

SRI International



PROBABILISTIC LOGIC

Technical Note 321

February 6, 1984

By: Nils J. Nilsson, Director

Artificial Intelligence Center
Computer Science and Technology Division

This paper is based on a talk given on
December 8, 1983, at a symposium honoring
Marvin Denicoff on the occasion of his
retirement from the Office of Naval Research.

CONTENTS

LIST OF ILLUSTRATIONS	iii
ABSTRACT	1
I INTRODUCTION	2
II NOTATION	3
III ASSIGNING PROBABILITIES TO SENTENCES	7
IV PROBABILISTIC ENTAILMENT	11
V COMPUTATIONS APPROPRIATE FOR SMALL SENTENCE MATRICES	15
VI DEALING WITH LARGE SENTENCE MATRICES	21
VII DISCUSSION AND CONCLUSIONS	24
VIII ACKNOWLEDGEMENTS	26

ILLUSTRATIONS

1	A Semantic Tree for the Sequence $\{A, A \supset B, B\}$	5
2	Consistent Valuation Vectors for the Sequence $\{A, A \supset B, B\}$	6
3	The Region of Consistent Probabilistic Valuation Vectors for the Sequence $\{A, A \supset B, B\}$	9
4	A Semantic Tree for the Sequence $\{(\exists y)A(y),$ $(\forall x)[A(x) \supset B(x)], (\exists z)B(z)\}$	13
5	The Region of Consistent Probabilistic Valuation Vectors for the Sequence $\{(\exists y)A(y),$ $(\forall x)[A(x) \supset B(x)], (\exists z)B(z)\}$	14

ABSTRACT

Because many artificial intelligence applications require the ability to deal with uncertain knowledge, it is important to seek appropriate generalizations of logic for that case. We present here a semantical generalization of logic in which the truth-values of sentences are probability values (between 0 and 1). Our generalization applies to any logical system for which the consistency of a finite set of sentences can be established. (Although we cannot always establish the consistency of a finite set of sentences of first-order logic, our method is usable in those cases in which we can.) The method described in the present paper combines logic with probability theory in such a way that probabilistic logical entailment reduces to ordinary logical entailment when the probabilities of all sentences are either 0 or 1.

I INTRODUCTION

Several artificial intelligence (AI) applications require the ability to deal with uncertain knowledge. For example, in "expert systems" many of the rules obtained from experts, as well as data provided by users, are not known with certainty. Since ordinary logic is so useful in those cases in which this sort of information *is* known, AI researchers have been interested in generalizations of logic that would be appropriate for representing and reasoning with uncertain knowledge.

There is extensive mathematical literature on probabilistic and plausible inference, which we will not review here. (See for example, [14, 2, 8, 17, 4, 15, 1, 18].) One of the early expert systems in AI embodying a technique designed to handle uncertain knowledge was MYCIN [16]. The PROSPECTOR system [5] used a reasoning method based on Bayes' rule and is quite similar to MYCIN. Lowrance and Garvey [12, 13] have adapted Shafer-Dempster theory to AI applications. AI researchers have also investigated methods based on finding maximum-entropy probability distributions [10, 11, 9, 3]. Halpern and Rabin [7] propose a modal logic with a "likelihood operator." Although a number of reasoning methods have been explored in AI, expert-systems practice seems largely based on ad hoc techniques that have little theoretical justification. The goal of the present paper is to contribute to a better theoretical understanding of probabilistic reasoning among AI researchers.

We present a semantical generalization of ordinary logic in which the truth-values of sentences are probability values (between 0 and 1). Our generalization applies to any logical system for which the consistency of a finite set of sentences can be established. (Although we cannot always establish the consistency of a finite set of sentences of first-order logic, our method is usable in those cases in which we can.)

II NOTATION

We will be representing logical sentences (and truth valuations over sets of them) as vectors. We introduce this notation first for the ordinary, non-probabilistic case. We let \mathcal{S} denote a finite sequence of L sentences arranged in arbitrary order:

$$\mathcal{S} = \{S_1, S_2, \dots, S_L\}$$

Let an L -dimensional, binary-component, column vector \mathbf{V} be composed of the truth valuations of the sentences of \mathcal{S} . We say that \mathbf{V} is a (*true-false*) *valuation vector* for \mathcal{S} .

$$\mathbf{V} = \begin{bmatrix} v_1 \\ v_2 \\ \cdot \\ \cdot \\ \cdot \\ v_L \end{bmatrix},$$

where

$$v_i = 1 \text{ if } S_i \text{ has value } \textit{true}$$

$$= 0 \text{ otherwise.}$$

We say that \mathbf{V} is *consistent* if it corresponds to a consistent valuation of the sentences of \mathcal{S} .

Let \mathcal{V} be the set of all consistent valuation vectors for \mathcal{S} . Suppose K is the cardinality of \mathcal{V} . K is bounded from above by 2^L , and will be less than 2^L whenever there exist inconsistent valuations on \mathcal{S} .

To each consistent vector \mathbf{V} there corresponds an equivalence class of interpretations over the sentences in \mathcal{S} . (Each interpretation in a given class yields the same valuation vector.) Alternatively, we can think of each consistent vector \mathbf{V} as corresponding to an equivalence class of *possible worlds* in which the sentences in \mathcal{S} are *true* or not, according to the values of the components of \mathbf{V} .

As an example, consider the sentences

$$\mathcal{S} = \{A, A \supset B, B\}$$

The set \mathcal{V} for these sentences is

$$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

One method for constructing \mathcal{V} is based on developing a binary “semantic tree.” At each node we branch left or right, depending on whether or not we assign one of the sentences in \mathcal{S} a value of *true* or *false*, respectively. Just below the root we branch on the truth-value of S_1 , next on the truth-value of S_2 , etc. Each path in the tree corresponds to a particular assignment of truth-values to the sentences of \mathcal{S} . We check the consistency of the truth-value assignments as we go and close off those paths corresponding to inconsistent valuations. A semantic tree for our example is shown in Figure 1. Closed-off paths are indicated by an \times ; consistent valuation vectors are indicated in columns at the tips of their corresponding paths.

