Distributed reasoning and learning in Bayesian expert systems*

F. J. Díez J. Mira

Dpto. Informática y Automática. UNED
Senda del Rey. 28040 Madrid. Spain

Abstract

This paper presents Bayesian networks as a framework for distributed reasoning in expert systems. We discuss methods for evidence propagation, for learning, with emphasis on sequential learning, and for generating linguistic explanations. When a parallel implementation is possible, we describe the computational power, i.e. the information that must be stored and the local calculations that must be performed at every node, in order to get a distributed expert system. Finally, a brief comparison to neural networks is offered.

1 Introduction

1.1 Historical development

In the 1950s, Newell, Shaw and Simon developed the first AI-programs, based on a logic approach. Their most renowned programs were the Logic Theorist and the General Problem Solver (GPS) [56, 57]. Nevertheless, the initial optimism in the new-born field of Artificial Intelligence was soon frustrated, specially since the famous failure of machine translation projects. After 20 years of research, it became apparent that the power of an intelligent program depends more on the specific knowledge it possesses about a particular domain than on the reasoning strategies it employs. The basic idea of the expert system paradigm was to separate the domain information (knowledge base) from the reasoning strategies (inference engine).

Since first order logic turned out to be untractable, even for relatively small models, predicate calculus was substituted by the rule-based formalism [6], thus sacrificing expressiveness as a trade for efficiency.

Another shortcoming of a pure logic approach is the treatment of the uncertainty that arises from the limitations of the model (lack of precision in the knowledge obtained, undeterministic relations among variables) as well as from the inaccuracy and incompleteness of the data at hand. One of the first expert systems, PROSPECTOR [24, 25], used a simplified model of Bayesian inference. Nevertheless, the inconsistencies of the

*To appear in: C. A. Ntuen, editor. *Advances in Fault-Diagnosis Problem Solving.* CRC Press, Boca Raton, FL.

model, the unjustified assumption of conditional independence and other considerations on efficiency originated severe criticism about the adequacy of probability as a framework for reasoning in AI [53, 76]. This led to the development of *ad hoc* techniques for dealing with uncertainty. The most famous of them was MYCIN's certainty factor model [73], originally designed for therapy recommendation, which was later employed in different domains as the core of an expert system building tool, EMYCIN [77].

Despite the empirical success of MYCIN, whose competence was comparable to that of human experts, some doubts arose about the correctness of the underlying mathematical model [1]. Heckerman and Horvitz, in different works that appeared in the 1980's, clearly showed the deficiencies and inconsistencies of certainty factors [31, 43, 34, 35]. They concluded that the error was to extend a logic-like formalism from certain to uncertain reasoning [34]. The arguments in [35] and [55, pp. 125ff] criticizing the shortcomings of rule-based systems, can be generalized to all extensional systems, including fuzzy-logic approaches and most of the systems based on Dempster-Shafer theory, where dependencies and independencies (in the sense of probability theory) are not explicitly shown [62, pp. 4–12].

In those years, Cheeseman [7, 8] passionately advocated the use of probability as the most accurate representation scheme for uncertain reasoning. In the strong controversy of that period, the main objection against probability was that it was untractable even for relatively small domains. The only solution seemed to be the introduction of unjustified assumptions of conditional independence, which undermined the most important argument for probability: its theoretical soundness.

The situation took a favorable turn for the side of probability with the apparition of Bayesian networks in this scenario. Although they had been independently discovered by researchers from different fields, they were born to the field of Artificial Intelligence in 1986, after a four-year gestation [59, 49, 48], with the publication of the famous paper by Pearl [60], where an efficient algorithm for singly-connected networks was presented. Their consolidation took place two years later, with the book *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* [62] and the algorithm by Lauritzen and Spiegelhalter [51] for dealing with general networks.

The main advantage of Bayes networks is that they explicitly represent all dependencies and conditional independencies and, by default, all the other independencies. In fact, the name of *independence networks* has been suggested for BNs, in order to emphasize their most essential feature [55]. Other usual names are belief networks [60, 62] and causal probabilistic networks [4, 10, 51]. Influence diagrams are an extension of Bayes networks including decision nodes and value nodes [44, 72].

Besides the theoretical arguments cited above, there is also empirical research suggesting that BNs have several advantages, both in terms of the time to build the expert system and of the accuracy of the results [32, 40, 47, 79]. The debate among followers of the different approaches for uncertain reasoning is far from reaching a conclusion. It is only possible to say that probability is again considered as a powerful and efficient scheme for artificial reasoning and is gaining more and more supporters who defend it as the best framework for dealing with uncertainty. The different subfields of Bayes networks are active areas of research all over the world.

1.2 Bayesian networks for diagnosis

Early systems for computer aided diagnosis [14, 30] were based on Bayesian theory and decision analysis. After the above mentioned period in which the rule-based approach was preferred over purely numerical methods, there is again nowadays an increasing interest in applying probabilistic reasoning to diagnosis problems. The systems developed at Aalborg University, mainly HUGIN [2] and MUNIN [4], and the Pathfinder Project of Stanford University [36, 37] are the most widely known systems based on Bayes networks and have already matured into commercial products. Other applications, most of them in the field of medicine, can be found in [3, 5] and references therein.

From the point of view of diagnosis, Bayesian networks are related to those systems based on deep knowledge, since they are designed as a causal model, which must be as detailed as possible in order to justify the assumptions of conditional independence. In this sense, BNs are a form of *diagnosis from first principles*, but differ from the schemes that follow a logic approach [13, 15, 29, 67]. These schemes assume there is a deterministic description of the system to be diagnosed. BNs, on the other hand, can naturally tackle with probabilistic relationships. Furthermore, they are able to take into account *a priori* probabilities, which are a valuable information used —often unconsciously— by human experts [26, 19].

If a node in the network represents a binary variable corresponding to the presence of an anomaly, we can diagnose that it is present when its probability exceeds a certain threshold. Due to the propagation of new evidence, probability can cross the threshold several times, so that Bayes networks constitute a kind of non-monotonic reasoning, based on probability rather than on logic. Clearly, several variables corresponding to anomalies may exceed the threshold, so multiple-fault diagnosis appears as a natural capability of Bayesian expert systems.

Diagnosis in BNs is performed by clamping evidence nodes (in general corresponding to different information sources) to its observed values. Then, evidence is propagated through the algorithms in section 2. Unlike rule-based expert systems, in which reasoning is unidirectional, in BNs evidence propagates from causes to effects and vice versa.[1] For this reason, BNs more closely mimic human reasoning, which is able to modify the belief of a hypothesis from both causal and evidential support. It is not surprising, since associative networks were introduced in Artificial Intelligence as an attempt to model human memory [65].

