Computer Science Department Technical Report
University of California
Los Angeles, CA 90024-1596

NON-BAYESIAN FORMALISMS FOR MANAGING UNCERTAINTY

Judea Pearl

July 1988
CSD-880054

TECHNICAL REPORT
(R-106)
January 1988

NON-BAYESIAN FORMALISMS FOR MANAGING UNCERTAINTY*

Judea Pearl

UCLA Computer Science Department

Cognitive Systems Laboratory

Los Angeles, California, 90024

ABSTRACT

This report presents and compares four non-Bayesian formalisms for managing uncertainty in reasoning systems: (1) The Dempster-Shafer approach (2) Truth-maintenance systems (TMS) (3) Incidence Calculus and (4) Nilsson's Probabilistic Logic. The unifying framework of this comparison treats each of these formalisms as an attempt to deal with partially specified probabilistic information. The first three approaches are shown to have common semantics; they are based on the notion of *provability* as the basic relationship between evidence and conclusion. The fourth maintains the traditional probabilistic relationships but allows the latter to vary over the space of all models consistent with the specifications available.

Also discussed are: comparisons with Bayes formalism, the nature of probability intervals, the use of TMS for interval computations, and applications to rule-based systems and default reasoning.

* This work was supported in part by NSF Grant #DCR-8313875, Naval Research Laboratory Grant #N00014-87-K-2029 and Micro/Aerojet Grant #W870928.

Table of Contents

Overview........ 3
1 THE DEMPSTER-SHAFER THEORY 4
1.1 Basic Concepts..4
1.2 Comparing Bayes and Dempster-Safter Formalisms 9
1.3 Dempster's Rule of Combination 13
1.4 More on the Nature of Probability Intervals 22
1.5 Applications to Rule-Based Systems 24
1.6 Bayesian vs. Dempster-Shafer: A Semantic Clash 30
2 TRUTH MAINTENANCE SYSTEMS 34
2.1 Naming the Assumptions 35
2.2 Uncertainty Management with ATMS 37
2.3 Incidence Calculus 39
3 NILSSON'S PROBABILISITC LOGIC 40

Overview

Pure Bayesian theory requires the specification of a complete probabilistic model before reasoning can commence, namely, determining for each variable X the conditional probabilities that govern the values of X, given their causal factors. When a full specification is not available, Bayes practitioners have devised approximate methods of completing the model, in line with prevailing patterns of human reasoning. For example, the noisy-OR-gate model of causal interactions represents such a model-completion approximation; if we are given the strength of each individual cause but not the combined impact of several causes, we assume that they combine disjunctively, and that all exceptions to the rules are independent [Pearl, 1987b]. A more extreme example of model completion approximation is demonstrated by the Certainty-Factors formalism, used in the Mycin system [Shortliffe, 1977]. Originally proposed as a departure from probability theory, this formalism has been shown to be equivalent to a restricted probability model, one that corresponds to a tree-structured Bayes network (as in [Pearl, 1982]) with inferences restricted to flow from evidence to hypotheses [Heckerman, 1985].

An alternative method of handling partially specified models is provided by the Dempster-Shafer theory [Shafer, 1976], which is the main topic of this report. Rather than completing the model, the D-S theory sidesteps the missing specifications and resigns instead, to less ambitious inference tasks: computing probabilities of provability rather than probabilities of truths. The partially specified model is used only for extracting qualitative relationships of compatibility (or possibility) among the propositions involved, and these qualitative relationships are then used as a logic for assembling proofs, leading from evidence to conclusions. The stronger the evidence the more likely it is for a complete proof to be assembled.

The current popularity of the D-S theory stems both from its readiness to admit partially specified models and its compatibility with the classical, proof-based style of logical inference. As such, the approach matches the syntax of deductive databases and logic programming but may inherit many of the problems associated with monotonic logic, some of which are demonstrated in Sections 1.2, 1.4, 1.5. and 1.6.