We can now construct an $L \times K$ matrix \mathbf{M} whose columns, in no particular order, are the vectors in \mathcal{V} . The rows of \mathbf{M} can be taken to represent (semantically) the sentences in \mathcal{S} ; the i -th row specifies the truth-values for S_i under all interpretations that lead to consistent valuations over \mathcal{S} . We call \mathbf{M} the

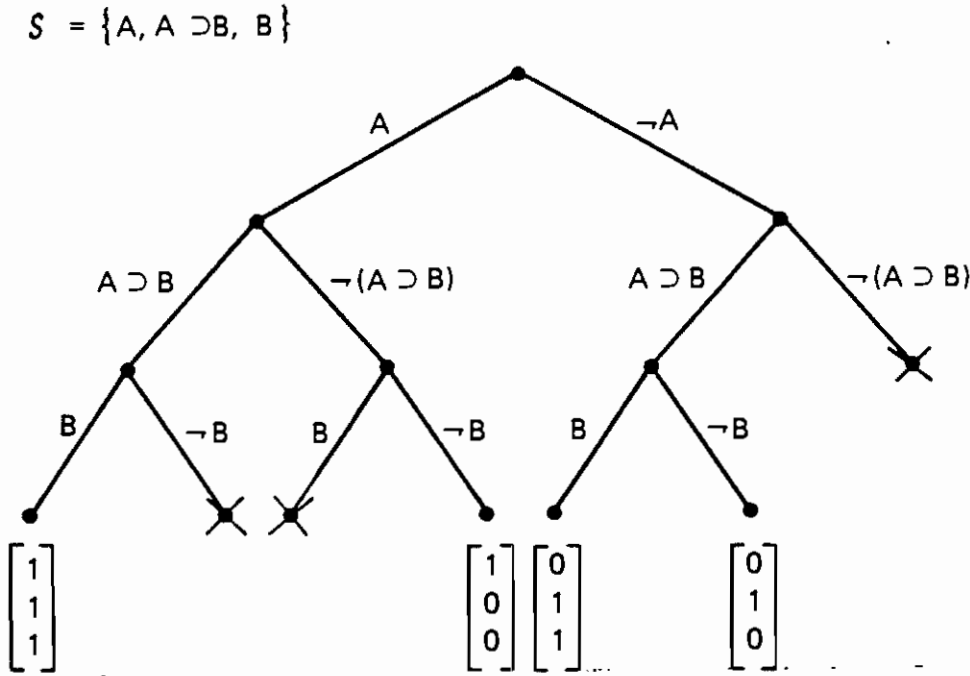


Figure 1: A Semantic Tree for the Sequence $\{A, A \supset B, B\}$

sentence matrix for S . We denote the i -th row of \mathbf{M} by the row vector \mathbf{S}_i . It has 1's in just those positions corresponding to classes of interpretations that satisfy S_i . (Or, we could say, it has 1's in just those positions corresponding to classes of possible worlds in which S_i is *true*.)

Each equivalence class of interpretations can be represented by a K -dimensional unit column vector, \mathbf{P} . The i -th interpretation class is represented by the vector whose i -th component is 1 and all of whose other components are 0. The consistent valuations of the sentences in S induced by some particular interpretation vector \mathbf{P} can be calculated by the matrix equation:

$$\mathbf{M} \mathbf{P} = \mathbf{V}$$

The consistent valuation vectors are a subset of the 2^L vertices of the L -dimensional unit cube in \mathbf{V} -space. In Figure 2 we show this geometric representation of the consistent valuations for the sentences $\{A, A \supset B, B\}$.

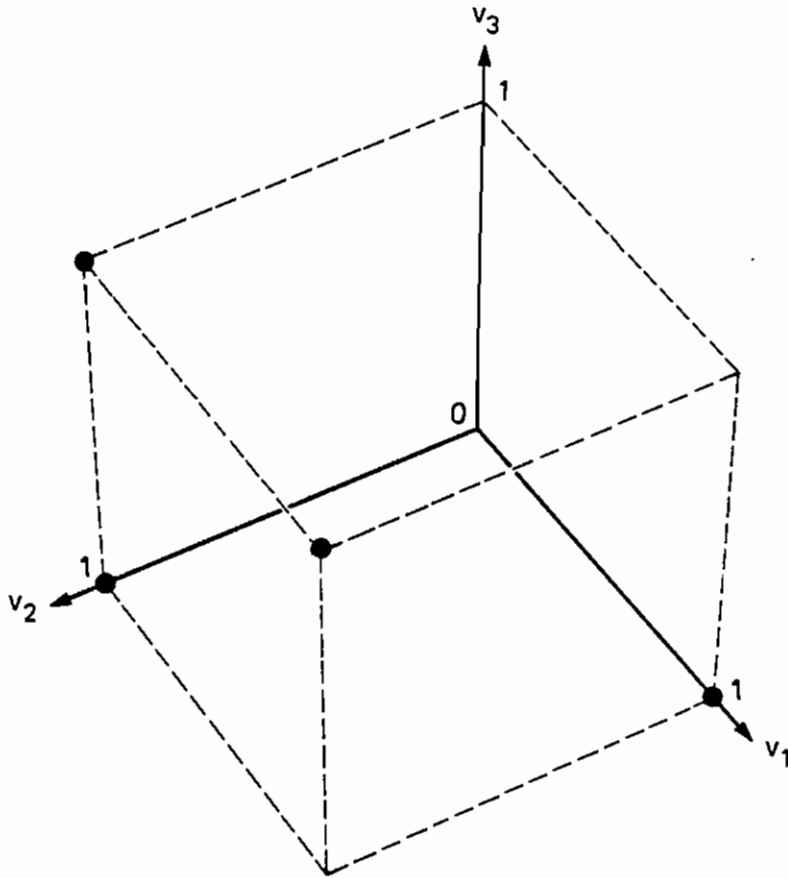


Figure 2: Consistent Valuation Vectors for the Sequence $\{A, A \supset B, B\}$

III ASSIGNING PROBABILITIES TO SENTENCES

Our method of assigning probabilities to sentences involves generalizing the notions of an interpretation vector and of the truth values of sentences. In the nonprobabilistic case, interpretation classes (or classes of possible worlds) were mutually exclusive; each interpretation vector had a single component equal to 1, with all the others equal to zero. In the probabilistic case, we allow a "smearing" over interpretation classes. Each component of a *permissible probabilistic* interpretation vector can be a number between 0 and 1 such that the sum of the components is equal to 1. That is,

$$\mathbf{P} = \begin{bmatrix} p_1 \\ p_2 \\ \cdot \\ \cdot \\ p_K \end{bmatrix}$$

with $0 \leq p_i \leq 1$ and $\sum p_i = 1$.

