# Mathematical Programming Methods for Reasoning under Uncertainty *

J. N. Hooker
GSIA, Carnegie Mellon University, Pittsburgh, PA 15213 USA
Matematisk Institut, Århus Universitet, 8000 Århus, Denmark

October 1991

**Abstract**

We survey three applications of mathematical programming to reasoning under uncertainty: a) an application of linear programming to probabilistic logic; b) an application of nonlinear programming to Bayesian logic, a combination of Bayesian inference with probabilistic logic; and c) an application of integer programming to Dempster-Shafer theory, which is a method of combining evidence from different sources.

## 1 Introduction

In recent years the methods of mathematical programming have been applied to reasoning under uncertainty. We will present the basic ideas behind three of these applications: that of linear programming to probabilistic logic, that of nonlinear programming to Bayesian logic, and that of integer programming to Dempster-Shafer theory. A mathematical programming approach not only provides a practical means of computing inferences, as in probabilistic logic and Dempster-Shafer theory, but it can suggest new types of logic for dealing with uncertainty, as in the case of Bayesian logic.

*Probabilistic logic* replaces the "true" and "false" of propositional logic with probabilities. It was originally conceived by George Boole, who came very close to realizing that one could reason in probabilistic logic by solving what we now call a linear programming problem [3, 4, 18, 20, 30]. Probabilistic logic is an attractive alternative to the "confidence factors" often

*This work is partially supported by AFOSR grant 91-0287.

used in expert systems, not only because it, unlike they, is well grounded in theory, but also because it has some attractive practical features. Nonetheless probabilistic logic does not receive the attention it deserves, and when it does receive attention, it is often in the form of skepticism about the possibility of solving the computational problem it poses. But thanks to recent applications of column generation techniques for linear programming, the computational problem is now well solved for fairly large instances.

*Bayesian logic* extends probabilistic logic by applying its semantics to Bayesian inference, and in particular to Bayesian networks, which represent dependence and independence relations among propositions [32]. Bayesian networks are perhaps best known for their role in influence diagrams [33], which are a recent alternative to decision trees. Bayesian logic combines the advantages of probabilistic logic (flexible input requirements, ability to deal with molecular propositions, mathematical programming model) with those of Bayesian networks (ability to capture causal and conditional independence relations).

*Dempster-Shafer* theory [34] addresses the problem of mathematically combining evidence from sources that may conflict, a problem that probabilistic logic cannot solve. But it can pose an onerous computational problem, and we show here how that problem can be attacked with a particular type of integer programming model known as a set covering model.

More detailed treatments of these topics can be found in [1, 8].

We should remark in passing that mathematical programming methods can also be applied to *inductive reasoning*, which is important for the construction of expert systems and other rule bases. Some of these methods are deterministic [35, 23], but one approach [5, 6] infers rules from noisy data. It treats inductive inference as a statistical regression problem in which the fitted formula is a logical rather than a numerical formula. See [5] for a readable introduction.

## 2 Probabilistic Logic

Probabilistic logic is the result of George Boole's effort to capture uncertainty in logical inference [3, 4]. Its formulas are identical to those of propositional logic, but they are assigned continuous probability values rather than simply truth or falsehood.

In a probabilistic knowledge base, each formula is assigned a probability or an interval within which its probability lies. Some conditional proba-

bilities may be specified (or bounded) as well. The inference problem is to determine the probability with which a given conclusion can be drawn. It turns out that this probably can be any number in an interval of real numbers.

In his study of Boole's work [18, 20], T. Hailperin pointed out that the problem of calculating this interval can be naturally captured in a linear programming model, which Boole himself all but formulated. About a decade later N. Nilsson reinvented probabilistic logic and its linear programming formulation [30], and his paper sparked considerable interest [9, 13, 14, 16, 17, 28, 29, 31, 36, 37]. Hailperin provides a historical survey of probabilistic logic in [19].

The theoretical advantage of probabilistic logic, relative to the confidence factors commonly used in expert systems, is that it provides a principled means of computing the probability of an inference rather than an *ad hoc* formula. The main practical advantage is that it allows one to use only as much probabilistic information as is available. A perennial weakness of probability-based reasoning, such as that in decision trees and influence diagrams, is the necessity of supplying a large number of prior and conditional probabilities before any results can be computed. Probabilistic logic makes no such demands.

If one is unhappy with a *range* of probabilities for the inferred proposition, he can obtain a point value by finding an entropy-maximizing solution. This was in fact suggested in Nilsson's paper [30]; see also [26]. But this approach not only poses a much harder, nonlinear computational problem, but it could mislead by deriving a point probability value when a possibly wide range of probabilities are consistent with the known probabilities.