Section 2 presents two additional formalisms for dealing with uncertainty: truth-maintenance systems (Sections 2.1 and 2.2) and Incidence Calculus (Section 2.3). Although independently developed from different motivations, these two approaches are discussed as cousins to the Dempster-Shafer theory because, like the latter, they are based on *provability* as the basic relationship connecting evidence with a conclusion. Truth maintenance systems provide a symbolic machinery for identifying the set of assumptions sufficient for assembling the desired proofs and, hence, when we are given probabilities on these assumptions, the systems can be used as symbolic engines for computing the belief functions sought by the D-S theory. Incidence Calculus (Section 1.3) provides a stochastic simulation approach to computing these belief functions -- subjecting a theorem prover to randomly sampled facts and counting the fraction of time that a proof can be assembled.

Still a third way of dealing with partially specified models is to consider the space of all models consistent with the specifications available, and compute bounds, instead of point values, for the probabilities required. Nilsson's probabilistic logic (Section 3) represents such an approach. It differs from the D-S approach in that it uses complete probabilistic models to do the basic inferences, while logical relationships between sentences are used to define the bounds on the probabilities computed.

1 THE DEMPSTER-SHAFER THEORY

1.1 Basic Concepts

We introduce the Dempster-Shafer (D-S) theory of Belief functions using the classical 3-prisoner puzzle. The story involves three prisoners A, B, and C awaiting their verdict, knowing that one of them will be found guilty and the other two released. Prisoner A asks the jailer, who knows the verdict, to pass a letter to some other prisoner who is to be released. Later, prisoner A asks the jailer for the name of the letter recipient and, having learned that the jailer gave the letter to prisoner B, the problem is to assess the chances of A being the guilty one. The problem can be formulated in terms of three mutually exclusive and exhaustive propositions G_A, G_B, and G_C where G_i stands for "prisoner i was found guilty". Coupled with these, we also have the jailer testimony which could have been either 'B' or 'C', so, can be treated as a bi-value variable L (connoting "letter recipient") taking on the values $\{L_B, L_C\}$.

In the Bayesian treatment of the problem we make two tacit assumptions. First, we assume that not having any prior knowledge regarding the verdict translates to equal prior probabilities on the component of G; $\pi(G_A) = \pi(G_B) = \pi(G_C) = 1/3$. Second, we assume that, in case G_A was true, the jailer would choose the letter recipient at random, giving equal chance to B and C. These two assumptions yield the Bayesian network of Figure 1(a), where Figure 1(b) depicts the link matrix $M_{L|G}$ necessary for full characterization of the information source L. The answer obtained from this model is $P(G_A \mid L_B) = 1/3$, meaning that the jailer testimony is totally irrelevant relative A's prospects of release. In general, if the letter is not handed at random, we then assume $P(L_B \mid G_A) = q$, and obtain $P(G_A \mid L_B) = q(1+q)^{-1}$ which varies continuously from 0 (if B is avoided, $q = 0$) to ½ (if C is avoided, $q = 1$).

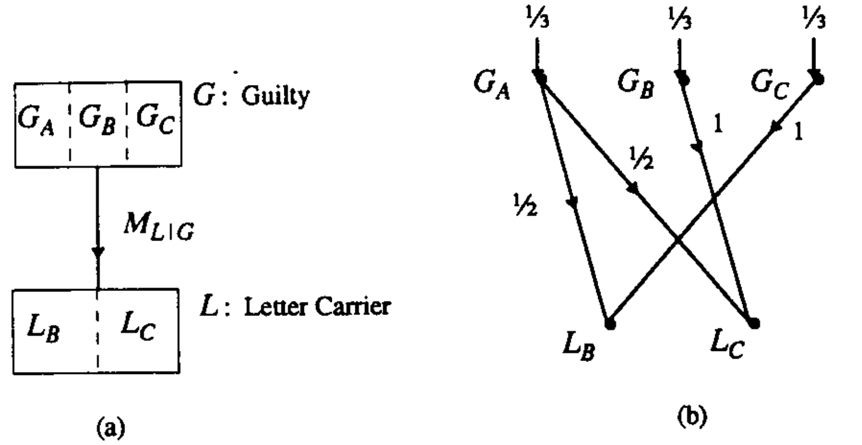


Figure 1. (a) A Bayes network representation of the 3-prisoners puzzle
and (b) the conditional-probability matrix characterizing the link $G \rightarrow L$