The main shortcoming of BNs is that they must be represented by *acyclic* directed graphs. This constitutes a limitation as to the models which *can* be represented in this formalism. Accordingly, they are not suited for domains where feedback plays an important role. A related drawback is that, despite recent advances (see some papers in [23, 33]), it is still difficult to implement temporal reasoning in BNs.

1.3 Overview

The purpose of this paper is to present BNs from the viewpoint of distributed computation. In fact, Pearl says [63]:

[1] Although in many systems rules can be chained backwards and forwards, reasoning is unidirectional because the consequent is asserted when premises are true, there is never the assertion of one of the premises from the other clauses.

> The idea of studying distributed probabilistic computations on graphical models began brewing in my mind in the late 1970s, after I read Rumelhart's paper on reading comprehension [68]. In this paper, Rumelhart presented compelling evidence that text comprehension must be a distributed process that combines both top-down and bottom-up inferences. [...] I thus began to explore the possibility of achieving distributed computation in a "pure" Bayesian framework.

Our objective is not to make an exhaustive review of the field, but rather to show how knowledge representation and reasoning (evidence propagation), as well as explanation and learning (parameter adjustment) can be performed in a distributed fashion having a processor for every node.

In order to structure the presentation, we will study each problem at the three levels defined by David Marr [52, section 2.1]: **theory**, **algorithm** and **implementation**. In the case of BNs, the theory of probabilistic inference is given by the definitions and the axioms of independence introduced in section 2.1. Some algorithms derived from the theory are presented in sections 2.2 and 2.4. All those algorithms admit parallel implementations, which are the object of this paper. The same three-level scheme is followed in section 3 for studying learning in BNs. Section 4 deals with explanation, but this is still a young field where no theory has yet been established, so we cannot adhere to the same scheme as in previous sections. Finally, a brief comparison with neural networks is shown.

2 Evidence propagation

2.1 Theory of Bayesian networks

This section introduces quite informally the fundamentals of BNs. For a proper presentation, see [62, 55].

In a first approximation, a *Bayes network* can be defined as a directed acyclic graph where nodes represent variables and links represent causal relations. "Acyclic" means that, if there is a loop, its arrows do not form a closed path. The meaning of "causal relations" is given from a mathematical point of view in definition 1. In this context, the terms "node" and "variable" can be used as synonyms. Singly-connected networks are called *polytrees*. We represent a node or variable by a capital letter, X, and its values in lower case, x.

A node X is a *parent* of Y is there is a link from X to Y; conversely, Y is a *child* of X. The set of parents of X is represented as $pa(X)$. The *family* of X is formed by X and its parents. Node U is an *ancestor* of Y if U is a parent of Y or there is a node X such that U is an ancestor of X and X is a parent of Y; conversely, Y is a *descendant* of U. In causal networks, parent is synonym of cause, and child of effect.

A subset of nodes, $\bar{W} = \{W_1, \ldots, W_N\}$, *separates* variables X and Y if the instantiation of the variables in the subset, $\bar{w}$, makes X and Y conditionally independent:

$$\forall x, \forall y, \forall \bar{w},\ P(y|\bar{w}) \neq 0 \Longrightarrow P(x|\bar{w}, y) = P(x|\bar{w}).$$

Observe that this definition is symmetric for X and Y. It can be immediately extended from single variables, such as X and Y, to subsets of them, $\bar{X}$ and $\bar{Y}$.

Definition 1 (Bayesian network) *Given a connected directed acyclic graph and a joint probability distribution for the variables it represents, they are said to constitute a* Bayes network *if the set of parents of any node X, $pa(X)$, separates it from every node that is not one of its descendants.*

2.2 Algorithm for polytrees

Our objective is to find the *evidential probability* $P^*(x)$, defined as the probability of proposition "The value of variable X is x" given the observed evidence $\mathbf{e}$:

$$P^*(x) \equiv P(X = x|\mathbf{e}) \tag{1}$$

In the case of a singly-connected network, an arbitrary node X divides this evidence into that coming from its causes, $\mathbf{e}_X^+$, and that coming from its effects, $\mathbf{e}_X^-$. Accordingly, we can define

$$\pi(x) \equiv P(x, \mathbf{e}_X^+) \tag{2}$$

and

$$\lambda(x) \equiv P(\mathbf{e}_X^- \,|\, x) \tag{3}$$

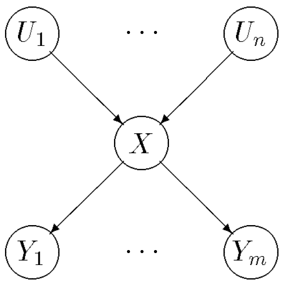

Figure 1: Parents and children of X.

If the causes of X are $U_1, \ldots, U_n$, and its effects $Y_1, \ldots, Y_m$ (see fig. 1), we introduce

$$\pi_X(u_i) \equiv P(u_i, \mathbf{e}_{U_i X}^+) \tag{4}$$

and

$$\lambda_{Y_j}(x) \equiv P(\mathbf{e}_{XY_j}^- \,|\, x) \tag{5}$$

where $\mathbf{e}_{XY}^+$ / $\mathbf{e}_{XY}^-$ represents the evidence above / below link $X \rightarrow Y$.

The axioms of independence for BNs allow us to compute these expressions easily. First, we have

$$\begin{aligned} P^*(x) &= \alpha \; P(x, \mathbf{e}_X^+, \mathbf{e}_X^-) \\ &= \alpha \; P(x, \mathbf{e}_X^+) \; P(\mathbf{e}_X^- \,|\, x) \\ &= \alpha \; \pi(x) \, \lambda(x) \end{aligned} \tag{6}$$

where $\alpha \equiv [P(\mathbf{e})]^{-1}$ is a normalization constant.

