BAYESIAN NETWORKS: A MODEL OF SELF-ACTIVATED MEMORY FOR EVIDENTIAL REASONING

Judea Pearl

April 1985
Report No. CSD-850017

R-43
April 1985

BAYESIAN NETWORKS: A MODEL OF SELF-ACTIVATED MEMORY FOR EVIDENTIAL REASONING*

Judea Pearl
Computer Science Department
University of California
Los Angeles, CA 90024
(judea@UCLA-locus)
(213) 825-3243

Topics: Memory Models
Belief Systems
Inference Mechanisms
Knowledge Representation

Submitted to the
Seventh Annual Conference of
the Cognitive Science Society
15-17 August 1985

Paper length: 5666

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

BAYESIAN NETWORKS: A MODEL OF SELF-ACTIVATED MEMORY FOR EVIDENTIAL REASONING

ABSTRACT

Bayesian networks are directed acyclic graphs in which the nodes represent propositions (or variables), the arcs signify the existence of direct causal 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 deep causal knowledge of an agent or a domain expert and 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 defines the properties of Bayes networks which are necessary to guarantee completeness and consistency, and shows how dependencies and conditional-independence relationships can be tested using simple link-tracing operations.

The second part of the paper 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 belief measure consistent with the observed data. We first argue 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 Bayes network itself. We then quote results which show that these objectives can be fully realized only in singly-connected networks, where there exists only one (undirected) path between any pair of nodes. Finally, the paper discusses several approaches to achieving belief propagation in more general networks, and argues for the feasibility of turning a Bayes network into a tree by introducing dummy variables, mimicking the way people develop causal models.

BAYESIAN NETWORKS: A MODEL OF SELF-ACTIVATED MEMORY FOR EVIDENTIAL REASONING

Judea Pearl

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 judgmental--namely, 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 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 marginal probabilities of a small subset, $H_1, \ldots, H_k$, of variables called hypotheses, conditioned upon a group of instantiated variables $e_1 \cdots e_m$ 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 an arithmetic tediousness 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 find some economical way of storing $P(x_1, \ldots, x_n)$ (or rules for generating it), there still remains the problem of manipulating it to compute the probabilities of propositions which people consider to be interesting. For example, to compute 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 x_j, $j \neq i$. Similarly, computing the conditional probability $P(x_i | x_j)$ from its textbook definition $P(x_i | x_j) = \frac{P(x_i, x_j)}{P(x_j)}$ would involve dividing two marginal probabilities, each resulting from summation over an exponentially large number of variable combinations. Human performance, by contrast, exhibits an opposite complexity ordering; probabilistic judgments on a small number of propositions (especially 2-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 is done with 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 judgmental 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 the conditional probability $P(x_i | x_j)$, no hesitation will normally be encountered when that person is asked to state merely whether x_i and x_j are dependent or independent, namely, whether knowing the truth of x_j will or will not alter the belief in x_i. The 3-place relationships of conditional dependency (i.e. x_i influences x_j given x_k) are likewise judged by people with a great deal of clarity, conviction, and consistency.

This suggests that the notions of dependence and conditional dependence are more basic than the numerical values attached to probability judgments, 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 the set of propositions which directly affect the truth value of q, and to describe them as the direct neighbors of q, which *isolate* q from all other influences. This suggests that the fundamental structure of human judgmental knowledge can be represented by dependency graphs and that mental tracing of links in these graphs are responsible for the basic steps in querying and updating that knowledge.

2. BAYESIAN 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 by weights that signify the strength and type of dependencies between the connected propositions. If these weights are to be interpreted later as conditional probabilities, two problems must first be attended to: *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.

One of the attractive features of the joint-distribution representation of probability is the transparency by which one can synthesize consistent probability models or detect inconsistencies therein. In this representation, all we need to do is to assign non-negative weights to the atomic compartments in the space (i.e., conjunctions of propositions), make sure the weights sum to one, and a complete model, free of inconsistencies is created. By contrast, the synthesis process in the graph representation is much 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 non-obvious relationship, there exists no probability model that will support all three probabilities.