Logically speaking, having no idea about the sentencing process only means that either one of the prisoners can be the guilty one, i.e., none can be ruled out conclusively. Similarly, not knowing the process by which the letter recipient was chosen, all we can assert with certitude is that L_B is *compatible* with both G_A and G_C and is incompatible with G_B (assuming the jailer is truthful). Thus, upon obtaining the testimony L_B, the only possible states of affairs are the two combinations: $\{(G_A, L_B), (G_C, L_B)\}$; all the others are ruled out. These legal states are called ''extensions'' in the language of logic, ''solutions'' in the language of constraint processing [Montanari, 1974], ''tuples'' in the language of relational databases, "possibilities" in the language of Fuzzy Logic [Zadeh, 1981] and ''singleton hypotheses'' in the Dempster-Shafer theory. The constraints that determine which extensions are legal are called *compatibility* relations (e.g., that only one prisoner will be found guilty) and represent items of information that one is forced to cast in hard, categorical terms, for lack of a more refined model.

Clearly, not having the parameters $\pi(G_i)$ and $P(L_j \mid G_i)$, prevents us from constructing a complete probabilistic model of the story and prevents us from answering probabilistic queries of the type: ''How certain is G_A in light of the jailer's testimony'', previously encoded as $P(G_A \mid L_B)$. In the partial model available, the probability $P(G_A \mid L_B)$ could be anything between zero and one depending on the prior probability π. On the other hand, if by *certainty* we mean the assurance that G_A can be *proven* true then the certainty of G_A is, logically speaking, zero.

The Dempster-Shafer theory stands between these two extremes, claiming that, even in the logical interpretation of certainty, the assurance of proving a proposition A can take on various degrees, depending on the strength of the evidence available, namely, how close it is to inducing a logical proof of A. This degree of assurance is called ''BELIEF'', and is denoted by $Bel(A)$[1]. In our story, both $Bel(G_A)$ and $Bel(\neg G_A)$ are zero because, having total ignorance re-

1. $Bel(A)$ is to be distinguished from $BEL(A)$, which is defined by: $BEL(A) \triangleq P(A \mid \text{all evidence})$

garding the trial and verdict process means we have no evidence that is capable of enabling a logical proof of either G_A or $\neg G_A$.

Under what conditions would these beliefs become anything but zero? One obvious condition is when the negation of a proposition becomes incompatible with the evidence. For example, since G_B is incompatible with L_B, we have $Bel(\neg G_B) = 1$, stating that $\neg G_B$ is *compelled* by the evidence. But the more interesting condition occurs when *partial* evidence becomes available in favor of some propositions. For example, had the jailer said "Gee, I forgot who I gave the letter to, I bet it was B but I am only 80% sure", we no longer are able to prove $\neg G_B$. Yet, taking the jailer testimony literally, we could say that there is 80% chance that his memory is correct, compelling the truth of $\neg G_B$ and, so, $Bel(\neg G_B) = 0.8$. Similarly, if we have good reason to believe that the testimonies in the trial are equally supportive of each prisoner's innocence and that the verdict process would reflect these testimonies fairly, then, and only then, we would take the liberty of assigning equal *weights* to the components of G.