## 2.1 A Linear Programming Model

Suppose that we have a knowledge base consisting of three formulas,

$$\begin{aligned} & x_1 \\ & x_1 \supset x_2 \\ & x_2 \supset x_3. \end{aligned} \tag{1}$$

A *possible world* is an assignment of truth values, true or false, to every atomic proposition $x_j$. In propositional logic, a *model* is simply a possible world, and $x_3$ can be inferred from (1) because $x_3$ is true in every model in which (1) is true.

Suppose, however, the formulas (1) are not known with certainty but have probabilities 0.9, 0.8 and 0.7, respectively. Suppose also that the conditional probability of $x_3$, given that $x_1$ and $x_2$ are true, is 0.8. That is,

$$Pr(x_3|x_1, x_2) = Pr(x_1, x_2, x_3)/Pr(x_1, x_2) = 0.8. \tag{2}$$

We want to know what probabilities can consistently be assigned $x_3$.

Probabilistic logic solves this problem by letting a model be, not a possible world, but a distribution of probabilities over all possible worlds.

In this example, there are $2^3 = 8$ possible worlds, corresponding to the 8 truth assignments,

$$\begin{aligned}(x_1, x_2, x_3) \quad = \quad & (0,0,0),(0,0,1),(0,1,0),(0,1,1), \\ & (1,0,0),(1,0,1),(1,1,0),(1,1,1).\end{aligned} \tag{3}$$

Therefore, if we let $p_1, \ldots, p_8$ respectively be the probabilities assigned to these 8 worlds, we can write the equations,

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & -0.8 & 0 & 0 & 0 & 0.2 \end{bmatrix} \begin{bmatrix} p_1 \\ \vdots \\ p_8 \end{bmatrix} = \begin{bmatrix} 0.9 \\ 0.8 \\ 0.7 \\ 0 \end{bmatrix} \tag{4}$$

$$\sum_{i=1}^{8} p_i = 1, \quad p_i \geq 0, \quad i = 1, \ldots, 8.$$

The 8 columns of the matrix correspond to the 8 possible worlds (3). Since $x_1$ is true in the last 4 worlds (indicated by the 1's in the first row of the matrix), (4) says that $x_1$'s probability 0.9 is the sum of the probabilities $p_5, \ldots, p_8$ of these 4 worlds. The probabilities 0.8 and 0.7 of $x_1 \supset x_2$ and $x_2 \supset x_3$ are similarly computed in rows 2 and 3. The last row of the matrix equation is simply the result of writing (2) in the form,

$$Pr(x_1, x_2, x_3) - 0.8Pr(x_1, x_2) = 0,$$

which is equivalent to,

$$p_8 - 0.8(p_4 + p_8) = 0.$$

The constraints in the last line of (4) ensure that $(p_1, \ldots, p_8)$ is a probability distribution.

The unknown probability $\pi_0$ of $x_3$ is given by,

$$\pi_0 = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} p = c^T p. \tag{5}$$

It is clear that $\pi_0$ can have any value $c^T p$ for which $p$ solves (4). Since (4) and (5) are linear, the possible values of $\pi_0$ lie in an interval that can be found by minimizing and maximizing $c^T p$ subject to the constraints in (4). This is a linear programming problem. The minimum value of $c^T p$ is 0.5, and the maximum value is 0.7, which means that $\pi_0$ can be any probability in the range from 0.5 to 0.7.

### 2.2 Column Generation Techniques

A serious difficulty with the linear programming model of the probabilistic inference problem is that the number of variables $p_j$ can increase exponentially with the number of atomic propositions. This problem can be alleviated using column generation techniques, which are well known in mathematical programming. Their rationale is that they introduce variables into the problem only as they are needed to improve the solution, so that only a small fraction of the total variable set may eventually be used.

Column generation was suggested for probabilistic logic by Nilsson [30] and by Georgakopolous, Kavvadias and Papadimitriou [14] in their paper on probabilistic satisfiability. Three column generation methods for probabilistic logic have been developed in detail—those proposed by Hooker [21], by Jaumard, Hansen, Aragaö and Brun [22, 7], and by Kavvadias and Papadimitriou [24].

The second and third methods above have been computationally tested, with promising results. Jaumard *et al.*, for instance, solve problems with 70 atomic propositions and 100 clauses in about a minute on a Sun Sparc computer. About 600 columns are generated, out of a possible $2^{70}$.

## 3 Bayesian Logic

One weakness of probabilistic logic is that it fails to take into account what may be our most useful source of probabilistic knowledge—the independence of events. We can understand the world only if we assume that most events are significantly influenced by relatively few other events.

An adequate knowledge base should therefore incorporate independence assumptions when they can be made. Ordinary probabilistic logic cannot

Figure 1: A simple Bayesian network.

do this, since independence assumptions give rise to nonlinear constraints that destroy the linearity of its linear programming model. But we can accommodate independence assumptions in a nonlinear programming model, provided there are not too many of them.