The causes of X have no common ancestor (we are in a singly-connected network), so they are *a priori* independent and in consequence

$$\begin{aligned}\pi(x) &= \sum_{u_1,\dots,u_n} P(x|\,u_1,\,\dots,\,u_n)\prod_{i=1}^{n} P(u_i|\,\mathbf{e}^+_{U_iX}) \\ &= \sum_{\bar{u}} P(x|\,\bar{u})\prod_{i=1}^{n}\pi_X(u_i) \end{aligned}\tag{7}$$

Conditional independence on X leads to

$$\begin{aligned}\lambda(x) &= P(\mathbf{e}^-_{XY_1},\,\dots,\,\mathbf{e}^-_{XY_m}|\,x) \\ &= \prod_{j=1}^{m} P(\mathbf{e}^-_{XY_j}|\,x) = \prod_{j=1}^{m}\lambda_{Y_j}(x)\end{aligned}\tag{8}$$

Similarly, node X separates Y_j from other effects and from causes of X. Therefore, after definition (4),

$$\begin{aligned}\pi_{Y_j}(x) &= P(x,\mathbf{e}^+_{XY_j}) = P(x,\mathbf{e}^+_X,\mathbf{e}^-_{XY_k}\ k\neq j) \\ &= \pi(x)\prod_{k\neq j}\lambda_{Y_k}(x)\end{aligned}\tag{9}$$

In order to calculate $\lambda_{Y_j}(x)$, defined in eq. (5), let us denote by $\bar{V}$ the set of causes of Y_j different from X. We then have

$$\begin{aligned}\lambda_{Y_j}(x) &= \sum_{y_j,\bar{v}} P(\mathbf{e}^-_{Y_j},\mathbf{e}^+_{VY_j},y_j,\bar{v}|\,x) \\ &= \sum_{y_j}\left[P(\mathbf{e}^-_{Y_j}|\,y_j)\sum_{v} P(y_j|\,\bar{v},x)\,P(\bar{v},\mathbf{e}^+_{VY_j})\right]\end{aligned}$$

The causes of Y_j are *a priori* independent. In consequence,

$$\lambda_{Y_j}(x) = \sum_{y_j}\left[\lambda(y_j)\sum_{\bar{v}} P(y_j|\,x,\bar{v})\prod_{k=1}^{p}\pi_{Y_j}(v_k)\right]\tag{10}$$

The ten equations above display five definitions and the corresponding formulas to compute them. According to these formulas, the only data needed by a general node are the probability of every value of the variable for every instantiation of its causes, $P(x|\,u_1,\,\dots,\,u_n)$. Instead, for a top node having no causes explicit in the model, the *a priori* probability $P(x)$ must be provided. A node Z without descendants must be assigned a vector $\lambda(z)$: if there is no observed evidence for this node, its initial value is a constant vector; otherwise, the observed value is assigned a 1 and the other values of the variable are assigned 0's. This assignment of π's and λ's for top and leaf nodes respectively guarantees a termination condition for the recursive formulas given above.

2.3 Distributed implementation

The algorithm presented above requires that every node stores some information, both static and dynamic. Here, "static" means independent of the observed evidence, such as the structure of the network and the conditional probabilities.

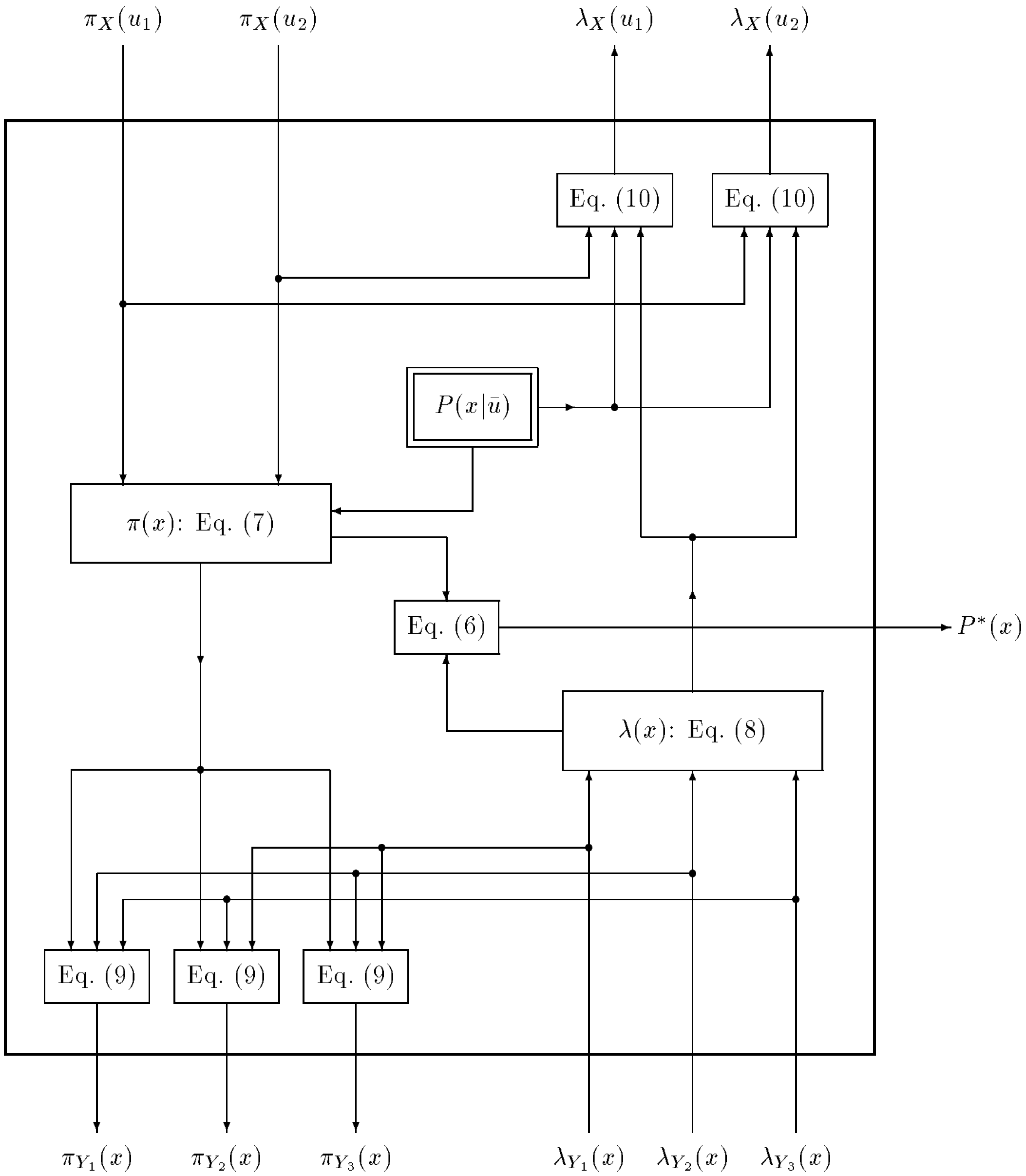

Figure 2: Computations performed at node X.