Let us first focus on the equal weight case, ignoring for the moment the jailer information. The weight distribution process is modeled as a chance event: Imagine a *switch* that oscillates randomly between three positions G_A, G_B, and G_C and, in each of these positions assigns the value *true* to the corresponding proposition and to that proposition only (Fig. 2(a)). If we are asked now about the chance of G_A being provable, the answer would be ⅓, because the switch spends one third of the time in position G_A where G_A is fully confirmed. In this portion of the time the truth of G_A is established externally while G_B and G_C can be proven false by virtue of being incompatible with G_A. Thus, averaging over all three positions of the switch,

$$Bel(G_A) = Bel(G_B) = Bel(G_C) = 1/3$$

and

$$Bel(\neg G_A) = Bel(\neg G_B) = Bel(\neg G_C) = 2/3,$$

exactly as in the Bayesian treatment.

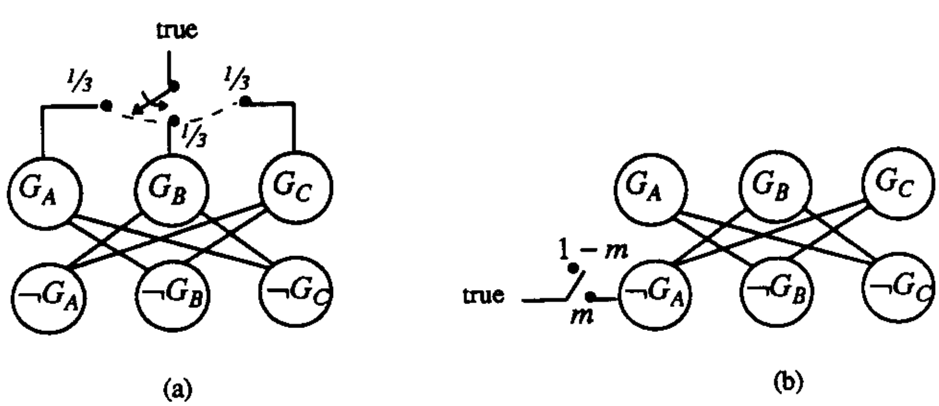


Figure 2. Random switch model representing: (a) equal prior probabilities, and (b) an alibi, weakly supporting A's innocence.

The departure from Bayes formalism surfaces when we venture to devise more fanciful mechanisms for the weight-distributing switch so as to form more faithful models of how, according to D-S advocates, people encode incomplete knowledge. Assume, for example, that the evidence gathered during the trial is not available to us in its entirety, but, rather, we have access to only a small portion of it, containing an alibi weakly supporting prisoner A's innocence. Assume, further, that the alibi bears exclusively on A's whereabout at the time of the crime but bears no direct relationship to B's or C's involvement. The D-S theory will model this case by the switch shown in Fig. 2(b): m percent of the time the switch will force the truth of $\neg G_A$, while the remaining $1 - m$ percent of the time it will stay in a "neutral" position, lending support to no specific hypothesis, or, equivalently, supporting the universal hypothesis $\theta = G_A \vee G_B \vee G_C$.

To calculate the belief functions $Bel(G_A)$ and $Bel(\neg G_A)$ we first identify the positions of the switch in which G_A can be *proven* true, then calculate the percentage of time spend in these positions. In the first position, representing the validity of the alibi, the switch forces the truth of $\neg G_A$, while in the neutral position, it is compatible with both G_A and $\neg G_A$ so none can be proven. Hence, $Bel(\neg G_A) = m$ and $Bel(G_A) = 0$. The belief acquired by the other elementary propositions is zero (prior to the jailer's testimony) because, even in the first position, the switch is compatible with each of the four propositions: G_B, G_C, $\neg G_B$, $\neg G_C$.