A useful device for capturing independence constraints is a Bayesian network [32]. Bayesian networks encode events as nodes and dependence as arcs, and they can capture complex conditional independence relations (i.e., which events are independent, given the occurrence or nonoccurrence of certain other events) [32]. Lauritzen and Spiegelhalter developed a computational approach to solving Bayesian networks [25].

Andersen and Hooker [1] applied the semantics of probabilistic logic to Bayesian networks to obtain "Bayesian logic," which has both the flexibility of the former and the expressive power of the latter. In particular, two weakness of Bayesian networks vanish when they are combined with probabilistic logic: a) their demand for a large number of prior and conditional probabilities as input data, b) their inability to accommodate molecular (as opposed to atomic) propositions without making possibly unwarranted independence assumptions.

### 3.1 Bayesian Networks

A Bayesian network is a directed network in which each node represents an event and each arc a probabilistic dependence. For our purposes, an event is always the truth or falsehood of a proposition, so that we can identify nodes with propositions.

An example is depicted in Fig. 1, in which a symptom (node 1) can be evidence for either disease 2 or disease 3 (nodes 2 and 3). The occurrence of the diseases can in turn be influenced by the presence of a certain genetic trait (node 4), to which one is predisposed by the possession of either of two particular genes (nodes 5 and 6).

In classical Bayesian networks, each node $j$ is associated with atomic proposition $x_j$. The probability that $x_j$ is true is $Pr(x_j)$, and the probability that $x_j$ is false is the probability $Pr(\neg x_j) = 1 - Pr(x_j)$ that its denial $\neg x_j$ is true. It will be convenient to let $X_j$ be a variable whose value is either $x_j$ or $\neg x_j$. The *conditional probability* of $X_i$ given $X_j$ is $Pr(X_i|X_j) = Pr(X_iX_j)/Pr(X_j)$, where $Pr(X_iX_j)$ is an abbreviation for

$Pr(X_i \wedge X_j)$. Two propositions $X_i$ and $X_j$ are *independent* if $Pr(X_i, X_j) = Pr(X_i)Pr(X_j)$. *Conditional independence* is defined as follows: $X_i$ and $X_j$ are independent, given that a set $S$ of propositions are true, if $Pr(X_iX_j|S) = Pr(X_i|S)Pr(X_j|S)$.

The essence of a Bayesian network is that the probability that a node is true, when conditioned on the truth values of all the other nodes, is equal to the probability it is true, conditioned only on the truth values of its immediate predecessors. In Fig. 1 the probability of observing the symptom depends (directly) only on which diseases the patient has, which is to say $Pr(x_1|X_2X_3X_4X_5X_6) = Pr(x_1|X_2X_3)$ for all values of $X_2, \ldots, X_6$. To put it differently, any influence on the probability of the symptom other than the diseases is mediated by the diseases.

Two nodes $i, j$ in a Bayesian network are (conditionally) independent if $X_i, X_j$ are (conditionally) independent for all values of $X_i, X_j$. Clearly, two nodes are independent if they have no common ancestor. Two nodes are conditionally independent, given any fixed set of truth values for a set $S$ of nodes, if they have no common ancestor when the nodes in $S$ are removed from the network.

### 3.2 Possible World Semantics for Bayesian Networks

It is straightforward to interpret a Bayesian network with the semantics of probabilistic logic. When we specify prior or conditional probabilities, we use the same sort of constraints as in ordinary probabilistic logic. For instance, suppose in the example of Fig. 1 that the conditional probabilities of observing the symptoms are,

$$\begin{aligned} Pr(x_1|x_2x_3) = 0.95 \quad Pr(x_1|\neg x_2x_3) = 0.8 \\ Pr(x_1|x_2\neg x_3) = 0.7 \quad Pr(\neg x_2\neg x_3) = 0.1. \end{aligned} \tag{6}$$

Suppose also that the predisposition to either disease is captured in the following conditional probabilities.

$$\begin{aligned} Pr(x_2|x_4) = 0.4 \quad Pr(x_2|\neg x_4) = 0.05 \\ Pr(x_3|x_4) = 0.2 \quad Pr(x_3|\neg x_4) = 0.1. \end{aligned} \tag{7}$$

(6) and (7) can be written as linear constraints, as before. We can of course specify bounds, rather than exact values, for any of these probabilities.

