k)$ in (26), for $i=1,\ldots,n$ and j, $k=1, 2,\ldots,m$. The functions $P(x_i|w_j)$ assigned to the peripheral branches of the tree are determined directly from the star decomposition of triplets involving adjacent leaves. In Fig. 7, for example, the star decomposition of $P(x_1, x_2, x_5)$ yields $P(x_1|w_1)$ and $P(x_2|w_1)$. The conditional probabilities $P(w_j|w_k)$ assigned to interior branches are determined by solving matrix equations. For example, $P(x_1|w_2)$ can be obtained from the star decomposition of (x_1, x_5, x_7), and it is related to $P(x_1|w_1)$ via

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

This matrix equation has a solution for $P(w_1|w_2)$ because $P(x_1|w_1)$ 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 them to determine more interior branches, and so on, until all the interior conditional probabilities $P(w_i|w_j)$ 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. 8(a)), 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. 8(a)) must also be the same. While the value of α depends on a third-order probability, the difference f_i-g_i depends only on second-order terms via (36). Thus, requiring that f_1-g_1 in Fig. 8(a) obtain the same value in the star decomposition of $(1,2,3)$ as in that of $(1,2,4)$ leads to the equation:

$$[12][13]/[23]=[12][14]/[24] \tag{42}$$

which, using (28), yields

$$\rho_{13}\rho_{42}=\rho_{14}\rho_{32}\,. \tag{43}$$

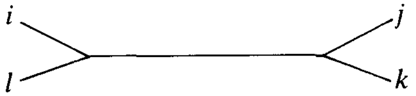
An identical equality will be obtained for each f_i-g_i, $i=1, 2, 3, 4$, relative to the topology of Fig. 8(a). Similarly, the topology of Fig. 8(b) dictates

$$\rho_{12}\rho_{43}=\rho_{14}\rho_{23} \tag{44}$$

and that of Fig. 8(c) dictates:

$$\rho_{12}\rho_{34}=\rho_{13}\rho_{24}\,. \tag{45}$$

Thus, we see that each of these three topologies is characterized by its own distinct equality, while the topology of Fig. 8(d) is distinguished by all three equalities holding 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 permutations with distinct topologies, 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.

3.5. Conclusions and open questions

This section 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 works only when there exists an underlying model that is tree-structured. In practice, we often have only sample estimates of the correlation coefficients; therefore, it is unlikely that criteria based on equalities (as in (43)) 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 from 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* for 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 3.4, 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 A. Derivation of the Updating Rules for Singly Connected Networks

A.1. Updating BEL

Starting with

$$\mathrm{BEL}(A_i) \overset{\Delta}{=} P(A_i | D^+_{BA}, D^+_{CA}, D^-_{AX}, D^-_{AY}),$$

we apply Bayes' rule, and obtain

$$\mathrm{BEL}(A_i) = \alpha P(D^-_{AX}, D^-_{AY} | A_i, D^+_{BA}, D^+_{CA}) P(A_i | D^+_{BA}, D^+_{CA}).$$

The conditional independence of (14) now yields (15):

$$\mathrm{BEL}(A_i) = \alpha P(D^-_{AX}, A_i) P(D^-_{AY} | A_i) P(D^-_{AY} | A_i) P(A_i | D^+_{BA}, D^+_{CA}).$$

Conditioning and summing over the values of B and C, we get

$$\begin{aligned}\mathrm{BEL}(A_i) &= \alpha P(D^-_{AX} | A_i) P(D^-_{AY} | A_i) \\ &\quad \cdot \sum_{B,C} P(A_i | D^+_{BA}, D^+_{CA}, B, C) P(B, C | D^+_{BA}, D^+_{CA}) \\ &= \alpha P(D^-_{AX} | A_i) P(D^-_{AY} | A_i) \\ &\quad \cdot \sum_{B,C} P(A_i | B, C) P(B | D^+_{BA}) P(C | D^+_{CA})\end{aligned}$$

making use of the fact that B and C are independent, given data from nondescendants of A. This confirms (16):

$$\mathrm{BEL}(A_i) = \alpha P(D^-_{AX}|A_i)P(D^-_{AY}|A_i)$$

$$\cdot\Big[\sum_{j,k} P(A_i|B_j, C_k)P(B_j|D^+_{BA})P(C_k|D^+_{CA})\Big]$$

and, using the λ-π notation

$$\lambda_X(A_i) = P(D^-_{AX}|A_i)\,, \qquad \pi_A(B_j) = P(B_j|D^+_{BA})\,,$$

we obtain (19)

$$\mathrm{BEL}(A_i) = \alpha\lambda_X(A_i)\lambda_Y(A_i)\Big[\sum_{j,k} P(A_i|B_j, C_k)\pi_A(B_j)\pi_A(C_k)\Big]\,.$$

A.2. Updating π

$$\pi_X(A_i) = P(A_i|D^+_{AX}) = P(A_i|D - D^-_{AX})$$

$$= \mathrm{BEL}(A_i|\lambda_X(A) = (1, 1, \ldots, 1))$$

$$= \alpha\lambda_Y(A_i)\Big[\sum_{j,k} P(A_i|B_j, C_k)\pi_A(B_j)\pi_A(C_k)\Big]\,,$$

thus confirming (22).