The parameter $m(A)$, measuring the strength of argument in favor of a proposition A, is called *basic probability assignment* and the proposition A, upon which the argument bears directly, is called *focal element*. If there is only one focal element A then the weight $1 - m(A)$ is assigned to the universal proposition θ and the belief in any other proposition, B, is given by

$$Bel(B) = \begin{cases} 1 & \text{if} \quad B = \theta \\ m(A) & \text{if} \quad A \supset B \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

A complex piece of evidence may be represented by a switch with more than two positions, each position forcing a different constraint on the knowledge base for a certain fraction of the time. For example, if an evidence was found suggesting that the guilty suspect was left-handed (with weight m_1) and black-haired (with weight m_2), and if suspects A and B are left handed while B and C have black hair, then the constraint $G_A \vee G_B$ will be imposed a fraction m_1 of the time, $G_B \vee G_C$ a fraction m_2 of the time, and the rest of the time, $1 - m_1 - m_2$, no external constraint will be imposed.

In general, if there are several focal elements A, the total weight still sums to unity

$$\sum_A m(A) = 1 \tag{2}$$

and $Bel(B)$ may be affected by all the A's, via

$$Bel(B) = \sum_{A : A \supset B} m(A) \tag{3}$$

The summation reflects the fact that if B can be proven from several positions of the switch then $Bel(B)$, the probability that B is provable, is the total time spent in all those positions.

The measure $1 - Bel(\neg A)$ is called the *plausibility* of A, denoted

$$Pl(A) = 1 - Bel(-A), \tag{4}$$

and represents the probability that A is compatible with the available evidence, i.e., that it cannot be disproven. In our example, $Pl(G_A) = 1 - m$, while G_B and G_C have plausibility 1. The interval

$$Pl(A) - Bel(A) = 1 - [Bel(A) + Bel(\neg A)] \geq 0$$

represents the probability (fraction of time) that both A and $\neg A$ are compatible with the available evidence.

1.2 Comparing Bayes and Dempster-Shafer Formalisms

We see that the D-S theory differs from probability theory in several aspects. First, it accepts an incomplete probabilistic model where some parameters (e.g., the prior or conditional probabilities) are missing. Second, the probabilistic information that is available, like the strength of evidence, is not interpreted as likelihood ratios but rather as random switches that distribute truth values to various propositions for a certain fraction of the time. This model permits a proposition and its negation to be simultaneously compatible (with the switch) for a certain portion of the time, and this may permit the sum of their beliefs to be smaller than unity. Finally, due to the incompleteness of the model, the D-S theory does not pretend to provide full answers to probabilistic queries but, rather, resigns to providing partial answers. It estimates how close the evidence is to forcing the truth of the hypothesis, instead of estimating how close the hypothesis is to being true.

This last point is the most important departure between the two formalisms and is best illustrated, in the three-prisoner's puzzle, by trying to incorporate the jailer's information L_B onto the equal-weight model $m(G_A) = m(G_B) = m(G_C) = 1/3$ (see Fig. 3).

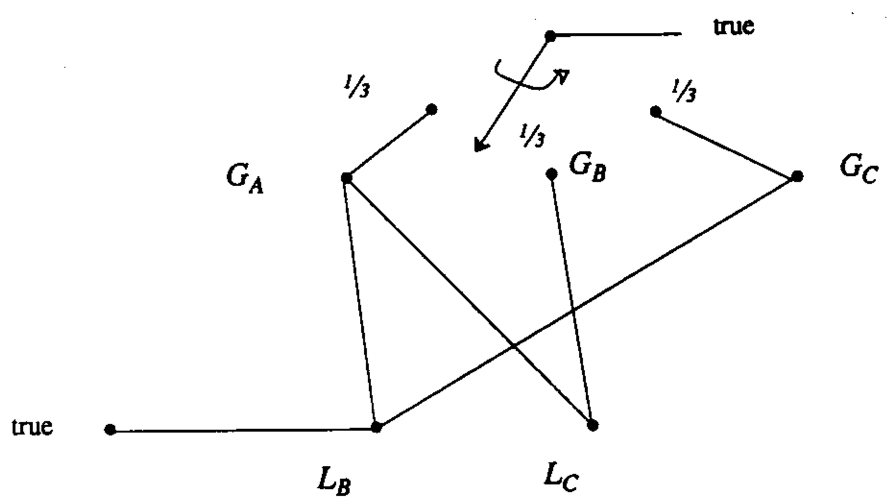


Figure 3. The D-S representation of the 3-prisoners story, incorporating equal prior probabilities and the evidence L_B = true.