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

$$P(x_1,x_2, \cdots x_n) = P(x_n|x_{n-1} \cdots x_1)P(x_{n-1}|x_{n-2} \cdots x_1) \cdots P(x_3|x_2,x_1)\, P(x_2|x_1)P(x_1)$$

In this formula, each factor contains only one variable on the left side of the conditioning bar, and in that way the formula can be used as a prescription for consistently quantifying the dependencies among the nodes of an arbitrary graph. Given a graph G, assign an arbitrary order to its nodes and impose directionality on the links pointing from low-order to high-order nodes. To each node x_i assign an arbitrary function $F_i(x_i,S_i)$ satisfying

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

$$0 \leq F_i(x_i,S_i) \leq 1$$

where S_i is the set of x_i's parents and the summation ranges over all values of x_i. This assignment is complete and consistent; it defines a joint distribution function given by the product:

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

and the functions $F_i(x_i,S_i)$ are the marginal distributions $P(x_i|S_i)$ dictated by $P(x_1, \cdots x_n)$. For example, the distribution corresponding to the graph of Figure 1 can be written by inspection:

$$P\ (x_1, x_2, x_3, x_4, x_5, x_6) = P(x_6|x_5)\ P(x_5|x_2, x_3)\ P(x_4|x_1x_2)\ P(x_3|x_1)\ P(x_2|x_1)\ P(x_1).$$

This also leads to a simple method of constructing a dependency-graph representation to any given joint distribution $P(x_1 \cdots x_n)$. We start by imposing an arbitrary order d on the set of variables, $x_1 \cdots x_n$, then choose x_1 as a root of the graph, and assign to it the marginal probability $P(x_1)$ dictated by $P(x_1, \cdots x_n)$. 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 i^{th} stage, we form the node x_i and establish a group of directed links to x_i from the smallest subset of nodes $S_i \subseteq \{x_1 \cdots x_{i-1}\}$ satisfying the condition:

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

It is easy to show that the minimal subset S_i is unique. Thus, the distribution $P(x_1, \cdots x_n)$, together with the order d uniquely prescribe a set of parent nodes for each variable x_i, and that constitutes a full specification of a directed acyclic graph which represents the dependencies imbedded in $P(x_1, \cdots x_n)$. We shall call this graph "*Bayes Network*" or "*Influence Network*", interchangeably; the former to emphasize the judgmental origin of the quantifiers, the latter to vindicate 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 is used mainly for computational convenience [Howard and Matheson, 1984].

In the strictest sense, these networks are not graphs but hypergraphs, because 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 in highlighting the essential interactions between the variables, and in modelling the computational processes involved in inferential reasoning.

Note that the topology of a Bayes network may be extremely sensitive to the node ordering d; a network which has an inverted-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_o, then the underlying influence network would be a complete graph.

This 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 agreement regarding the structure of influence networks stem from the dominant role *causality* plays in the formation of these networks. Thus, the standard ordering imposed by the direction of causation also induces identical topologies on the networks that people adopt for encoding experiential knowledge. It is easy to speculate that if it were not for the social convention to adopt a standard ordering of events, conforming to the flow of time and causation, human communication would become an impossible task.

3. CONDITIONAL INDEPENDENCE AND GRAPH SEPARABILITY

To facilitate the verification of dependencies between the variables in a Bayes network, we need to establish a clear correspondence between the topology of the network and various types of independence. Ideally, we would have liked to associate independence between variables with the lack of connectivity between their corresponding nodes. Likewise, we would have liked to require that if the removal of some subset S of nodes from the network renders nodes x_i and x_j disconnected, then this separation should indicate conditional independence between x_i and x_j given S, namely,

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

This would provide a clear graphical representation to the notion that x_j does not affect x_i directly but, rather, its influence is mediated by the variables in S.

Unfortunately, Bayes networks do not provide this simple representation of independence; a modified criterion of separability is required that takes into account 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, $x_1 \rightarrow x_2 \leftarrow x_3$

From the method of constructing the network, it is clear that (assuming x_1, x_2, x_3 are the only variables involved) 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 $\cdots$, 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 whose values are known at a given time.

DEFINITION: (a) A path P is *connected with respect to a subset* S_e of evidence variables if all successive links along P are *joined w.r.t.* S_e.

(b) Two links, meeting head-to-tail or tail-to-tail at node X, are *joined w.r.t.* S_e if X is not in S_e.

(c) Two links meeting head-to-head at node X, are *joined w.r.t.* S_e if X or any of its descendants is in S_e.