A.3. Updating λ

$$\lambda_A(B_i) = P(D^-_{AB}|B_i) = P(A, D^-_{AX}, D^-_{AY}, D^+_{CA}|B_i)$$

$$= \sum_{j,k} P(D^-_{AX}, D^-_{AY}, D^+_{CA}|B_i, C_j, A_k)P(C_j, A_k|B_i)$$

$$= \sum_{j,k} P(D^-_{AX}|A_k)P(D^-_{AY}|A_k)P(D^+_{CA}|C_j)$$

$$\cdot P(A_k|B_i, C_j)P(C_j|B_i)\,.$$

But $P(C_j|B_i) = P(C_j)$ because B and C are marginally independent, and

$$P(D^+_{CA}|C_j)P(C_j) = \alpha P(C_j|D^+_{CA})$$

by Bayes' rule. Therefore,

$$\begin{aligned}\lambda_A(B_i) &= \alpha \sum_{j,k} P(D^-_{AX}|A_k)P(D^-_{AY}|A_k)P(C_j|D^+_{CA})P(A_k|C_j, B_i)\\ &= \alpha \sum_{j,k} \lambda_X(A_k)\lambda_Y(A_k)\pi_A(C_j)P(A_k|B_i, C_j)\\ &= \alpha \sum_{j} \left[\pi_A(C_j) \sum_{k} \lambda_X(A_k)\lambda_Y(A_k)P(A_k|B_i, C_j)\right],\end{aligned}$$

which confirms (21).

Appendix B. Conditions for Star-decomposability

Let

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

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:

$$\begin{aligned}P(x_1 = 1, x_2 = 1, x_3 = 0) &= p_{12} - p_{123}\,,\\ P(x_1 = 1, x_2 = 0) &= p_1 - p_2\,, \quad \text{etc.}\end{aligned}$$

These probabilities will be used in the following analysis.

Assuming P is star-decomposable (equations (23) and (24)), we can express the joint-occurrence probabilities in terms of α, f_i, g_i and obtain seven equations for these seven parameters.

$$p_i = \alpha f_i + (1 - \alpha)g_i\,, \qquad \text{(B.2)}$$

$$p_{ij} = \alpha f_i f_j + (1 - \alpha)g_i g_j\,, \qquad \text{(B.3)}$$

$$p_{ijk} = \alpha f_i f_j f_k + (1 - \alpha)g_i g_j g_k\,. \qquad \text{(B.4)}$$

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

$$p_{ij} - p_i p_j = \alpha(1 - \alpha)(f_i - g_i)(f_j - g_j)\,, \qquad \text{(B.5)}$$

$$p_i p_{ijk} - p_{ij} p_{ik} = \alpha(1 - \alpha)f_i g_i (f_j - g_j)(f_k - g_k)\,. \qquad \text{(B.6)}$$

Equation (B.5) comprises three equations which can be solved for the differences $f_i - g_i$, $i = 1, 2, 3$, giving

$$f_i - g_i = S_i = \pm\Big[[ij][ik]/[jk]\Big]^{1/2} , \tag{B.7}$$

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

$$[ij] = p_{ij} - p_i p_j . \tag{B.8}$$

These, together with (B.2), determine f_i and g_i in terms of S_i and α (still unknown):

$$f_i = p_i + S_i[(1-\alpha)/\alpha]^{1/2} , \tag{B.9}$$

$$g_i = p_i - S_i[\alpha/(1-\alpha)]^{1/2} . \tag{B.10}$$

To determine α, we invoke (B.6) and obtain

$$[\alpha/(1-\alpha)]^{1/2} = t \quad \text{or} \quad \alpha = t^2/(1-t^2) , \tag{B.11}$$

where t is a solution to

$$t^2 + Kt - 1 = 0 , \tag{B.12}$$

and K is defined by:

$$K = \frac{S_i}{p_i} - \frac{p_i}{S_i} + \frac{\mu_i}{S_i p_i} , \tag{B.13}$$

$$\mu_i = [jk, i]/[jk] = (p_i p_{ijk} - p_{ij} p_{ik})/[jk] . \tag{B.14}$$

It can be easily verified that K (and, therefore, α) obtains the same value regardless of which index i provides the parameters in (B.13).