With regard to the static information, it is first necessary to represent the structure of the network. In a hardware implementation, every link is built as a wire or a bus between the two nodes it connects; in a software implementation, every node has usually two lists of pointers to other nodes: one list for its causes and another one for its effects. Besides, the network must contain the static numerical information, namely the conditional and *a priori* probabilities mentioned in the preceding section.

The dynamic information consists of the π and λ messages of evidence propagation. The most important property derived from the axioms of independence is that, in polytrees, every link decomposes the network into two parts, whose only interaction comes through that link, and the messages between them are orthogonal, i.e. $\pi_Y(x)$ can be computed independently of $\lambda_Y(x)$ and vice versa. In fig. 2, which shows the computation performed at node X, this property becomes apparent as the absence of loops in the information flow. Observe as well in the formulas of the last section that all the information required by a node for computing its messages is locally stored.

A node X will be able to send a message to its neighbor W if and only if X has received the messages from all of its other neighbors. Therefore, it is not necessary to have a global control mechanism and the model can be implemented as an *asynchronous network* in which the number of messages received by a node determines what other messages can be computed and sent.

If the computation at every node is bounded (by limiting the number of parents and values) and the algorithm is implemented sequentially, the time complexity is proportional to the number of nodes. In this case, it is more efficient to perform evidence propagation in two phases: evidence collection toward a pivot node and evidence distribution from this node, as was suggested in [51] for trees of cliques.

If there is a processor for every node, the time complexity of the algorithm is proportional to the maximum length in the network. Anyway, if the conceptual processors (the nodes) are implemented by a smaller number of hardware processors, they are queuing for access to a limited set of computational resources and we encounter the typical problem of parallel programming, i.e., what message has to be computed first in order to get the highest performance.

2.4 Extensions

In the general case, the number of conditional probabilities required by a node grows exponentially with the number of its causes. On the other hand, the OR–gate, introduced in [62, pp. 184–188] and generalized in [39, 17], requires only a number of parameters proportional the number of causes. In this case, a link is no longer a passive channel but an active processor capable of storing and sending information [20].

Exact inference in general networks poses an NP-hard problem because of the presence of loops [11]. The most widely used algorithm for general networks is known as clique-tree propagation [51, 46]. Computations in the resulting tree are very similar to the case of polytrees, and again a distributed implementation is possible.

Another method is local conditioning [16], which consists of removing some edges from the original network and assigning to every node the conditioning variables which break the loops. This is the only known exact algorithm which allows a distributed implementation with a processor for every node [20].

The complexity of exact inference fostered the development of simulation methods [38, 61]. However, it has been shown recently that "approximating probabilistic inference in Bayesian belief networks is NP-hard" as well [12]. Anyway, stochastic simulation is still a useful method for practical applications, in which all the processing can be naturally performed in parallel. In [20] we discuss with more detail the distributed implementation of these algorithms.

3 Learning

One of the bottlenecks of expert systems building is knowledge acquisition. In this process, there are two phases: model construction and model refinement. The objective is to build a network that, based on the observed cases, gives the optimal prediction for future ones. Fortunately, in this field there is a normative **theory** for learning, provided again by Bayesian analysis: we look for the most probable BN given the observed cases.

The first step is to choose the variables and to determine the qualitative relations among them by drawing causal links. Then, quantitative information will be obtained in the form of conditional probabilities.

The optimal situation occurs when there is an available database containing a big number of cases with the values of all variables deterministically specified. Efficient algorithms for eliciting both the structure and the parameters of the network have been developed. This process performed on a database is called *batch learning*, and allows the automated discovery of causal relationships.

In the batch-learning methods, there is usually a first phase in which some recount is performed for every node in order to know how often each configuration of the variables is present in the database, and then a global computation of entropy is carried out. In consequence, these methods are inherently global and cannot be distributed.[2]

There is another shortcoming: if the cases contained in the database are incomplete, the mathematical framework becomes complex and no definitive theory of learning is established yet. Very often, there is not even an incomplete database, and the knowledge engineer must resort to human experts who make use of their memory and of the literature published in their field. In this situation, the model needs refinement, and it is desirable to endow the system with some capability of adaptation as it executes. This activity is called *sequential learning*.

The first work on this topic was done by Spiegelhalter and Lauritzen [74]. Their work starts from assuming a prior distribution in which all the parameters of the model are pairwise independent. When a new case is examined by the network, parameters are updated. Unfortunately, if the new case is not complete, i.e. if some variables are not assigned a value, dependencies between parameters will in general emerge. However, the only known tractable solution is to ignore such dependencies and keep just an updated probability distribution for each node, thus allowing a sequential treatment of data.

The appeal of this method is that the probability distribution $P(\theta)$ of a parameter that determines the conditional probability of the family of X, $P(x|u,\theta)$, can be stored in node X and the updating is based just on the probability resulting for this family after evidence propagation. In other words, storage and sequential updating of parameters can

[2] Nevertheless, Laskey [50] and Wen [78] have proposed the use of Boltzmann machines [69] for building Bayes networks (see section 5).

be carried out naturally in a completely-distributed fashion, as shown below.

Spiegelhalter and Lauritzen proposed three models of probability distribution for the parameters. A forth model, proposed in [17], assumes a prior normal distribution $\mathcal{N}(\mu_x^u, \sigma_x^u)$ for every parameter $\theta_x^u \equiv P(x|u)$. The normal distribution can be updated by the following **algorithm**:

$$\mu_x^{u\prime} = E'(\theta_x^u) = \mu_x^u + \Delta_x^u \tag{11}$$

$$\sigma_x^{u\prime 2} = E'(\theta_x^{u\,2}) - \mu_x^{u\prime 2} = \sigma_x^{u\,2} - \Delta_x^{u\,2} \tag{12}$$

where

$$\Delta_x^u = (\sigma_x^u)^2 \, \frac{\left((\hat{\lambda}(x) - \hat{\lambda}(x_0)\right) \prod_i \hat{\pi}_X(u_i)}{\sum_{x'} \hat{\lambda}(x')\hat{\pi}(x')} \, . \tag{13}$$

Here, $\hat{\pi}(x)$, $\hat{\pi}_X(u_n)$ and $\hat{\lambda}(x)$ are the usual messages, computed by using the average values $\hat{P}(x|u) = \mu_x^u$ for every conditional probability, instead of the Gaussian distribution. For instance, the expression corresponding to eq. (7) turns out as

$$\hat{\pi}(x) = \sum_u \mu_x^u \prod_i \hat{\pi}_X(u_i). \tag{14}$$

Comments in [17] show the agreement between these equations and the intuitively expected results.