\mathbf{P} can be thought of as a probability distribution over classes of possible worlds. Viewed intuitively, the i -th component of \mathbf{P} , namely p_i , is the probability that the *real* world—that is, *our* world—is a member of the i -th class of worlds. We note that the permissible probabilistic interpretation vectors fall in the convex hull of the set of points defined by the unit vectors $\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_K$. Each \mathbf{P}_i is on the boundary of the convex hull.

Given a probabilistic interpretation vector \mathbf{P} we can compute a consistent *probabilistic valuation* vector \mathbf{V} over the sentences in \mathcal{S} :

$$\mathbf{V} = \begin{bmatrix} v_1 \\ v_2 \\ \cdot \\ \cdot \\ \cdot \\ v_L \end{bmatrix} = \mathbf{M} \mathbf{P}$$

where the v_i are generalized truth-values or *probabilities* of the S_i (and of the S_i) induced by the probabilistic interpretation. We sometimes denote the probability of a sentence S by $p(S)$ and that of a vector representation of a sentence \mathbf{S} by $p(\mathbf{S})$. We might say that $p(S)$ is the probability that *our* world is a member of one of those classes of possible worlds in which sentence S is *true*.

Thus, in order to compute a probability for a sentence S , we must first have defined a probability distribution over interpretation classes. This having been done, the probability of S is simply the sum of the probabilities of all the interpretations that satisfy S . (Or, put another way, it is the sum of the probabilities of all possible worlds in which S is *true*.)

A probabilistic valuation vector \mathbf{V} is *consistent* if and only if it can be obtained from a permissible interpretation vector, \mathbf{P} , by the equation $\mathbf{V} = \mathbf{M} \mathbf{P}$. Since the permissible interpretation vectors fall in a convex region and \mathbf{M} is a linear transformation, the consistent valuation vectors also fall within a convex region. We show in Figure 3 the region of consistent valuation vectors for the sentences used in our earlier example. Note that the consistent region in \mathbf{V} -space is bounded by planes that are transformations (under \mathbf{M}) of the planes in \mathbf{P} -space given by $p_i = 0$.

We can use this example to illustrate some important points about probabilistic logic. First, just as it is possible to assign an inconsistent *true-false* valuation vector to a set of sentences, it is also possible to assign them an inconsistent probabilistic one. In fact, for the sentences $\{A, A \supset B, B\}$, any valuation vector outside the convex region shown in Figure 3 is inconsistent.

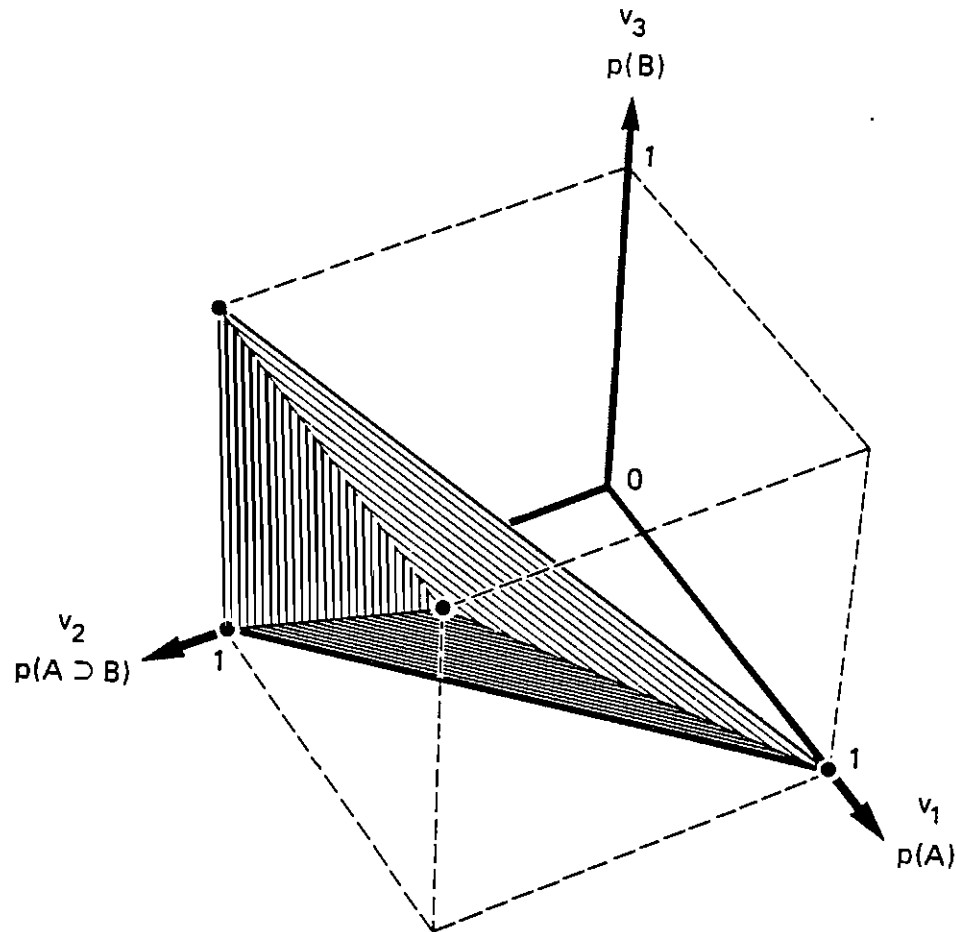


Figure 3: The Region of Consistent Probabilistic Valuation Vectors for the Sequence $\{A, A \supset B, B\}$

(Assignment of consistent subjective probability values to sentences is a well-known problem in expert systems.) Second, even if consistent valuations are assigned for the sentences A and $A \supset B$, the probability of B is not, in general, determined uniquely. Rather, it is bounded by the expressions

$$p(A \supset B) + p(A) - 1 \leq p(B) \leq p(A \supset B)$$

Thus, we can expect that the probabilistic analogue of logical entailment will, as a rule, merely bound (rather than precisely specify) the probabilistic truth-value of the entailed sentence.