Starting with $Bel(G_A) = 1/3$, we now ask for the revised value of $Bel(G_A)$ given L_B, i.e., the proportion of the time that proposition G_A is provable, considering all the evidence available. Clearly, the time spent by the switch in position G_B is incompatible with the evidence L_B, so, we exclude this time from the calculation. The remaining $2/3$ of the time is divided equally between G_A and G_C and, hence, G_A is forced to be true with probability $1/2$, yielding $Bel(G_A) = Bel(\neg G_A) = 1/2$. By comparison, the Bayesian analysis gives $P(G_A \mid L_B) = 1/3$ for the random choice model of Figure 1, while for a partial model with unknown $P(L_B \mid G_A)$, the posterior probability $P(G_A \mid L_B)$ may take on any value between 0 and $1/2$ (See Section 1.1).

This disparity is not surprising in view of the fact that we still have an incomplete probabilistic model on our hands, as the process by which B was selected remains unspecified. Conservatively speaking, it is quite possible that the jailer's choice was not random but marred by a deliberate attempt to avoid choosing C, whenever possible. Under such extreme circumstances, the jailer's answer L_B could only be avoided ⅓ of the time (when B is guilty), thus leaving A and C an equal chance of being the condemned. What may sound somewhat counter-intuitive is that, from among all possible ways of completing the model, D-S theory appears to select the one that puzzle books repeatedly warn us to avoid.

Actually the D-S theory never attempts to complete the model and, if such appears to be the case, it is only an artifact of the procedure. Prior to the jailer's answer, all four extensions $\{(G_A, L_B), (G_A, L_C), (G_B, L_C), (G_C, L_B)\}$ are compatible with the evidence, each of the last two receiving $m = 1/3$ while the remaining weight (1/3) is assigned to the disjunction of the first two, uncommitted to any one in particular. The facility of keeping some weight uncommitted (often associated with the notion of postponing judgement) is a distinct feature of the D-S theory and does not correspond to any probabilistic model. This uncommitted weight gets distributed only upon receiving the jailer's answer which, having excluded the extensions (G_A, L_C) and (G_B, L_C), routes all the uncommitted weight (1/3) to the remaining member of the first pair, (G_A, L_B). At this point the resulting weight distribution happens to be totally distributed to the individual extensions, resembling the weight distribution of complete probabilistic models, but this is where the resemblance ends.

The disparity between the answers produced by the two formalisms stems not from the weight distribution but, rather from the semantics of these answers. While the probabilistic approach interprets "belief in A" to mean the conditional probability that A is *true*, given the evidence e, the D-S approach calculates the probability that the proposition A becomes *provable* given the evidence e. Phrased another way, it computes the probability that some set of hypotheses suggested by the evidence would materialize (e.g., that the judges become convinced by the alibi), from which the truth of A can be derived out of logical necessity. Thus, instead of the conditional probability $P(A \mid e)$, the D-S theory computes the probability of the logical entailment $e \models A$. The entailment $e \models A$ is not a proposition in the ordinary sense, but a meta level relationship between e and A, requiring a logical, object-level theory by which a proof from e to A can be constructed. In the D-S scheme the object level theory consists of categorical *compatibility* relations among the propositions, stating, for example, that L_B is compatible with G_A but incompatible with G_B. Such compatibility relations are the only logical notions that precipitate, once we refrain from committing numbers to certain probabilities. It is remarkable that, while the calculation of $P(A \mid e)$ and even the probability of the material conditional $P(e \Rightarrow A)$ require complete probabilistic models, $P(e \models A)$ does not. For example, in the incomplete model of Figure 3, $P(L_B \models G_A)$ was calculated to ½ without any assumption on the process by which the letter recipient was selected; we simply take one minus the (normalized) weight assigned to all propositions compatible with both L_B and the negation of G_A, namely, one minus the (normalized) time the switch spends at G_C.