Distributed implementation. Fig. 3 shows the process of learning at node X. This scheme should be embedded in fig. 2. Observe that all the inputs appearing in eq. (13) are locally accessible to or stored at node X. Therefore, it is straightforward to get a distributed implementation. In this figure, we can see the feed-back typical of learning processes.

We have studied here the general case. A similar formalism for the generalized noisy OR–gate can also be found in [20].

Observe that the network is learning as it executes. Unlike backpropagation for neural networks, which separates training from execution, this model naturally integrates both processes.

4 Linguistic explanation

One of the features that distinguish an expert systems from other diagnostic programs is its ability to explain its results [9]. Therefore, a BN cannot be considered as an expert system until it includes some explanation facility. Nevertheless, explanation in BNs is still a young field and there is a lot to be done.

Unlike evidence propagation or learning, which are well defined for BNs, **no theory** of explanation is established yet. Furthermore, the first question to be approached is just to define the problem: What kind of explanation does the user want? What kind of explanation can a BN offer? Suermondt [75, chapter 2] presents some considerations on explanation from an epistemological point of view.

In earlier Bayesian diagnosis systems, it was possible to generate an explanation of the conclusions [27, 58, 64, 66]. Nevertheless, the lack of distinction between the model and

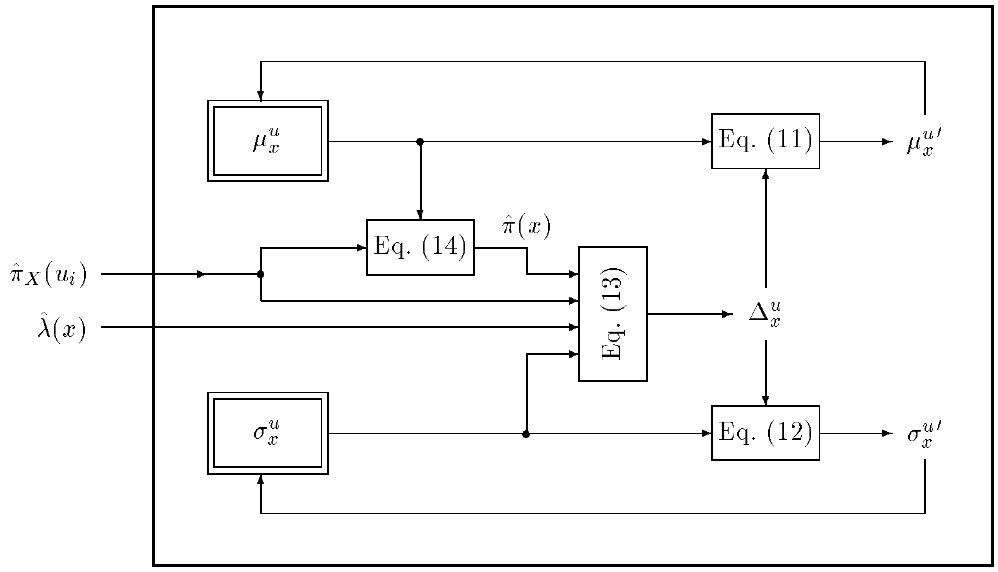

Figure 3: Learning at node X.

the inference mechanism precluded an explanation of the underlying model. In knowledge-based systems, and in particular in BNs, there are two possible types of explanation: *model explanation*, which consists of displaying the static causal knowledge embedded in the network, and *inference explanation*, also termed dynamic explanation [41], which shows how the observed evidence has modified the probability distribution. The second type must be based on the first.

In BNs, there is another classification orthogonal to the previous one, which considers two levels of explanation: micro and macro [70]. The micro level focuses on local propagation of evidence between neighbor variables. The macro level follows the main lines of reasoning over the network as a whole or over certain paths.

In the micro level, Elsaesser and Henrion [27, 28] describe how to relate numbers to qualitative explanations. Sember and Zuckerman [71] offer a method for explaining the changes in the causal and the evidential support of a node. Since only local operations are performed, the micro level poses no problem with regard to a parallel implementation.

A different question is the macro level. One of the proposed approaches is the INSITE methodology [75, chapter 4], based on measuring the influence of observed evidence and the influence of chains of nodes, through a cost-function—cross-entropy by default. Although a large part of the calculations can be performed at each node, the process is basically global and does not admit a parallel implementation.

Two other methods have been developed by Druzdzel and Henrion. One of them is based on *scenarios*, composed of "selected outcomes of relevant variables ordered in the causal direction" [21]. Since this is also a global approach which does not consider the propagation of evidence, it cannot be implemented in a distributed fashion either. *Qualitative inference*, on the other hand, examines the impact of the propagation of evidence from a variable to its neighbors [41]. There are three types of interaction: predictive (from cause to effect), diagnostic (from effect to cause) and intercausal (between two causes of an effect). An elegant **distributed capability of explanation** will emerge, but the method is still under development [22].

In [18, chapter 8] we describe a different approach that includes both static and dynamic explanation. It has been used in the construction of DIAVAL, an expert system for echocardiography, although the essence of the method is applicable to other fields different from medicine. The start point consists of distinguishing two types of interaction (the general model and the OR–gate) and six classes of links: influence, parameter ("X is measured as Y"), observation, type-of, causality ("X produces Y") and manifestation ("X is characterized by Y"). There are several options for every node; some of them explain the *model* (values, causes, effects, prevalence, conditional probability, etc.) while others explain the *inference* (probability *a posteriori*, probability that each U_i has produced X, etc.), and there are similar options for the links involved in an OR–gate. Fig. 4 shows the effects of the rheumatic mitral stenosis; by selecting one of them, the user can obtain more information and further navigate across the network. At any given moment in the consultation, the attention focuses on a node or on a link, and all the information displayed to the user is located at that point or in its nearest neighborhood. Therefore, this model lends itself directly to a **distributed implementation**, although in this case parallelism does not improve efficiency, since at any time only one of the processors would be active.