Incidentally, in expert systems like PROSPECTOR [5], inference is usually based on a form of Bayes' rule rather than on *modus ponens*. Instead of using the probability of the sentence $A \supset B$ to link sentences A and B , PROSPECTOR uses conditional probabilities, such as $p(A|B)$. In our terms, specification of this conditional probability, for example, could be achieved by using the two sentences $A \wedge B$ and B and giving the ratio of their probabilities.

Our *modus ponens*-type example illustrates only a simple case of probabilistic logical entailment. In general, given a set of sentences \mathcal{S} with consistent probability values, we then want to compute permissible probabilities of other sentences. It is to this problem that we now turn.

IV PROBABILISTIC ENTAILMENT

We could state the problem of probabilistic logical entailment in the following way. We are given the probabilities of some of the sentences in a sequence \mathcal{S} , and we are to compute bounds on the others. We could proceed by setting up the matrix equation $\mathbf{M}\mathbf{P} = \mathbf{V}$ and by using the constraints on \mathbf{P} and the known v_i to compute bounds on the unknown v_j . Our notation is simplified somewhat if, without loss of generality, we state the problem as follows. Given a sequence \mathcal{S} of sentences and a consistent probabilistic valuation vector \mathbf{V} over these sentences, what are the bounds on the probability $p(S)$ of another sentence, S , entailed by \mathcal{S} and \mathbf{V} ? We can, also without loss of generality, assume \mathcal{S} to be satisfiable; that is, we assume that the vector whose components are 1's is in \mathcal{V} . (If it were not, we could merely negate appropriate sentences in \mathcal{S} and complement the corresponding probabilities in \mathbf{V} .) It is convenient also to assume that \mathcal{S} includes as its first element the sentence T , which has value *true* under all interpretations. Then, of course, \mathbf{V} will have 1 as its first element.

Our procedure for computing the probability of S first constructs a sentence matrix \mathbf{M}' as follows:

- (1) Construct \mathcal{S}' by appending S to the end of \mathcal{S} .
- (2) Construct the set \mathcal{V}' of all of the consistent *true-false* valuation vectors for \mathcal{S}' .
- (3) Construct a sentence matrix \mathbf{M} from \mathcal{V}' .
- (4) Construct a matrix \mathbf{M}' from \mathbf{M} by deleting the last row of the latter (which last row is the vector representation \mathbf{S} for S).

The matrix \mathbf{M}' is a representation for the sentences in \mathcal{S} based on interpretation classes that are also a sufficient basis for representing S . The top

row of \mathbf{M}' will be a vector of all 1's which can be thought of as a representation of the sentence T .

The *probabilistic entailment* of S by \mathbf{S} and \mathbf{V} can now be computed as follows:

- (1) Find the permissible solutions \mathbf{P} to the matrix equation $\mathbf{M}' \mathbf{P} = \mathbf{V}$. (Note, that these equations include the constraint $\sum p_i = 1$.)
- (2) Use these solutions to calculate the probability $p(S) = \mathbf{S} \cdot \mathbf{P}$. (The vector \mathbf{S} was the deleted last row in Step 4 of constructing \mathbf{M}' .)

There are two problems with this computational procedure. First, and of lesser importance, the system $\mathbf{M}' \mathbf{P} = \mathbf{V}$ is usually quite underdetermined because $L \ll K$. This difficulty merely means that, as expected, we cannot usually obtain a unique value for $p(S)$. In principle, however, we can obtain bounds. The more serious difficulty is that the matrix \mathbf{M}' is usually too large even to permit practical computation of bounds. We shall deal with these matters in the next two sections after first showing how probabilistic entailment works for a simple problem in first-order logic.

Let \mathcal{S} be the sequence $\{(\exists y)A(y), (\forall x)[A(x) \supset B(x)]\}$, and let S be the sentence $(\exists z)B(z)$. We want to compute the bounds on the probability of $(\exists z)B(z)$. We could proceed exactly as specified in the steps defined above, but, for small-dimensional problems such as this one, it is easier just to graph the convex hull of the set of all consistent *true-false* valuation vectors of the sentences in the sequence $\{(\exists y)A(y), (\forall x)[A(x) \supset B(x)], (\exists z)B(z)\}$.

We establish the consistent *true-false* valuation vectors first by the semantic tree method illustrated in Figure 4. In that figure, we have represented sentences and their negations in Skolem form; a , b , and c are Skolem constants. Paths corresponding to inconsistent valuations are closed off by \times 's. The consistent *true-false* valuation vectors are indicated in columns at the tips of their

corresponding paths. These valuation vectors are graphed in Figure 5, and their convex hull is indicated. This region contains all consistent probabilistic valuations for the sentences in $\{(\exists y)A(y), (\forall x)[A(x) \supset B(x)], (\exists z)B(z)\}$. For consistent values of $p[(\exists y)A(y)]$ and $p\{(\forall x)[A(x) \supset B(x)]\}$, the bounds on $p[(\exists z)B(z)]$ are given by

$$p[(\exists y)A(y)] + p\{(\forall x)[A(x) \supset B(x)]\} - 1 \leq p[(\exists z)B(z)] \leq 1 .$$

These bounds loosen markedly as we move away from $p[(\exists y)A(y)]$ and $p\{(\forall x)[A(x) \supset B(x)]\}$ equal to 1.