This definition permits us to define *separability* with respect to a subset of observations which, in turn, provides a graphical criterion for testing conditional independence.

DEFINITION: A subset of variables S_e is said to *separate* x_i from x_j if there is no path between x_i and x_j which is connected w.r.t. S_e.

It is not hard to see that if S_e separates x_i from x_j, then x_i is conditionally independent of x_j given S_e. Moreover, the procedure involved in testing separation w.r.t. a given subset S_e is only slightly more complicated than that of testing whether S_e is a separating cut set, and can be handled by visual inspection. In Figure 1, for example, one can easily verify that variables x_2 and x_3 are separated w.r.t. $S_e = \{x_1\}$ or $S_e = \{x_1, x_4\}$ but not w.r.t. $S_e = \{x_1, x_6\}$, because x_6, being a descendant of x_5, "joins" the head-to-head links at x_5, which amounts to forming a connected path between x_2 and x_3.

4. AUTONOMOUS PROPAGATION AS A COMPUTATIONAL PARADIGM

Once an influence network is constructed, it can be used to represent the generic causal 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 and calculating its impact on the probabilities of a set of variables designated as hypotheses. In general, this process can be executed by an external interpreter who 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 seems foreign to the reasoning process normally exhibited by humans [Shastri and Feldman, 1984]. Our limited short-term memory and narrow focus of attention, combined with our inflexibility of shifting rapidly between alternative lines of reasoning seem to suggest that our reasoning process is fairly local, progressing incrementally along prescribed pathways. Moreover, the speed and ease with which we perform some of the low level interpretive functions, such as recognizing scenes, comprehending 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 modelling 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, logically 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 update its own. In this fashion the impact of new evidence may propagate up and down the network until equilibrium is reached.

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 and the individual computations from the control mechanism which schedules these computations to achieve some control strategy goal. This guarantees an ultimate flexibility for a sequential controller; the computations can be performed in any order, without the need to remember which parts of the network have or have not been updated already. 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 between individual processes can be simulated using a blackboard architecture [Lesser and Erman, 1977] 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.

The asynchronous nature of this model also requires a solution to an instability problem. If a stronger belief in a given hypothesis means a greater expectation for the occurrence of a certain manifestation and if, in turn, a greater certainty in the occurrence of that manifestation adds further credence to the hypothesis, how can one avoid an infinite updating loop when the two processors begin to communicate with one another?

5. PROPAGATION IN SINGLY-CONNECTED NETWORKS

It turns out that this, as well as other problems associated with asynchronous propagation of beliefs, can be solved completely if the network is singly connected, namely, if there is one underlying path between any pair of nodes. These include trees, where each node has a single parent, as well as graphs with multi-parent nodes, representing events with several causal factors. The analysis of trees is carried out in Pearl [1982], and the extension to general singly connected graphs is reported in Kim and Pearl [1983]. In both cases, the belief-updating scheme possesses the following properties:

1. New information diffuses through the network in a single pass, i.e., the time required for completing the diffusion (in parallel) is equal to the diameter of the network.

2. Instabilities (or infinite relaxation) due to cyclic inferences are eliminated by using multiple belief parameters in every node, each representing the degree of belief contributed by a different source of information.

3. The primitive processors are simple, repetitive, and save for performing matrix multiplications, they require no working memory. 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.

4. 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. 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 cognitive tasks such as visual recognition, reading comprehension [Rumelhart, 1976], and associative retrieval [Anderson, 1983], where unsupervised parallelism is an uncontested mechanism.

It is also interesting to note that the marginal conditional probabilities on the links of the network retain their viability throughout the updating process. This is remarkable because $P(A|B)$ only defines the belief of A under very special sets of circumstances, namely, when the value of B is known with absolute certainty, and when no other evidential data is available. In normal circumstances, though, all internal nodes in the network are subject to some uncertainty and, more seriously, after observing evidence e, the relation between BEL(A) and BEL(B) is no longer governed by $P(A|B)$, but by $P(A|B, e)$, which may be vastly different. The ability to maintain a constant set of weights on the links is essential, since having to adjust the weights with the arrival of each new data would be computationally prohibitive. One is tempted to speculate, therefore, that this may be the reason that people choose the marginal conditional probabilities as standard primitives for organizing stable conceptual information which, in turn, also explains why people are more proficient in assessing the magnitude of these relationships rather than of any other probabilistic quantity.