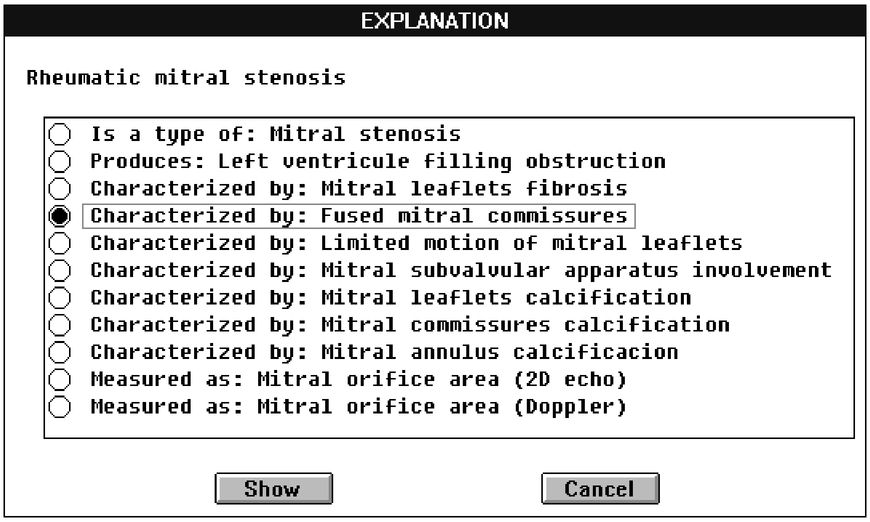

Figure 4: Explanation screen for the effects of a node.

5 Comparison to neural networks

Unlike the case of Bayesian networks, in which the model is precisely defined by a set of axioms, the neural networks (NNs) paradigm includes many different models [69]. An exhaustive comparison of both schemes is beyond the scope of this paper and will constitute the subject of a future work. We will now offer only some outlines.

From the point of view of evidence propagation, a node in a Bayes network can be considered as a complex neuron which receives information from and transmits it to its neighboring "neurons". Nevertheless, in BNs the message transmitted from X to

Y depends on the information collected from all the other neighbors of X, and so in BNs it is not possible to make a distinction between dendritic and axonic fields, since every node interchanges messages with all its neighbors. Therefore, BNs are similar to Hopfield nets [42][3] but differ from other schemes such as those based on backpropagation [69]. Furthermore, although propagation is bidirectional in BNs, the defining axioms preclude the possibility of feedback (remember that algorithms for propagation in BNs must somehow reduce or cut the loops), while feedback plays a crucial role in advanced NN models.

Anyway, there is a much deeper difference. A network representation is said to be *localist* if there is a one-to-one relationship between nodes and concepts. In a *distributed representation*, on the contrary, a pattern (a concept) is stored in the set of weights as a whole and cannot be associated to a particular node or link; the reciprocal property, called superimposition, means that a node intervenes in representing several concepts. For instance, a stable state of a Hopfield network represents one of the patterns it can recognize. Most NNs models use a distributed representation, whereas BNs are clearly localist: every node represents a variable (a concept, in the last term) and information is exactly located: every probability table is stored in a node or a link. Distributed representations have the advantage of a higher fault-tolerance, while in a BN the removal of some nodes would eliminate the possibility of introducing some findings and/or diagnosing some anomalies. On the other hand, a BN has the capability of explaining how the evidence transmitted has influenced the resulting diagnosis.

Finally, from the point of view of learning, the framework for parameter adjustment presented in section 3 is designed to perform sequentially as the system is diagnosing new cases. This is a difference from backpropagation models, which separate training from execution. Nevertheless, there are NN models with local capacity of learning [54] and even proposals to use Boltzmann machines for constructing Bayes networks [50, 78].

6 Conclusion

In this paper, we have tried to show that distributed computation is possible for Bayesian expert systems. In every case (inference, learning and explanation) we have carried out our study at three levels: theory, algorithm and implementation. In singly-connected networks, evidence propagation can be naturally performed in a completely distributed fashion (local computation with local control). Other methods for dealing with loops, such as clustering and local conditioning, carry out a preprocessing of the network in order to build a tree, so that propagation becomes similar to that for singly-connected BNs.

Batch learning is inherently a global process and cannot be distributed. On the other hand, there exist frameworks for sequential learning that perform only local operations. In a similar fashion, some models for explanation perform global computations on the network as a whole, while other approaches allow a distributed implementation.

Discussions in this paper were based on some of the algorithms and models currently available. Our purpose was not, however, to study particular methods but rather to show the possibilities that the BN paradigm offers for distributed computation.

[3] There is a closer similarity when evidence is propagated in BNs by stochastic simulation [20]. Under certain conditions, this similarity becomes equivalence [45].

Acknowledgements

This research was supported by a grant from the Plan Nacional de Formación de Personal Investigador of the Spanish Ministry of Education and Science and benefited from the stay of one of the authors (F. J. D.) at the Computer Science Department, University of California Los Angeles, with research facilities provided by Prof. Judea Pearl.

References

[1] J. B. Adams. Probabilistic reasoning and certainty factors. In B. G. Buchanan and E. H. Shortliffe, editors, *Rule-based expert systems: The MYCIN experiments of the Stanford Heuristic Programming Project*, chapter 12, pages 263–271. Addison-Wesley, Reading, MA, 1984.

[2] S. K. Andersen, K. G. Olesen, F. V. Jensen, and F. Jensen. HUGIN: A shell for building Bayesian belief universes for expert systems. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI-89)*, pages 1080–1085, Detroit, MI, 1989.

[3] S. Andreasen, F. V. Jensen, and K. G. Olesen. Medical expert sustems based on causal probabilistic networks. *International Journal of Biomedical Computing*, 28:1–30, 1991.

[4] S. Andreassen, M. Woldby, B. Falck, and S. K. Andersen. MUNIN — A causal probabilistic network for interpretation of electromyographic findings. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence (IJCAI–87)*, pages 366–372, Milan, 1987.

[5] R. Bellazzi, S. Quaglini, C. Berzuini, and M. Stefanelli. GAMEES: A probabilistic environment for expert systems. *Computer Methods and Programs in Biomedicine*, 35:177–191, 1991.

[6] B. G. Buchanan and E. H. Shortliffe, editors. *Rule-based expert systems: The MYCIN experiments of the Stanford Heuristic Programming Project.* Addison-Wesley, Reading, MA, 1984.

[7] P. Cheeseman. In defense of probability. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence (IJCAI–85)*, pages 1002–1009, Los Angeles, CA, 1985.

[8] P. Cheeseman. An inquiry into computer understanding. *Computational Intelligence*, 4:58–66, 1988.

[9] W. J. Clancey. The epistemology of rule-based expert systems — A framework for explanation. *Artificial Intelligence*, 20:215–251, 1983.

[10] G. F. Cooper. *NESTOR: A Computer-Based Medical Diagnostic Aid that Integrates Causal and Probabilistic Knowledge.* PhD thesis, Dept. of Computer Science, Stanford University, STAN–CS–84–1031, 1984.