We can also associate nodes with molecular propositions. For instance, let us fix the conditional probabilities $Pr(x_4|X_5X_6)$ of having the genetic

trait so that the patient has it precisely when he has one or both of the genes. Thus,

$$\begin{aligned} Pr(x_4|x_5x_6) = 1 \quad Pr(x_4|\neg x_5x_6) = 1 \\ Pr(x_4|x_5\neg x_6) = 1 \quad Pr(x_4|\neg x_5\neg x_6) = 0. \end{aligned} \tag{8}$$

This in effect associates the molecular proposition $x_5 \vee x_6$ with node 4, even within the framework of a conventional Bayesian network. But the network of Fig. 1 is inappropriate, because it shows nodes 5 and 6 as independent, and we may have no reason to suppose they are independent. We might remove the independence assumption by drawing an arrow from, say, node 5 to node 6. But then we could not solve the network unless we knew the conditional probability $Pr(x_6|x_5)$. In a conventional Bayesian network, then, there is no way to associate a molecular proposition with a node without making possibly unwarranted independence assumptions or presupposing conditional probabilities that may not be available.

Probabilistic logic, however, provides an easy solution to this dilemma. We simply omit nodes 5 and 6 from the Bayesian network, but retain the conditional probability statements (8). This in effect associates $x_5 \vee x_6$ with node 4 without making additional assumptions.

Finally, let us suppose that we can estimate prior probabilities for the occurrence of the two genes:

$$Pr(x_5) = 0.25 \quad Pr(x_6) = 0.15. \tag{9}$$

It remains to encode the conditional independence constraints. Using the definition of conditional probability, we can compute the joint probability distribution of the atomic propositions $x_1, \ldots, x_4$ in Fig. 1 as follows. (We omit $x_5$ and $x_6$ since we dropped the corresponding nodes.)

$$Pr(X_1X_2X_3X_4) = Pr(X_1|X_2X_3X_4)Pr(X_2|X_3X_4)Pr(X_3|X_4)Pr(X_4) \tag{10}$$

The computation is valid for any substitution of $x_j$ or $\neg x_j$ for each $X_j$. Due to the structure of the Bayesian network, two of the conditional probabilities in (10) simplify as follows:

$$Pr(X_1|X_2X_3X_4) = Pr(X_1|X_2X_3) \tag{11}$$

$$Pr(X_2|X_3X_4) = Pr(X_2|X_4). \tag{12}$$

This means that the joint probability in (10) can be computed,

$$Pr(X_1 \ldots X_4) = Pr(X_1|X_2X_3)Pr(X_2|X_4)Pr(X_3|X_4)Pr(X_4). \tag{13}$$

We conclude that the independence constraints (11) and (12) are adequate for calculating the underlying joint distribution and therefore capture the independence properties of the network. Again using the definition of conditional probability, we write (11) and (12) as the following nonlinear constraints:

$$Pr(X_1X_2X_3X_4)Pr(X_2X_3) = Pr(X_1X_2X_3)Pr(X_2X_3X_4) \quad (14)$$

$$Pr(X_2X_3X_4)Pr(X_4) = Pr(X_2X_4)Pr(X_3X_4). \quad (15)$$

Here each variable $X_j$ varies over $x_j$ and $\neg x_j$. To treat (14) and (15) as constraints, we must define each probability $Pr(F)$ in terms of the vector $p$. This is readily done by treating each '$Pr(F)$' as a variable and adding a constraint of the form,

$$Pr(F) = a^T p, \quad (16)$$

where,

$$a_j = \begin{cases} 1 & \text{if } F \text{ is true in world } j, \\ 0 & \text{otherwise.} \end{cases} \quad (17)$$

Now suppose we want to calculate bounds on the probability $Pr(x_2|x_1x_3)$ that a patient with disease 3 and the genetic condition has disease 2. We minimize and maximize $Pr(x_1x_2x_3)/Pr(x_1x_3)$ subject to a) the linear constraints representing (6)-(9), b) the independence constraints (14) and (15), and c) constraints of the form (16) that define all variables $Pr(F)$, including $Pr(x_1x_2x_3)$ and $Pr(x_1x_3)$. In this case we obtain the bounds 0.2179 and 0.2836.

### 3.3 Computational Considerations

The exponential explosion of variables $p_j$ in probabilistic logic remains when nonlinear independence constraints are added. In [1] we show how to apply Benders decomposition to the nonlinear programming problem, so that the linear part can be isolated in a subproblem. This allows application of the same column generation techniques that are being used for ordinary probabilistic logic.

Also if the problem is formulated naïvely, the number of nonlinear constraints required to capture independence relations can grow exponentially with the number of nodes in the network. But we show in [1] that if the constraints are properly formulated, their number grows only linearly with the number of nodes for a large class of networks.

Specifically, we show that the number of nonlinear constraints grows exponentially, not with the size of the entire network, but with the size of the largest "extended ancestral set" in the network. When the size of this set is bounded, the number of constraints grows linearly with the number of nodes.