From (B.13) we see that the parameters S_i and μ_i of P govern the solutions of (B.12) 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 be real,
(b) $0 \leqslant f_i \leqslant 1$,
(c) $0 \leqslant g_i \leqslant 1$.

Requirement (a) implies that, of the three brackets in (B.7), either all three are nonnegative, or exactly two are negative. These brackets are directly related to the correlation coefficient via:

$$\rho_{ij} = [ij][p_i(1-p_i)]^{-1/2}[p_j(1-p_j)]^{-1/2} = [ij]/\sigma_i\sigma_j \tag{B.15}$$

and so, 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.

Now attend to requirement (b). Equation (B.9) 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 (B.7). However, the choice of negative S_i yields a solution (f_i', g_i', α') which is symmetrical to (f_i, g_i, α) stemming from a positive S_i, with $f_i' = g_i$, $g_i' = f_i$, $\alpha' = 1 - \alpha$. Thus, S_i and f_i can be assumed to be nonnegative, and it remains to examine the condition $f_i \leqslant 1$ or, equivalently, $t \geqslant S_i/(1-p_i)$ (see (B.9) and (B.11)). Imposing this condition in (B.12) translates to:

$$p_{ijk} \leqslant p_{ij}p_{ik}/p_i + \sigma_k\sigma_j[\rho_{jk} - \rho_{ij}\rho_{ik}]\,. \tag{B.16}$$

Similarly, inserting requirement (c), $g_i \geqslant 0$, in (B.12) yields the inequality:

$$p_{ik}p_{ij}/p_i \leqslant p_{ijk} \tag{B.17}$$

which, together with (B.16), lead to Theorem 3.1.

ACKNOWLEDGMENT

I thank many people for helping me prepare this manuscript. N. Dalkey has called my attention to the work of Lazarfeld [16] and has provided a continuous stream of valuable advice. Thomas Ferguson made helpful comments on Section 3. Jin Kim is responsible for deriving the propagation equations of Section 2.3 [14] and for using the propagation scheme in his decision support system CONVINCE [13]. Michael Tarsi has helped develop the tree-construction algorithm of Section 3.4 and has proved its optimality [26]. Eli Gafni has spent many hours discussing the relations among Markov fields, stochastic relaxation, and belief propagation. Ed Pednault, Nils Nilsson, Henry Kyburg, and Igor Roizen have thoroughly reviewed earlier versions of this paper and have suggested many improvements.

The results of Section 3 were presented at the International Joint Conference on Artificial Intelligence, University of California, Los Angeles, August 19–23, 1985.

REFERENCES

1. Anderson, J.R., *The Architecture of Cognition* (Harvard University Press, Cambridge, MA, 1983).
2. Chow, C.K. and Liu, C.N., Approximating discrete probability distributions with dependence trees, *IEEE Trans. Inf. Theory* **14** (1968) 462–467.
3. Dechter, R. and Pearl, J., The anatomy of easy problems: A constraint-satisfaction formulation, in: *Proceedings Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA, (1985) 1066–1072.
4. Dell, G.S., Positive feedback in hierarchical connectionist models: Applications to language production, *Cognitive Sci.* **9** (1) (1985) 3–24.