[11] G. F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.

[12] P. Dagum and M. Luby. Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artificial Intelligence*, 60:141–153, 1993.

[13] R. Davis and B. Buchanan. Meta-level knowledge. In B. G. Buchanan and E. H. Shortliffe, editors, *Rule-based expert systems: The MYCIN experiments of the Stanford Heuristic Programming Project*, chapter 28, pages 507–530. Addison-Wesley, Reading, MA, 1984.

[14] F. T. de Dombal, J. R. Leaper, J. R. Staniland, A. McCann, and J. Horrocks. Computer-aided diagnosis of acute abdominal pain. *British Medical Journal*, 2:9–13, 1972.

[15] J. de Kleer and B. C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32:97–130, 1987.

[16] F. J. Díez. Local conditioning in Bayesian networks. Technical Report (R–181), Cognitive Systems Laboratory, University of California, Los Angeles, 1992. Submitted to *Artificial Intelligence*.

[17] F. J. Díez. Parameter adjustement in Bayes networks. The generalized noisy OR–gate. In *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, pages 99–105, Washington D.C., 1993. Morgan Kaufmann, San Mateo, CA.

[18] F. J. Díez. *Sistema Experto Bayesiano para Ecocardiografía.* PhD thesis, Dept. Informática y Autom'atica, UNED, Madrid, 1994.

[19] F. J. Díez and J. Mira. Causal Bayesian reasoning in medicine. *Cybernetics and Systems*, 23:417–429, 1992.

[20] F. J. Díez and J. Mira. Distributed inference in Bayesian networks. *Cybernetics and Systems*, 25:39–61, 1994.

[21] M. J. Druzdzel and M. Henrion. Using scenarios to explain probabilistic inference. In *Proceedings of the AAAI-90 Workshop on Explanation*, pages 133–141, Boston, MA, 1990.

[22] M. J. Druzdzel and M. Henrion. Efficient reasoning in qualitative probabilistic networks. In *Proceedings of the 11th National Conference on AI (AAAI-93)*, Washington D.C., 1993.

[23] D. Dubois, M. P. Wellman, B. D'Ambrosio, and P. Smets, editors. *Proceedings of the 8th Conference on Uncertainty in Artificial Intelligence*, Stanford University, 1992. Morgan Kaufmann, San Mateo, CA.

[24] R. O. Duda, J. Gaschnig, and P. E. Hart. Model design in the PROSPECTOR consultant program for mineral exploration. In D. Michie, editor, *Expert Systems in the Microelectronic Age*, pages 153–167. Edinburgh University Press, Edinburgh, 1979.

[25] R. O. Duda, P. E. Hart, and N. Nilsson. Subjective Bayesian methods for rule-based inference systems. *Proceedings of the 1976 National Computer Conference (AFIPS)*, 45:1075–1082, 1976.

[26] D. M. Eddy. Probabilistic reasoning in clinical medicine: Problems and opportunities. In D. Kahneman, P. Slovic, and A. Tversky, editors, *Judgement under uncertainty: Heuristics and biases*, chapter 18, pages 249–267. Cambridge University Press, Cambridge, UK, 1982.

[27] C. Elsaesser. Explanation of probabilistic inference. In L.N. Kanal, T.S. Levitt, and J.F. Lemmer, editors, *Uncertainty in Artificial Intelligence 3*, pages 387–400. Elsevier Science Publishers, Amsterdam, 1989.

[28] C. Elsaesser and M. Henrion. Verbal expressions for probability updates: How much more probable is "much more probable"? In M. Henrion, R. D. Shachter, L.N. Kanal, and J.F. Lemmer, editors, *Uncertainty in Artificial Intelligence 5*, pages 319–328. Elsevier Science Publishers, Amsterdam, 1990.

[29] M. R. Genesereth. The use of design descriptions in automated diagnosis. *Artificial Intelligence*, 24:411–436, 1984.

[30] G. A. Gorry and G. O. Barnett. Experience with a model of sequential diagnosis. *Computers and Biomedical Research*, 1:490–507, 1968.

[31] D. Heckerman. Probabilistic interpretations for MYCIN's certainty factors. In L. N. Kanal and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence*, pages 167–196. Elsevier Science Publishers, Amsterdam, 1986.

[32] D. Heckerman. An empirical comparison of three inference methods. In R. D. Shachter, T.S. Levitt, L.N. Kanal, and J.F. Lemmer, editors, *Uncertainty in Artificial Intelligence 4*, pages 283–302. Elsevier Science Publishers, Amsterdam, 1990.

[33] D. Heckerman and A. Mamdani, editors. *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, Washington, D.C., 1993. Morgan Kaufmann, San Mateo, CA.

[34] D. E. Heckerman and E. J. Horvitz. On the expresiveness of rule-based systems for reasoning with uncertainty. In *Proceedings of the 6th National Conference on AI (AAAI-87)*, pages 121–126, Seattle, WA, 1987.

[35] D. E. Heckerman and E. J. Horvitz. The myth of modularity in rule-based systems for reasoning with uncertainty. In J. F. Lemmer and L. N. Kanal, editors, *Uncertainty in Artificial Intelligence 2*, pages 23–34. Elsevier Science Publishers, Amsterdam, 1988.

[36] D. E. Heckerman, E. J. Horvitz, and B. N. Nathwani. Toward normative expert systems: Part I — The Pathfinder Project. *Methods of Information in Medicine*, 31:90–105, 1992.

[37] D. E. Heckerman and B. N. Nathwani. Toward normative expert systems: Part II — Probability-based representations for efficient knowledge acquisition and inference. *Methods of Information in Medicine*, 31:106–116, 1992.

[38] M. Henrion. Propagation of uncertainty by logic sampling in Bayes' networks. In J. F. Lemmer and L. N. Kanal, editors, *Uncertainty in Artificial Intelligence 2*, pages 149–164. Elsevier Science Publishers, Amsterdam, 1988.

[39] M. Henrion. Some practical issues in constructing belief networks. In L. N. Kanal, T. S. Levitt, and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence 3*, pages 161–173. Elsevier Science Publishers, Amsterdam, 1989.

[40] M. Henrion and D. R. Cooley. An experimental comparison of knowledge engineering for expert systems and for decision analysis. In *Proceedings of the 6th National Conference on AI (AAAI-87)*, pages 471–476, Seattle, WA, 1987.