At this point, it is natural to ask whether conditional probability information, if available, can be incorporated in the D-S model, and whether it will lead to the same answer as the Bayes model. The answer is positive. Instead of dealing with individual variables, we now create the set of all feasible extensions, and attach to each extension a weight m equal to the appropriate joint probability dictated by the probabilistic model. To illustrate, if in the 3-prisoners example we accept the equal prior and random selection models, then the four feasible extensions $\{(G_A, L_B)(G_A, L_C)(G_B, L_C)(G_C, L_B)\}$ initially receive the weights $\{1/6, 1/6, 1/3, 1/3\}$ (see Figure 1(b)). This assignment can be modeled by a 4-position switch whose contacts represent extensions rather than atomic propositions. When the evidence $e = L_B$ obtains, it rules out two extensions, (G_A, L_C) and (G_B, L_C) and forces the switch to spend $1/3$ of its time at (G_C, L_B) and $1/6$ of the time at (G_A, L_B). Thus,

$$Bel(G_A) = \frac{1/6}{1/3 + 1/6} = 1/3, \quad Bel(\neg G_A) = \frac{1/3}{1/3 + 1/6} = 2/3$$

as in the Bayes analysis.

We see from this example that any complete probabilistic model can be encoded in the D-S formalism, albeit in a somewhat clumsy manner. Probabilities are encoded as weights assigned to individual extensions, instead of conditional probabilities among propositions.

It is also natural to raise the converse question, can probability theory answer the kind of queries that the D-S theory does, namely, how close the evidence comes to rendering a proposition *provable?* In principle, the answer should be positive, because possessing a complete model should give one the power to do everything that can be done with a partial model and perhaps more. Let us examine this point in detail.

Let us assume that we possess the complete probabilistic model specified in Figure 1, including the evidence $e = L_B$, and we now ask for the probability that G_A can be *proven* true. Obviously, after calculating $P(G_A | L_B) = 1/3$, the answer ought to be zero; there is no way to prove a proposition true if that proposition has only $1/3$ probability of being true. Moreover, before probability theory can answer questions of this sort , the process of proving or disproving propositions must be integrated as part of the available probabilistic model, else the query is vacuous.

This was precisely the reason for invoking the random-switch metaphor in the D-S formalism (alternative metaphors, using databases [Zadeh, 1986] or voting [Hummel and Manevitz, 1987] models can serve the same purpose). The function of the switch is to cast the event of proving or disproving a proposition in traditional probabilistic setting. Once the position of the switch is determined and the compatibility relations agreed upon, the existence of a proof becomes an testable event, similar to the output of a physical logic circuit whose model is well understood (we leave out, at this discussion, questions of decidability). Thus, the random switch together with the compatibility constraints, constitute a complete meta-level probabilistic model, that can be interrogated to answer any probabilistic query within its vocabulary. It is, in a way, an autoepistemic model (i.e., concerning one's own knowledge), as it involves hypothetical reasoning about what one ought to believe as true had certain facts (constraints) become known (i.e., forced by the switch).

Such meta-level exercises are not foreign to probabilistic theories. In the Bayesian literature it is a common practice to compute the value of information sources by envisioning, hypothetically, what the impact of a test result would be on the belief in some hypothesis [Spiegelhalter, 1986, Pearl, 1987c]. This was computed by simulating the propagation of a 'would be' evidence through a knowledge base represented as a Bayes network. Similarly, we can measure the confidence level of our beliefs by envisioning how they would vary in the face of hypothetical events called contingencies [Pearl, 1987a]. It stands to reason, therefore, that meta-level reasoning similar to that used in the D-S approach could very well be formulated in purely probabilistic terms, for example, by taking Bayes networks as the object-level knowledge base, instead of non-probabilistic compatibility constraints.