5. Doyle, J. A truth maintenance system, *Artificial Intelligence* **12** (1979) 231–272.
6. Duda, R.O., Hart, P.E. and Nilsson, N.J., Subjective Bayesian methods for rule-based inference systems, in: *Proceedings 1976 National Computer Conference* (*AFIPS Conference Proceedings*) **45** (1976) 1075–1082.
7. Freuder, E.C., A sufficient condition of backtrack-free search, *J. ACM* **29** (1) (1982) 24–32.
8. Geman, S. and Geman, D., Stochastic relaxations, Gibbs distributions, and the Bayesian restoration of images, *IEEE Trans. Pattern Anal. Machine Intelligence* **6** (6) (1984) 721–742.
9. Hinton, G.E., Sejnowski, T.J. and Ackley, D.H., Boltzman machines: Constraint satisfaction networks that learn, Tech. Rept. CMU-CS-84-119, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, 1984.
10. Howard, R.A. and Matheson, J.E., Influence diagrams, in: R.A. Howard and J.E. Matheson (Eds.), *The Principles and Applications of Decision Analysis* (Strategic Decisions Group, Menlo Park, CA, 1984).
11. Jeffrey, R., *The Logic of Decisions* (McGraw-Hill, New York, 1965).
12. Kemeny, J.G., Snell, J.L. and Knapp, A.W., *Denumerable Markov Chains* (Springer, Berlin, 2nd ed,. 1976).
13. Kim, J., CONVINCE: A CONVersational INference Consolidation engine, Ph.D. Dissertation, University of California, Los Angeles, CA, 1983.
14. Kim, J. and Pearl, J., A computational model for combined causal and diagnostic reasoning in inference systems, in: *Proceedings Eighth International Joint Conference on Artificial Intelligence*. Karlsruhe, F.R.G. (1983) 190–193.
15. Lauritzen, S.L., *Lectures on Contingency Tables* (University of Aalborg Press, Aalborg, Denmark, 2nd ed., 1982).
16. Lazarfeld, P.F., Latent structure analysis, in: S.A. Stouffer, L. Guttman, E.A. Suchman, P.F. Lazarfeld, S.A. Star and J.A. Claussen (Eds.), *Measurement and Prediction* (Wiley, New York, 1966).
17. Lesser. V.R. and Erman, L.D., A retrospective view of HEARSAY II architecture, in: *Proceedings Fifth International Joint Conference on Artificial Intelligence*, Cambridge, MA (1977) 790–800.
18. Levy, H. and Low, D.W., A new algorithm for finding small cycle cutsets, Rept. G 320-2721, IBM Los Angeles Scientific Center, Los Angeles, CA, 1983.
19. Lowrance, J.D., Dependency-graph models of evidential support, COINS Tech. Rept. 82-26, University of Massachusetts at Amherst, MA, 1982.
20. McAllester, D., An outlook on truth maintenance, AIM-551, Artificial Intelligence Laboratory, MIT, Cambridge, MA, 1980.
21. Pearl, J., Reverend Bayes on inference engines: A distributed hierarchical approach, in: *Proceedings Second National Conference on Artificial Intelligence*, Pittsburgh, PA (1982) 133–136.
22. Pearl, J., A constraint-propagation approach to probabilistic reasoning, in: *Proceedings Workshop on Uncertainty and Probability in AI*, Los Angeles, CA (1985) 31–42; also in: L.N. Kanal and J.F. Lemmer (Eds.), *Uncertainty in Artificial Intelligence* (North-Holland, Amsterdam, 1986) 357–370.
23. Pearl, J., Distributed diagnosis in causal models with continuous variables, Tech. Rept. CSD-860051, Cognitive Systems Laboratory, Computer Science Department, University of California, Los Angeles, 1985.
24. Pearl, J. and Paz, A., Graphoids: A graph-based logic for reasoning about relevancy relations, Tech. Rep. CSD-850038, Cognitive Systems Laboratory, Computer Science Department, University of California, Los Angeles, 1985.
25. Pearl, J., On evidential reasoning in a hierarchy of hypotheses, *Artificial Intelligence* **28** (1986) 9–15.
26. Pearl, J. and Tarsi, M., Structuring causal trees, *J. Complexity* **2** (1) (1986) 60–77.

27. Rosenfeld, A., Hummel, A. and Zucker, S., Scene labelling by relaxation operations, *IEEE Trans. Syst. Man Cybern.* **6** (1976) 420–433.

28. Rumelhart, D.E., Toward an interactive model of reading, *Center for Human Information Proceedings CHIP-56*, University of California, San Diego, La Jolla, CA, 1976.

29. Rumelhart, D.E. and McClelland, J.L., An interactive activation model of context effects in letter perception: Part 2. The contextual enhancement effect and some tests and extensions of the model, *Psychol. Rev.* **89** (1982) 60–94.

30. Shastri, L. and Feldman, J.A., Semantic networks and neural nets, TR-131, Computer Science Department, The University of Rochester, Rochester, NY, 1984.

31. Simon, H.A., Spurious correlations: A causal interpretation, *J. Am. Stat. Assoc.* **49** (1954) 469–492.

32. Spiegelhalter, D.J., Probabilistic reasoning in predictive expert systems, in: L.N. Kanal and J.F. Lemmer (Eds.), *Uncertainty in Artificial Intelligence* (North-Holland, Amsterdam, 1986) 47–68.

33. Suppes, P., *A Probabilistic Theory of Causality* (North-Holland, Amsterdam, 1970).

34. Tverski, A. and Kahneman, D., Causal schemata in judgments under uncertainty, in: M. Fishbein (Ed.), *Progress in Social Psychology* (Erlbaum, Hillsdale, NJ., 1977).

35. Waltz, D.G., Generating semantic descriptions from drawings of scenes with shadows, AI TR-271, Artificial Intelligence Laboratory, Cambridge, MA, 1972.

Received January 1982; revised version received February 1986