[41] M. Henrion and M. J. Druzdzel. Qualitative propagation and scenario-based schemes for explaining probabilistic reasoning. In P. P. Bonissone, M. Henrion, L. N. Kanal, and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence 6*, pages 17–32. Elsevier Science Publishers, Amsterdam, 1991.

[42] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. In *Proceedings of the National Academy of Sciences USA*, pages 2554–2558, 1982.

[43] E. Horvitz and D. Heckerman. The inconsistent use of measures of certainty in Artificial Intelligence research. In L. N. Kanal and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence*, pages 137–151. Elsevier Science Publishers, Amsterdam, 1986.

[44] R. A. Howard and J. E. Matheson. Influence diagrams. In R. A. Howard and J. E. Matheson, editors, *Readings on the Principles and Applications of Decision Analysis*, pages 719–762. Strategic Decisions Group, Menlo Park, CA, 1984.

[45] T. Hrycej. Common features of neural-network models of high and low level human information processing. In T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, editors, *Artificial Neural Networks*, pages 861–866. Elsevier Science Publishers, Amsterdam, 1991.

[46] F. V. Jensen, K. G. Olesen, and S. K. Andersen. An algebra of Bayesian belief universes for knowledge-based systems. *Networks*, 20:637–660, 1990.

[47] J. Kalagnanam and M. Henrion. A comparison of decision analysis and expert rules for sequential diagnosis. In R. D. Shachter, T.S. Levitt, L.N. Kanal, and J.F. Lemmer, editors, *Uncertainty in Artificial Intelligence 4*, pages 271–281. Elsevier Science Publishers, Amsterdam, 1990.

[48] J. H. Kim. *CONVINCE: A conversational inference consolidation engine*. PhD thesis, Dept. of Computer Science, University of California, Los Angeles, 1983.

[49] J. H. Kim and J. Pearl. A computational model for combined causal and diagnostic reasoning in inference systems. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence (IJCAI-83)*, pages 190–193, Karlsruhe, Germany, 1983.

[50] K. B. Laskey. Adapting connectionist learning to Bayes networks. *International Journal of Approximate Reasoning*, 4:261–282, 1990.

[51] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B*, 50:157–224, 1988.

[52] D. Marr. *Vision*. Freeman, New York, 1982.

[53] J. McCarthy and P. Hayes. Some phylosophical problems from the standpoint of Artificial Intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, Edinburgh, 1969.

[54] J. Mira, A. E. Delgado, M. Santos, A. P. de Madrid, and J. R. Alvarez. Local learning in networks of universal analogic neurons. *Cybernetics and Systems*, 25:259–273, 1994.

[55] R. E. Neapolitan. *Probabilistic Reasoning in Expert Systems: Theory and Algorithms*. Wiley-Interscience, New York, 1990.

[56] A. Newell and H. A. Simon. GPS, a program that simulates human thought. In E. A. Feigenbaum and J. Feldman, editors, *Computers and Thought*, pages 279–293. McGraw-Hill, New York, 1963.

[57] A. Newell and H. A. Simon. *Human Problem Solving*. Prentice-Hall, Englewood Cliffs, NJ, 1972.

[58] S. W. Norton. An explanation mechanism for Bayesian inference systems. In J. F. Lemmer and L. N. Kanal, editors, *Uncertainty in Artificial Intelligence 2*, pages 165–174. Elsevier Science Publishers, Amsterdam, 1988.

[59] J. Pearl. Reverend Bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the 2nd National Conference on AI (AAAI-82)*, pages 133–136, Pittsburgh, Pennsylvania, 1982.

[60] J. Pearl. Fusion, propagation and structuring in belief networks. *Artificial Intelligence*, 29:241–288, 1986.

[61] J. Pearl. Evidential reasoning using stochastic simulation of causal models. *Artificial Intelligence*, 32:245–257, 1987.

[62] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, 1988. Revised second printing, 1991.

[63] J. Pearl. Belief networks revisited. *Artificial Intelligence*, 59:49–56, 1993.

[64] P. E. Politser. Explanation of statistical concepts: Can they penetrate the Haze of Bayes? *Methods of Information in Medicine*, 2:99–108, 1984.

[65] M. R. Quillian. Word concepts: A theory and simulation of some basic semantic capabilities. *Behavioral Science*, 12:410–430, 1967.

[66] J. A. Reggia and B. T. Perricone. Answer justification in medical decision support systems based on Bayesian classification. *Computers in Biology and Medicine*, 15:161–167, 1985.

[67] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.

[68] D. E. Rumelhart. Toward an interactive model of reading. Technical Report #CHIP-56, University of California, La Jolla, CA, 1976.

[69] D. E. Rumelhart and J. L. McClelland, editors. *Parallel Distributed Processing.* MIT Press, Cambridge, MA, 1986.

[70] P. Sember and I. Zukerman. On the generation of explanations for Bayesian belief networks. In *Proceedings of the 12th Australian Computer Science Conference*, February 1989.

[71] P. Sember and I. Zukerman. Strategies for generating micro explanations for Bayesian belief networks. In *Proceedings of the 5th Workshop on Uncertainty in Artificial Intelligence*, pages 295–302, Windsor, Ontario, 1989.

[72] R. D. Shachter. Evaluating influence diagrams. *Operations Research*, 34:871–882, 1986.

[73] E. H. Shortliffe and B. G. Buchanan. A model of inexact reasoning in medicine. In B. G. Buchanan and E. H. Shortliffe, editors, *Rule-based expert systems: The MYCIN experiments of the Stanford Heuristic Programming Project*, chapter 11, pages 233–262. Addison-Wesley, Reading, MA, 1984.

[74] D. J. Spiegelhalter and S. L. Lauritzen. Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20:579–605, 1990.

[75] H. J. Suermondt. *Explanation in Bayesian Belief Networks.* PhD thesis, Dept. of Computer Science, Stanford University, STAN–CS–92–1417, 1992.

[76] P. Szolovits and S. G. Pauker. Categorical and probabilistic reasoning in medicine. *Artificial Intelligence*, 11:115–144, 1978.

[77] W. van Melle. A domain-independent production-rule system for consultation programs. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI-89)*, pages 923–925, Detroit, MI, 1989.

[78] W. X. Wen and A. Jennings. Parallel distributed belief networks that learn. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91)*, pages 1210–1215, Sydney, Australia, 1991.

[79] B. P. Wise and M. Henrion. A framework for comparing uncertain inference systems to probability. In L. N. Kanal and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence*, pages 69–83. Elsevier Science Publishers, Amsterdam, 1986.