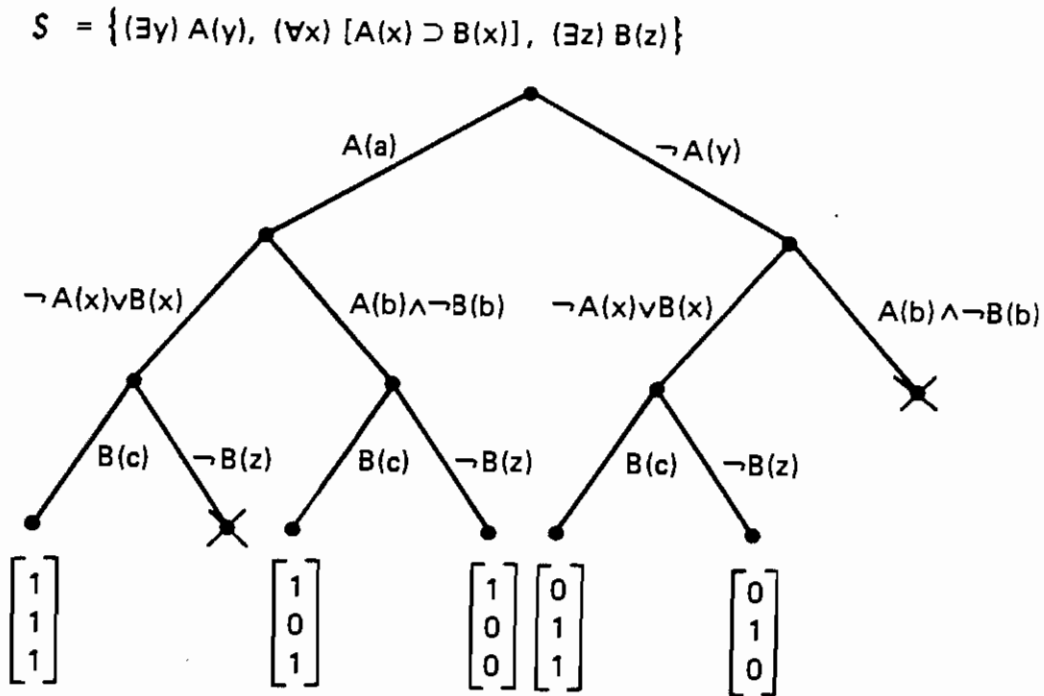


Figure 4: A Semantic Tree for the Sequence $\{(\exists y)A(y), (\forall x)[A(x) \supset B(x)], (\exists z)B(z)\}$

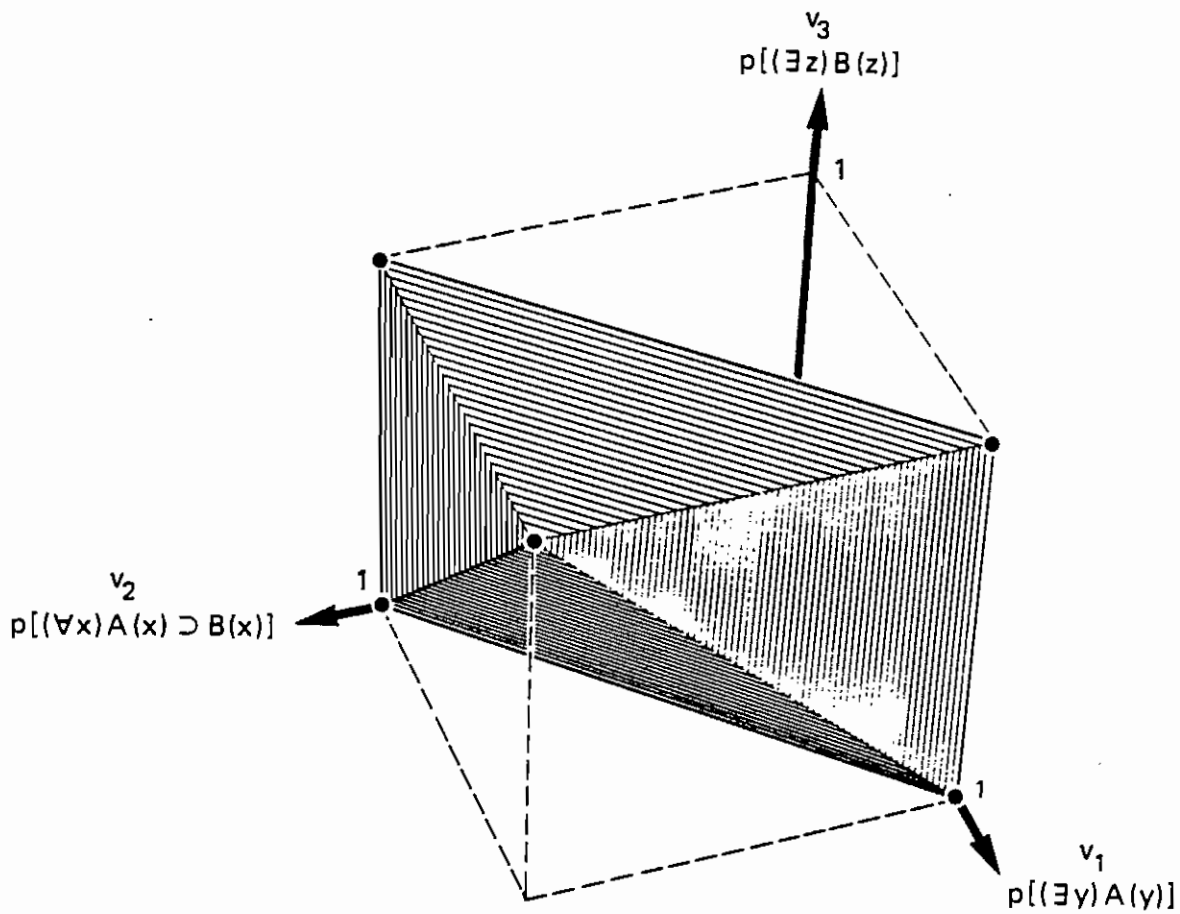


Figure 5: The Region of Consistent Probabilistic Valuation Vectors for the Sequence $\{(\exists y)A(y), (\forall x)[A(x) \supset B(x)], (\exists z)B(z)\}$

V COMPUTATIONS APPROPRIATE FOR SMALL SENTENCE MATRICES

In certain degenerate cases we can compute a unique \mathbf{P} from \mathbf{M}' and \mathbf{V} . For example, if \mathbf{S} happens to be identical to the i -th row of \mathbf{M}' , then $\mathbf{S}\cdot\mathbf{P} = v_i$. More generally, if \mathbf{S} can be written as a linear combination of rows of \mathbf{M}' , then $\mathbf{S}\cdot\mathbf{P}$ can be simply written as the same linear combination of the v_i . For example, the following identities can readily be established by the method of representing the vector whose probability is being computed as a linear combination of rows of an appropriate \mathbf{M}' :

$$p(B) = p(A) + p(A \supset B) - p(B \supset A)$$

$$p(B) = p(A \supset B) + p(\neg A \supset B) - 1$$

We might also imagine that, if \mathbf{S} can be approximated (in some sense) by a linear combination of the rows of \mathbf{M}' , then $\mathbf{S}\cdot\mathbf{P}$ can be approximated by the same linear combination of the v_i . Such approximations may well be useful and worth looking for. An approximation that we might consider is \mathbf{S}^* , the projection of \mathbf{S} onto the subspace defined by the row vectors of \mathbf{M}' . By definition, \mathbf{S}^* will be a linear combination of the vectors in \mathbf{M}' :

$$\mathbf{S}^* = \sum c_i \mathbf{S}_i$$

Our approximation of the probability of \mathbf{S} would then be given by

$$p(\mathbf{S}^*) = \mathbf{S}^* \cdot \mathbf{P} = \sum c_i \mathbf{S}_i \cdot \mathbf{P} = \sum c_i v_i$$