To have an approximate understanding of this claim, we can imagine that the nodes of a Bayesian network are people, and the immediate predecessors of a node are that person's parents. (We assume a person can have more than two parents.) Beginning with any person in the network, we group his parents so that no parent in one group has a common ancestor with anyone in another. We do the same with his grandparents, and so on with earlier generations. The resulting groups are "ancestral sets." An ancestral set joined by all the parents of its members is an "extended ancestral set." If all the extended ancestral sets are small, the problem is relatively easy to solve.

# 4 Dempster-Shafer Theory

In an effort to develop a more adequate mathematical theory of evidence than Bayesian statistics, G. Shafer [34] extended some statistical ideas of A. P. Dempster [11, 12] to obtain a theory that is closely related to probabilistic logic. A readable exposition of the role of Dempster-Shafer theory in expert systems can be found in [15]. We will present the basic ideas of Dempster-Shafer theory and show how an integer programming model can help solve its inference problems.

## 4.1 Basic Ideas of Dempster-Shafer Theory

Dempster-Shafer theory is best introduced with an example. Sherlock Holmes is investigating a burglary of a shop and determines, by examining the opened safe, that the burglar was left-handed. Holmes also has evidence that the theft was an inside job. One clerk in the store is left-handed, and Holmes must decide with what certainty he can accuse the clerk.

We first identify a *frame of discernment*, which is very much like a set of possible worlds: it is a set $\Theta$ of exhaustive and mutually exclusive possible states of the world. If $L$ means that the thief is left-handed (and $\overline{L}$ that he is not), and if $I$ means that he is an insider (and $\overline{I}$ that he is not), then in this case we have the four possible states $LI$, $L\overline{I}$, $\overline{L}I$, $\overline{LI}$. Every subset of $\Theta$ corresponds to a proposition in the obvious way: $\{LI\}$ to the proposition

Table 1: Dempster's Combination Rule

| | $m_1(\{LI, L\overline{I}\}) = 0.9$ | $m_1(\Theta) = 0.1$ |
|---|---|---|
| $m_2(\Theta) = 0.2$ | $m(\{LI, L\overline{I}\}) = 0.18$ | $m(\Theta) = 0.02$ |
| $m_2(\{LI, \overline{L}I\}) = 0.8$ | $m(\{LI\}) = 0.72$ | $m(\{LI, \overline{L}I\}) = 0.08$ |

that the thief was a left-handed insider, $\{LI, L\overline{I}\}$ to the proposition that he was left-handed, and so on.

We next assign *basic probability numbers* to each subset of $\Theta$, to indicate Holmes' degree of belief in the corresponding proposition. Perhaps the evidence from the safe leads Holmes to assign a probability number of 0.9 to the set $\{LI, L\overline{I}\}$, indicating the "portion of belief" that is "committed to" the hypothesis that the thief is left-handed. Since the basic probabilities numbers must sum to one, Holmes assigns the remaining 0.1 to the entire frame $\Theta = \{LI, L\overline{I}, \overline{L}I, \overline{LI}\}$, indicating that this portion of belief is committed to no particular hypothesis. This defines a basic probability function $m_1$, with $m_1(\{LI, L\overline{I}\}) = 0.9$, $m_1(\Theta) = 0.1$, and $m_1(A) = 0$ for all other subsets $A$ of $\Theta$. Evidence of an inside job is treated similarly to obtain a second basic probability function $m_2$. In this example, either piece of evidence focuses belief on a single subset of $\Theta$ other than $\Theta$ itself, but belief can in general be divided among several subsets.

Our next task is to combine the left-handed evidence with the insider evidence. For this Shafer uses Dempster's combination rule, which produces a new basic probability function $m = m_1 \oplus m_2$. The combined basic probability $m(C)$ of a set $C$ is the sum of $m_1(A)m_2(B)$ over all sets $A$ and $B$ whose intersection is $C$. If $m_2$ is given by $m_2(\{LI, \overline{L}I\}) = 0.8$ and $m_2(\Theta) = 0.2$, the results are displayed in Table 1. For instance, Holmes should commit $(0.9)(0.8) = 0.72$ of his belief to the proposition that the thief is a left-handed insider (i.e., the left-handed clerk). A portion 0.18 of his belief is committed to the more general proposition that the burglar was left-handed, and 0.08 to the general proposition that he was an insider. A small portion 0.02 remains uncommitted.

If some of the intersections are empty, Shafer normalizes the other basic

probabilities so that they sum to one. Thus if we define,

$$\tilde{m}(C) = \sum_{\substack{A,B \subset \Theta \\ A \cap B = C}} m_1(A) m_2(B), \tag{18}$$

then,

$$m(C) = m_1 \oplus m_2(C) = \frac{\tilde{m}(C)}{1 - \tilde{m}(\emptyset)}. \tag{19}$$

If there are three basic probability functions, then $m = (m_1 \oplus m_2) \oplus m_3 = m_1 \oplus (m_2 \oplus m_3)$, and similarly for larger numbers of functions.