6. MANAGING LOOPS AND THE DEVELOPMENT OF CAUSAL MODELS

The efficacy of singly-connected networks in supporting autonomous propagation raises the question of whether similar propagation mechanisms exist in less restrictive networks (e.g., the one in Figure 1), in which multiple parents possess common ancestors, thus forming (undirected) loops. So far, our investigation has failed to find a propagation method for loops that retains all the advantages cited above. For example, a straightforward way of handling the network of Figure 1 would be to appoint a local interpreter for the loop x_1, x_2, x_3, x_5 that will pass messages directly between x_1 and x_5, accounting 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_1, x_2). The method works well on small loops, but as soon as the number of variables exceeds 3 or

4, collapsing requires handling huge matrices and washes away the natural conceptual structure imbedded in the original network.

An alternative method would be for each node to continue communicating with its neighbors as if the network was singly-connected, ignoring the possibility of loops. This will set up messages circulating indefinitely around the loop, until equilibrium is approached. The convergence and coherence properties of such a process are yet uncertain, but all indications point to difficulties in achieving stability.

A third method of propagation is based on "stochastic relaxation" [Geman and Geman, 1984]. Each processor interrogates the states of the variables within its Markov neighborhood (see Section 3), computes a belief distribution for the values of its host variable, then selects one of these values with probability that equals the computed belief. The value chosen will subsequently be interrogated by the neighbors upon computing their beliefs, and so on. This scheme is guaranteed convergence, but usually requires very long relaxation times to reach a steady state.

Finally, an approach is described in Pearl [1984], which introduces auxiliary variables that turn the network into a tree. Consider an arbitrary tree-structured network. The leaves in this network are tightly coupled in the sense that no two of them can separate the other two. Therefore, if we were to construct an influence network based on these variables *alone*, a complete graph would ensue. Yet, the inclusion of the intermediate variables manages to turn that graph into a tree. The question is now: Which networks can be broken up into trees by introducing dummy variables?

In some respect, this method is similar to that of appointing external interpreters to handle non-separable components of the graph, because the dummy variables are assigned processors that mediate between the original variables. However, the dummy-variables scheme enjoys the added advantage of uniformity: the processors representing the dummy variables can be identical 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.

To take advantage of this centrally-organized architecture, people often invent 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.

Computationally speaking, causes are names given to auxiliary variables which encode a summary of the interaction between the visible variables and, once calculated, permit us to treat the visible variables as if they were mutually independent. Thus, the restructuring of Bayes networks into trees by introducing auxiliary variables shares many computational features with the development of causal models in people. It is very suggestive, therefore, to conjecture that the auxiliary 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 features similar to those offered by tree-structured Bayes networks.

REFERENCES

Anderson, John R., (1983), "The Architecture of Cognition", Harvard University Press, Cambridge, MA.

Geman, S. and Geman, D., (1984), Stochastic Relaxations, Gibbs Distributions, and the Bayesian Restoration of Images, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-6, No. 6, 721-742, November.

Hinton, G.E., Sejnowski, T.J., and Ackley, D.H., (1984), Boltzman Machines: Constraint Satisfaction Networks that Learn, Technical Report CMU-CS-84-119, Department of Computer Science, Carnegie-Mellon University.

Kim, J. and Pearl, J., (1983), A Computational Model for Combined Causal and Diagnostic Reasoning in Inference Systems, *Proceedings of IJCAI-83*, 190-193.

Lesser, V.R. and Erman, L.D., (1977), A Retrospective View of HEARSAY II Architecture, *Proc. 5th Int. Joint Conf. AI*, Cambridge, MA, 790-800.

Pearl, J., (1982), "Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach", *Proc. AAAI Nat'l. Conf. on AI*, Pittsburgh, PA, pp. 133-136, August.

Pearl, J., (1984), "Learning Hidden Causes from Empirical Data", Technical Report R-38, UCLA Computer Science Dept. To be published in the *Proceedings of IJCAI-85*.

Rumelhart, D.E., (1976), Toward an Interactive Model of Reading, *Center for Human Info. Proc. CHIP-56*, UC San Diego, La Jolla, CA.

Shastri, L. and Feldman, J.A., (1984), Semantic Networks and Neural Nets, TR-131, Computer Science Dept., The University of Rochester, Rochester, NY, June.