Suppose we use this method to approximate the probability of B , given the sentence A , with probability p , and the sentence $A \supset B$, with probability q . \mathbf{M}' and \mathbf{V} are then given by

$$\mathbf{M}' = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix} \quad \mathbf{V} = \begin{bmatrix} 1 \\ p \\ q \end{bmatrix}$$

The vector representation for B is $\mathbf{B} = (1 \ 0 \ 1 \ 0)$, and its projection onto the subspace defined by the row vectors of \mathbf{M}' is $\mathbf{B}^* = (1, 0, 1/2, 1/2)$. The coefficients c_i are given by $c_1 = -1/2$, $c_2 = 1/2$, and $c_3 = 1$. Using these, $p(\mathbf{B}^*) = \mathbf{S}^* \cdot \mathbf{P} = p/2 + q - 1/2$. Note that this value happens to be midway between the two previously established bounds on $p(B)$.

Another technique that can be used when we are given underdetermined (but consistent) \mathbf{M}' and \mathbf{V} is to select from among the possible solutions for \mathbf{P} the one that has maximum entropy. This distribution assumes the minimum additional information about \mathbf{P} , given the L sentences in S and their probabilities.

Using a technique proposed by Cheeseman [3], we write the entropy H of a distribution \mathbf{P} as follows:

$$H = -\mathbf{P} \cdot \log \mathbf{P} + l_1(v_1 - \mathbf{S}_1 \cdot \mathbf{P}) + l_2(v_2 - \mathbf{S}_2 \cdot \mathbf{P}) \\ + \dots + l_L(v_L - \mathbf{S}_L \cdot \mathbf{P}) \quad ,$$

where l_1, \dots, l_L are Lagrange multipliers; v_1, v_2, \dots, v_L are the components of \mathbf{V} , and $\mathbf{S}_1, \dots, \mathbf{S}_L$ are the (row) vectors of \mathbf{M}' .

Differentiating this expression with respect to p_i and setting the result to zero yields

$$-\log p_i - 1 - l_1 S_{1i} - \dots - l_L S_{Li} = 0 \quad ,$$

where S_{ji} is the i -th component of the j -th row vector in \mathbf{M}' .

Thus, the distribution that maximizes the entropy has components

$$p_i = e^{-1} e^{-(l_1 S_{1i})} \dots e^{-(l_L S_{Li})} .$$

The following definitions can be used to simplify this expression:

$$a_1 = e^{-1} e^{-(l_1)}$$

$$a_j = e^{-(l_j)}, j = 2, \dots, L$$

We then see that each p_i can be written as a product of some of the a_j , where a_j is included in p_i if S_{ji} is 1 and is not included otherwise. We note that a_1 is included in each of the p_i because $S_{1i} = 1$ for all i .

Now we can solve directly for the a_j by substituting these expressions for p_i as components of \mathbf{P} and solving the equations $\mathbf{M}' \mathbf{P} = \mathbf{V}$ for the a_j .

We will gain some insight by considering some examples.

1. First, we analyze the previous example in which we were given the sentences A , with probability p , and $A \supset B$, with probability q , and wished to find the probability of B . \mathbf{M}' and \mathbf{V} are then given by

$$\mathbf{M}' = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix} \quad \mathbf{V} = \begin{bmatrix} 1 \\ p \\ q \end{bmatrix}$$

We can read down the columns of \mathbf{M}' to express each (entropy-maximizing) p_i in terms of products of the a_j :

$$p_1 = a_1 a_2 a_3$$

$$p_2 = a_1 a_2$$

$$p_3 = a_1 a_3$$

$$p_4 = a_1 a_3$$

Using these values in $\mathbf{M}' \mathbf{P} = \mathbf{V}$ yields the equations

$$a_1 a_2 a_3 + a_1 a_2 + 2a_1 a_3 = 1$$

$$a_1 a_2 a_3 + a_1 a_2 = p$$

$$a_1 a_2 a_3 + 2a_1 a_3 = q$$

Solving yields

$$a_1 = (1 - p)(1 - q)/2(p + q - 1)$$

$$a_2 = 2(p + q - 1)/(1 - p)$$

$$a_3 = (p + q - 1)/(1 - q)$$

Thus, the entropy-maximizing \mathbf{P} is given by

$$\mathbf{P} = \begin{bmatrix} p + q - 1 \\ 1 - q \\ (1 - p)/2 \\ (1 - p)/2 \end{bmatrix}$$

Using this probability distribution, we see that the probability of B is given by $(1 \ 0 \ 1 \ 0) \cdot \mathbf{P} = p/2 + q - 1/2$. (This is the same value calculated by the "projection approximation" method! It would be interesting to know the conditions on \mathbf{M}' , \mathbf{V} and \mathbf{S} such that the probability of \mathbf{S}^* obtained from the projection approximation method is the same as the maximum-entropy probability of \mathbf{S} .)

2. We are given the sentences A , with probability p , and B , with probability q , and wish to find the probability of $A \wedge B$. \mathbf{M}' and \mathbf{V} are then given by

$$\mathbf{M}' = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \quad \mathbf{V} = \begin{bmatrix} 1 \\ p \\ q \end{bmatrix}$$

In this case, the p_i are given by

$$p_1 = a_1 a_2 a_3$$

$$p_2 = a_1 a_2$$

$$p_3 = a_1 a_3$$

$$p_4 = a_1$$

Solving for the a_i in the usual manner yields

$$a_1 = (1 - p)(1 - q)$$

$$a_2 = p/(1 - p)$$

$$a_3 = q/(1 - q) .$$

Thus,

$$\mathbf{P} = \begin{bmatrix} pq \\ p(1 - q) \\ q(1 - p) \\ (1 - p)(1 - q) \end{bmatrix}$$

which is the maximum-entropy distribution. It is also the distribution in the case in which A and B are independent binary random variables. Using this probability distribution, the probability of $A \wedge B$ is $(1, 0, 0, 0) \cdot \mathbf{P} = pq$.