Now that the composite basic probability function $m$ has been computed, it remains to determine how much credence should be placed in each subset of $\Theta$. To do this Shafer defines a *belief function* $Bel$, where $Bel(A)$ is the total credence that should be given a subset $A$ of $\Theta$. He takes $Bel(A)$ to be the sum of the basic probability numbers of all subsets of $A$:

$$Bel(A) = \sum_{B \subset A} m(B). \tag{20}$$

Thus the belief allotted to a proposition is the sum of the basic probability numbers of all the propositions that entail it. For instance, the belief Holmes should allocate to the guilt of the clerk is $Bel(\{LI\}) = 0.72$, since the only subset of $\{LI\}$ with a positive basic probability is $\{LI\}$ itself. The belief he should allocate to the proposition that the thief was left-handed is $Bel(\{LI, L\overline{I}\}) = 0.72 + 0.18 = 0.9$. No credence is given to the proposition that the thief is either a left-handed outsider or a right-handed insider, since no subset of $\{L\overline{I}, \overline{L}I\}$ has a positive basic probability.

## 4.2 A Set Covering Model

The difficulty of combining basic probability functions increases exponentially with the number of functions. To see this, take the simplest case in which each basic probability functions $m_i$ assigns a positive value to only one set $S_i$ other than the entire frame $\Theta$ (i.e., $m_i$ is a *simple support function*). To compute $\tilde{m}(C)$ using formula (18), we must enumerate all intersections of $S_i$'s that are equal to $C$. Equivalently, if $T_j = S_j \setminus C$, we must enumerate all intersections of $T_j$'s that are empty. If $S_1, \ldots, S_{k'}$ contain $C$ and $S_{k'+1}, \ldots, S_k$ do not, we must check all $2^{k'}$ subsets of $\{T_1, \ldots, T_{k'}\}$ to find those whose intersection is empty.

There are various ways to make the enumeration more efficient. Barnett [2] describes a method that applies when each simple support function assigns probability to a singleton. We will present a method valid for all simple support functions that is based on a set covering model. Let us say that an intersection of the sets in a subset S of $\{T_1, \ldots, T_{k'}\}$ is *minimally empty* if the intersection of the sets in no proper subset of S is empty. Then we need only enumerate minimally empty intersections when computing $\tilde{m}(C)$ with formula (18). But some care must be taken in doing the computation.

A *set covering problem* has the form,

$$\begin{aligned} & Ax \geq e \\ & x_j \in \{0,1\}, \quad j = 1, \ldots, N \end{aligned} \tag{21}$$

where $A$ is a 0-1 matrix and $e$ is a vector of 1's. The columns of $A$ correspond to sets and the rows to elements the sets collectively contain. We set

$$a_{ij} = \begin{cases} 1 & \text{if set } j \text{ contains element } i, \\ 0 & \text{otherwise.} \end{cases}$$

Thus a vector $x$ solves (21) if the union of the sets $j$ for which $x_j = 1$ contains all the elements. Such an $x$ is called a *cover*. We say $x$ is a *prime cover* if it properly contains no cover; that is, if no cover $y$ satisfies $y \leq x$ with $y_j \neq x_j$ for some $j$.

Let us associate the sets $\overline{T}_1, \ldots, \overline{T}_{k'}$ with the columns of $A$ and the elements in their union with the rows of $A$, where $\overline{T}$ is the complement of $T$. It is clear that if $x$ is a cover, then the intersection of the sets $T_j$ such that $x_j = 1$ is empty. Furthermore, if $x$ is a prime cover, then the intersection is minimally empty. It therefore suffices to generate all the prime covers for (21) and to use them in an appropriate calculation to obtain $\tilde{m}(C)$. We will first show how to generate the prime covers and then how to do the calculation.

We can obtain an initial cover simply by finding a feasible solution $x^1$ of (21), such as $x^1 = (1, 1, \ldots, 1)$. We can reduce $x^1$ to a prime cover $y^1$ by removing sets from the cover, one by one, until no further sets can be removed without producing a noncover. Suppose, then, that we have generated distinct prime covers $y^1, \ldots, y^t$. To obtain a $(t+1)$-st distinct prime cover, we add the following constraints to (21) and use an integer programming algorithm to find a feasible solution $x^{t+1}$ of the resulting system:

$$\sum_{y_j^\tau = 1} x_j \leq e^T y^\tau, \quad \tau = 1, \ldots, t, \tag{22}$$

where $e$ is a vector of ones. Note that each constraint in (22) excludes any cover that contains a cover already enumerated. We next reduce $x^{t+1}$ to a prime cover $y^{t+1}$, which clearly must be distinct from the prime covers already generated. The process continues until there is no feasible solution of (21) with the additional constraints (22).