Incidentally, this example provides a case in which the projection approximation method clearly does not give the maximum-entropy probability. The maximum-entropy probability of $A \wedge B$ cannot be given as a *linear* combination of the v_i , since, in fact, it must be the *product* of v_2 and v_3 . As a matter of interest, in this example the projection approximation method estimates the probability of $A \wedge B$ to be $p/2 + q/2 - 1/4$ —not a very good estimate when p and q are either both 0 or both 1. (In computing this estimate, we first compute $\mathbf{S}^* = (3/4, 1/4, 1/4, -1/4)$, using $c_1 = -1/4, c_2 = 1/2, c_3 = 1/2$.)

VI DEALING WITH LARGE SENTENCE MATRICES

The techniques described in the last section all involved computing permissible interpretation vectors \mathbf{P} from \mathbf{M}' and \mathbf{V} . However, from a permissible \mathbf{P} we can derive not only a consistent probability for the originally specified sentence S , but also consistent probabilities for any other sentence that can be represented by using the same interpretation classes. One might hope that there might be simpler methods for computing consistent probabilities for just S and that these simpler methods might be computationally feasible even when computing a permissible \mathbf{P} is not. Or perhaps there are much simpler techniques for computing approximate probabilities when these would suffice.

One simplification is based on subdividing S into smaller sequences. Suppose, for example, that S could be partitioned into two parts, namely, S_1 and S_2 , with no atom that occurs in S_1 occurring in S_2 or in S . Clearly S_1 could be eliminated from S without any effect on probabilistic entailment calculations for S . In this case, we say that the subsequence S_2 is a *sufficient* subsequence for S .

As a more complex example, suppose that two sentences, S_1 and S_2 , could be found such that a subsequence of S , say S_1 , was sufficient for S_1 and another subsequence, say S_2 , was sufficient for S_2 . Then we could split the probabilistic entailment of S from S into two smaller parts: first compute the probabilistic entailments of S_1 from S_1 and of S_2 from S_2 . Next compute the probabilistic entailment of S from $\{S_1, S_2\}$. The idea here is to find sentences S_1 and S_2 , such that together they "give as much information" about S as does S . In this case, S_1 and S_2 are similar to what have been called *local event groups*. This method, of course, is only approximate; its accuracy depends on how well the values of S_1 and S_2 determine the value of S .

We next suggest a process for finding an "approximate" sentence matrix, given S , \mathbf{V} , and S . This approximate sentence matrix, which we denote by \mathbf{M}^* , can be made small enough (and in this sense approximate) to permit practical

computation of approximate probabilistic entailment. The approximation is exact in the nonprobabilistic case when \mathbf{V} consists of only 1's and 0's. It can be made increasingly more precise by making \mathbf{M}^* larger.

We follow the process described in Section 4 for computing \mathbf{M}' except that we modify Step 2 so as not to include *all* of the consistent *true-false* valuation vectors. Instead we construct a smaller set \mathcal{V}^* that includes only vectors "close to" the given probabilistic \mathbf{V} .

\mathcal{V}^* is constructed as follows:

- (1) Construct a *true-false* valuation vector \mathbf{V}_b from \mathbf{V} by changing to 1 the values of those components v_i whose values are greater than or equal to $1/2$. Change the values of the other components to 0.
- (2) If S can have value *true* consistent with the valuations for the sentences in S given by \mathbf{V}_b , then include in \mathcal{V}^* the vector formed from \mathbf{V}_b by appending to it a final component equal to 1. If S can have value *false* consistent with the valuations for the sentences in S given by \mathbf{V}_b , then include in \mathcal{V}^* the vector formed from \mathbf{V}_b by appending to it a final component equal to 0.
- (3) Reverse the values of the components of \mathbf{V}_b , one at a time, two at a time, and so on, starting with those components whose corresponding components in \mathbf{V} have values *closest* to $1/2$. For each of the altered *true-false* valuation vectors thus obtained that represent consistent *true-false* valuations over S , add new vector(s) to \mathcal{V}^* according to the procedure described in Step 2 immediately above. We use as many of these consistent, altered vectors as computational resources permit. The more vectors used, the better the approximation.

We next construct the approximate sentence matrix \mathbf{M}^* by first constructing an intermediate matrix whose columns are the elements of \mathcal{V}^* and

then deleting the last row of this intermediate matrix. (This last row is the approximate vector representation of S .)

It should be clear that, as we include more and more vectors in \mathcal{V}^* , it approaches \mathcal{V}' , and \mathbf{M}^* approaches \mathbf{M}' . Also, if \mathbf{V} is the vector with components all equal to 1, then $\mathbf{V} = \mathbf{V}_b$. In that case, if S logically follows from \mathcal{S} , \mathbf{M}^* need have only a single column (of 1's); $\mathbf{P} = [1]$; $\mathbf{S} = [1]$; and $p(S) = 1$. If $\neg S$ logically follows from \mathcal{S} , \mathbf{M}^* still need have only a single column (of all 1's); $\mathbf{P} = [1]$; $\mathbf{S} = [0]$; and $p(S) = 0$. If both S and $\neg S$ are consistent with \mathcal{S} , then \mathbf{M}^* would have two identical columns (of all 1's); \mathbf{P} could have permissible solutions $[1,0]$ and $[0,1]$; $\mathbf{S} = [1, 0]$; and $p(S)$ could range consistently between 0 and 1.

Thus, our approximation behaves well at the limits of large \mathcal{V}^* and at the nonprobabilistic extreme of valuation vectors. Simple continuity arguments suggest that performance ought to degrade only gradually as we depart from these extremes, although we have not yet tested the method on large examples. If we recall that the region of consistent probabilistic valuation vectors occupies the convex hull of the set of consistent *true-false* valuation vectors, we note that our approximation method constructs an approximate region—namely, the convex hull of just those *true-false* valuation vectors that are “close to” the given probabilistic valuation vector \mathbf{V} . We suspect that the more uncertain the sentences in \mathcal{S} are, the more vectors will have to be included in \mathcal{V}^* to achieve accurate entailment.

VII DISCUSSION AND CONCLUSIONS

We have presented a straightforward generalization of the ordinary (true-false) semantics for logical sentences to one that allows probabilistic values on sentences. Although implementation of the full procedure for probabilistic entailment would usually be computationally impractical, we also described a simple approximation method that may be appropriate for realistic problems.