Now that we know how to generate all the prime covers, we can illustrate the calculation of $\tilde{m}(C)$. Let us suppose that sets $S_1, \ldots, S_4$ contain $C$ and that a remaining set $S_5$ does not. Thus the formula for $\tilde{m}(C)$ is,

$$\sum_{\substack{U_1 \in \{S_1, \Theta\} \\ \vdots \\ U_4 \in \{S_4, \Theta\}}} m_1(U_1)m_2(U_2)m_3(U_3)m_4(U_4)m_5(\Theta). \tag{23}$$

Let us also suppose that there are three prime covers,

$$\begin{array}{l} \{\overline{T}_1, \overline{T}_2\} \\ \{\overline{T}_2, \overline{T}_3\} \\ \{\overline{T}_4\} \end{array} \tag{24}$$

Consider first the terms of (23) that correspond to the prime cover $\{\overline{T}_1, \overline{T}_2\}$ (i.e., the terms containing both $m_1(S_1)$ and $m_2(S_2)$):

$$\sum_{\substack{U_3 \in \{S_3, \Theta\} \\ U_4 \in \{S_4, \Theta\}}} m_1(S_1)m_2(S_2)m_3(U_3)m_4(U_4)m_5(\Theta). \tag{25}$$

Since each $m_i$ is a simple support function, we have $m_i(S_i) + m_i(\Theta) = 1$ for each $i$, and (25) can be simplified to

$$m_1(S_1)m_2(S_2)m_5(\Theta). \tag{26}$$

Consider next the terms of (24) that correspond to the prime cover $\{\overline{T}_2, \overline{T}_3\}$.

$$\sum_{\substack{U_1 \in \{S_1, \Theta\} \\ U_4 \in \{S_4, \Theta\}}} m_1(U_1)m_2(S_2)m_3(S_3)m_4(U_4)m_5(\Theta).$$

Some of these terms, namely those containing $m_1(S_1)$, have already been accounted for in (26). The sum of the remaining terms simplifies to,

$$m_1(\Theta)m_2(S_2)m_3(S_3)m_5(\Theta). \tag{27}$$

By a similar process we remove redundant terms from the summation

$$\sum_{\substack{U_1 \in \{S_1, \Theta\} \\ U_2 \in \{S_2, \Theta\} \\ U_3 \in \{S_3, \Theta\}}} m_1(U_1)m_2(U_2)m_3(U_3)m_4(S_4)m_5(\Theta).$$

that corresponds to the prime cover $\{\overline{T}_4\}$. After removing redundancies, we obtain the sum of three terms:

$$m_1(S_1)m_2(\Theta)m_4(S_4)m_5(\Theta) \tag{28}$$

$$m_1(\Theta)m_2(S_2)m_3(\Theta)m_4(S_4)m_5(\Theta) \tag{29}$$

$$m_1(\Theta)m_2(\Theta)m_4(S_4)m_5(\Theta) \tag{30}$$

$\tilde{m}(C)$ is now equal to the sum of (26)-(30). A precise statement of this algorithm will appear in [8].

## References


[1] Andersen, K. A., and J. N. Hooker, Bayesian logic, to appear in *Decision Support Systems.*

[2] Barnett, J. A., Computational methods for a mathematical theory of evidence, Information Sciences Institute, University of Southern California (not dated).

[3] Boole, G., *An Investigation of the Laws of Thought, on which are Founded the Mathematical Theories of Logic and Probabilities.* Dover Publications (New York, 1951). Original work published 1854.

[4] Boole, G., *Studies in Logic and Probability*, ed. by R. Rhees, Watts and Co (London) and Open Court Publishing Company (La Salle, Illinois, 1952).

[5] Boros, E., P. L. Hammer and J. N. Hooker, Boolean regression, working paper 1991-30, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA 15213 USA, 1991.

[6] Boros, E., P. L. Hammer, and J. N. Hooker, Predicting cause-effect relationships from incomplete discrete observations, working paper 1991-22, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA 15213 USA, 1991.

[7] Brun, T., Structure probabiliste en logique des propositions, Memoire d'Ingénieur, École des Hautes Études Commerciales, Montréal, Canada (1988).

[8] Chandru, V., and J. N. Hooker, *Optimization Methods for Logical Inference*, to be published by Wiley, 1992.

[9] Chen, S. S., Some extensions of probabilistic logic, in *Uncertainty in Artificial Intelligence* **2**, ed. J. F. Lemmer and L. N. Kanal, North-Holland (1988).

[10] Dubois, D., and H. Prade, A tentative comparison of numerical approximate reasoning methodologies, *International Journal Man-Machine Studies* **27** (1987) 149-183.