In applying these ideas to expert systems, an expert would supply [prior] probabilities on a sequence of sentences at the time the system is designed. (That is, for each sentence in the sequence, the expert would estimate the probability that any actual world encountered in practice would be one for which that sentence is *true*.) The user of the system would be asked for (or would volunteer) information about what he knows to be true in the specific case he is considering. If, for example, a user says that a particular sentence S_0 is *true* in the actual world he is considering, then each of those components of the interpretation vector that correspond to interpretations satisfying S_0 would have to be normalized by dividing by the prior, expert-supplied probability for S_0 . Those components that correspond to interpretations falsifying S_0 would be set to zero. This operation amounts to an application of Bayes' rule to yield the *conditional* probabilities of each of the possible worlds, *given* that the actual world must be one in which S_0 is *true* [6]. Once the normalized (or we might say *posterior*) probabilistic interpretation vector \mathbf{P} is computed, we can proceed with the computation of bounds for the probability of whatever sentence S the expert system is attempting to render a judgment about.

Expert-system applications will also require a technique for dealing with inconsistent probability values supplied by the expert and user. One possibility would be to move an inconsistent valuation vector to a "nearby" point in the consistent region, perhaps with a preference for larger adjustments to "user probabilities" than to "expert probabilities." Our technique can also be utilized in

an obvious way when the probabilities of sentences are merely bounded rather than having definite values. In this regard, it would be interesting to explore the relationships between the techniques presented here and Shafer-Dempster theory.

Some have proposed that probabilistic logics of various kinds might be non-monotonic and would therefore support *default* reasoning. The probabilistic logic presented here is actually *monotonic* in the sense that additional constraints on the probability values of sentences *reduce* the region of consistent valuations. Since adding such constraints never has the effect of *adding* to the region of consistent valuations, no new valuations can result.

VIII ACKNOWLEDGEMENTS

My first version of this paper dealt only with probabilistic *propositional* calculus. The set \mathcal{V} contained *true-false* consistent valuation vectors of sentences under the interpretations defined over the *true-false* valuations of all the atoms contained in the sentences. Stan Rosenschein pointed out that the method could easily be generalized by considering only *equivalence classes* of interpretations (equivalent with respect to their consistent valuation vectors over the sentences). I have also profited from discussions with Michael Georgeff, Peter Cheeseman, Kurt Konolige, Benjamin Grosf and John Lowrance.

References

- [1] Adams, E. W. and H. F. Levine.
"On the Uncertainties Transmitted from Premises to Conclusions in Deductive Inferences." *Synthese* 90 :429-460, 1975.

- [2] Carnap, R.
"The Two Concepts of Probability." *Logical Foundations of Probability*, :19-51, University of Chicago Press, Chicago, IL, 1950.

- [3] Cheeseman, P.
"A Method of Computing Generalized Bayesian Probability Values for Expert Systems." *Proceedings of the International Joint Conference on Artificial Intelligence*, William Kaufmann, Inc., Los Altos, CA, August, 1983.

- [4] Dempster, A. P.
"A Generalization of Bayesian Inference." *Journal of the Royal Statistical Society, Series B*, vol. 30, :205-247, 1968.

- [5] Duda, R. O., P. E. Hart, and N. J. Nilsson.
"Subjective Bayesian Methods for Rule-Base Inference Systems." *Proceedings 1976 National Computer Conference. AFIPS*, vol. 45, :1075-1082, 1976. Reprinted in *Readings in Artificial Intelligence*, Webber, B. W., and N. J. Nilsson (editors), Tioga Publishing Company, Palo Alto, CA, 1981.

- [6] Georgeff, M. P.
Private Communication, 1984.

- [7] Halpern, J. Y. and M. Rabin.
"A Logic to Reason about Likelihood." IBM Research Report RJ 4136 (45774), December 19, 1983. Also appears in *ACM Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, Boston, MA, (ACM Order No 508830), :310-319, April 25-27, 1983.
- [8] Hempel, C. G.
"Studies in the Logic of Confirmation." *Aspects of Scientific Explanation and other Essays in the Philosophy of Science*, :3-51, The Free Press, New York, 1965.
- [9] Konolige, K. G.
"An Information-Theoretic Approach to Subjective Bayesian Inference in Rule-Based Systems." SRI Working Paper, March, 1982. (A revision of Appendix D, "Bayesian Methods for Updating Probabilities," in Duda, R.O., et al, "A Computer-Based Consultant for Mineral Exploration," Technical Report, SRI International, Menlo Park, CA, October 1978.)
- [10] Lemmer, J. F., and S. W. Barth.
"Efficient Minimum Information Updating for Bayesian Inferencing in Expert Systems." *Proceedings National Conference on Artificial Intelligence*, :424-427, Pittsburgh, PA, 1982.
- [11] Lemmer, J. F.
"Generalized Bayesian Updating of Incompletely Specified Distributions." Working Paper, Par Technology Corporation, New Hartford, NY, November 30, 1982.
- [12] Lowrance, J. D., and T. D. Garvey.
"Evidential Reasoning: A Developing Concept" *IEEE 1982 Proceedings of the International Conference on Cybernetics and Society*, :6-9, October 1982.

- [13] Lowrance, J. D., and T. D. Garvey.
“Evidential Reasoning: An Implementation for Multisensor
Integration,” SRI AI Center Technical Note 307, SRI International,
Menlo Park, CA 94025. December 1983.
- [14] Lukasiewicz, J.
“Logical Foundations of Probability Theory.” In *Jan Lukasiewicz,
Selected Works*, Berkowski, L. (editor), :16-43, North-Holland
Publishing Co., Amsterdam, 1970.
- [15] Shafer, G. A.
Mathematical Theory of Evidence. Princeton University Press,
Princeton, NJ, 1979.
- [16] Shortliffe, E. H.
Computer-Based Medical Consultations: MYCIN. Elsevier, New York,
1976.
- [17] Suppes, P.
“Probabilistic Inference and the Concept of Total Evidence.”
Hintikka, J., and P. Suppes, (Editors), *Aspect of Inductive Logic*,
:49-65, North-Holland Publishing Co., Amsterdam, 1966.
- [18] Zadeh, L. A.
“Fuzzy Logic and Approximate Reasoning.”
Synthese 30, :407-428, 1975.