[11] Dempster, A. P., Upper and lower probabilities induced by a multivalued mapping, *Annals of Mathematical Statistics* bf 38 (1967) 325-339.

[12] Dempster, A. P., A generalization of Bayesian inference, *Journal of the Royal Statistical Society (Series B)* **30** (1968) 205-247.

[13] Dubois, D., and H. Prade, A tentative comparison of numerical approximate reasoning methodologies, *International Journal Man-Machine Studies* **27** (1987) 149-183.

[14] Georgakopolous, G., D. Kavvadias and C. H. Papadimitriou, Probabilistic satisfiability, *Journal of Complexity* **4** (1988) 1-11.

[15] Gorden, J., and E. H. Shortliffe, The Dempster-Shafer theory of evidence, in B. G. Buchanan and E. H. Shortliffe, eds., *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, Addison-Wesley (Reading, MA, 1984).

[16] Grosof, B. N., An inequality paradigm for probabilistic reasoning, in *Uncertainty in Artificial Intelligence* **1**, ed. J. F. Lemmer and L. N. Kanal, North-Holland (1986).

[17] Grosof, B. N., Non-monotonicity in probabilistic knowledge, in *Uncertainty in Artificial Intelligence* **2**, ed. J. F. Lemmer and L. N. Kanal, North-Holland (1986).

[18] Hailperin, T., *Boole's Logic and Probability*, Studies in Logic and the Foundations of Mathematics v. 85, North-Holland (1976).

[19] Hailperin, T., Probability logic, *Notre Dame Journal of Formal Logic* **25** (1984) 198-212.

[20] Hailperin, T., *Boole's Logic and Probability*, Second Edition, Studies in Logic and the Foundations of Mathematics v. 85, North-Holland (1986).

[21] Hooker, J. N., A mathematical programming model for probabilistic logic, working paper 05-88-89, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA 15213 (July 1988).

[22] Jaumard, B., P. Hansen and M. P. Aragaö, Column generation methods for probabilistic logic, mauscript, GERAD, École des Hautes Études Commerciales, 5255 avenue Decelles, Montrél QC Canada H3T 1V6 (December 1989).

[23] Kamath, A. P., N. K. Karmarkar, K. G. Ramakrishnan, M. G. C. Resende, A continuous approach to inductive inference, Mathematical Sciences Research Center, AT& T Bell Laboratories, Murray Hill, NJ 07974 USA.

[24] Kavvadias, D., and C. H. Papadimitriou, A linear programming approach to reasoning about probabilities, to appear in *Annals of Mathematics and Artificial Intelligence*.

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

[26] Lemmer, J. F., and S. W. Barth, Efficient minimum information updating for Bayesian inferencing in expert systems, Proceedings of the National Conference on Artificial Intelligence, Pittsburgh, PA, 1982. Morgan Kaufmann (Los Altos, CA, 1982) 424-427.

[27] Loveland, D. W., *Automated Theorem Proving: A Logical Basis*, North-Holland (1978).

[28] McLeish, M., Probabilistic logic: some comments and possible use for nonmonotonic reasoning, in *Uncertainty in Artificial Intelligence* **2**, ed. J. F. Lemmer and L. N. Kanal, North-Holland (1986).

[29] McLeish, M., Nilsson's probabilistic entailment extended to Dempster-Shafer theory, in *Uncertainty in Artificial Intelligence* **3** (1989) 23-34.

[30] Nilsson, N. J., Probabilistic logic, *Artificial Intelligence* **28** (1986) 71-87.

[31] Paass, G., Probabilistic logic, in *Non-standard Logics for Automated Reasoning*, ed. P. Smets em et al., Academic Press (New York, 1988) 213-251.

[32] Pearl, J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann (San Mateo, California, 1988).

[33] Shachter, R. D., Evaluating influence diagrams, *Operations Research* **34** (1986) 871-82.

[34] Shafer, G., *A Mathematical Theory of Evidence,* Princeton University Press, 1976.

[35] Trantaphyllou, E., A. L. Soyster and S. R. T. Kumara, Generating logical expressions from positive and negative examples via a branch-and-bound approach, manuscript, Industrial and Systems Engineering, Pennsylvania State University, University Park, PA 16802 USA.

[36] Ursic, S., Generalizing fuzzy logic probabilistic inferences, in J. F. Lemmer and L. N. Kanal, eds., *Uncertainty in Artificial Intelligence*, North-Holland (Amsterdam, 1988) 337-364.

[37] Wise, B. P., and M. Henrion, A framework for comparing uncertain inference systems to probability, *Proceedings of Workshop on Uncertainty and Probability in Artificial Intelligence,* AAAI (Los Angeles, 1985). (xx)