Let us demonstrate this exercise using the complete model of Figure 1. Assume that our objected-level model is given by the Bayes network of Figure 1 (with $\pi(G_A) = \pi(G_B) = \pi(G_C) = 1/3$) and that, instead of giving a categorical answer, the guard replies "Gee, I think I gave the letter to B but I am only 80% sure". Taking this testimony literally, we subject the model of Figure 1 to the influence of a random switch, forcing the truths of L_B and L_C with probability 0.8 and 0.2, respectively. Conceptually, this amounts to submitting the object-level model to a hypothetical meta-reasoner who observes the behavior of the former under two different pieces of evidence, L_B = true and L_C = true.

We know the answer to this exercise; the Bayesian analysis of either $\boldsymbol{e} = L_B$ or $\boldsymbol{e} = L_C$ gives $BEL(G_A) \triangleq P(G_A \mid \boldsymbol{e}) = 1/3$, hence, if someone asks for the value of $Bel(G_A)$, the probability of G_A being provable, the answer is, of course, zero. Provability means $BEL = 1$. Things are less trivial when we wish to calculate $Bel(G_C)$ and $Bel(\neg G_C)$. Under $\boldsymbol{e} = L_B$, the object-level model returns the probabilities $P(G_C \mid \boldsymbol{e}) = 2/3$ and $P(\neg G_C \mid \boldsymbol{e}) = 1/3$, while under $\boldsymbol{e} = L_C$ these probabilities become $P(G_C \mid \boldsymbol{e}) = 0, P(\neg G_C \mid \boldsymbol{e}) = 1$. This amounts to having zero chance of proving G_C and 20% chance of proving $\neg G_C$, i.e., $Bel(G_C) = 0$ and $Bel(\neg G_C) = 0.20$. Note that these answers are totally different than those computed with the D-S model, where we obtained $Bel(G_A) = 1/2$ even under $\boldsymbol{e} = L_B$ and even not knowing $P(L_B \mid G_A)$. The reason is that there, we modeled the trial process ($\pi(G_A) = 1/3$) as part of the meta-level reasoner, in a form of a random switch oscillating between G_A, G_B and G_C, while in the current exercise this information is included as part of the object-level theory, in order to retain its completeness.

We now address the question raised earlier, whether one can do more having a complete model as opposed to a partial model. The answer is, as anticipated, yes. Having a complete model at our disposal permits us to answer more sophisticated queries, not just about the provability of certain propositions. We can ask, for instance, what is the probability that the posterior probability $BEL(G_C) = P(G_C \mid \boldsymbol{e})$ will not exceed some constant α. The answer would yield a parametrized belief function:

$$Bel_{\alpha}(G_A) = P(e \models \text{``}BEL(G_C) \le \alpha\text{''}) = \begin{cases} 0.2 & 0 \le \alpha < 0.8 \\ 1 & 0.8 \le \alpha \le 1 \end{cases}$$

This may sound like computing probabilities of probabilities, a notion rejected in [Pearl, 987a] on the basis of being ill-defined. The separation between object-level and meta-level reasoning, now endows this notion with clear semantics, the semantics of hypothetical envisioning l ‹e the one used to reason about confidence levels (see [Pearl, 1987a]).

Why then hasn't the D-S theory incorporated such parametrized belief functions or confidence levels in its formalism? It turns out that to calculate these functions, one requires a complete probabilistic model at the object level and, when such a model is not available, primitive queries about provability is just about all that one can hope to answer.

!.3 Dempster's Rule of Combination

When several pieces of evidence are available, their impacts are combined by assuming hat the corresponding switches act independently of each other. For example, if in addition to 's alibi the trial records also include a testimony supporting A's guilt to a degree m_2, one ould imagine two random switches operating simultaneously and asynchronously; the first as c scribed in the preceding subsection, the second spending m_2 percent of the time constraining G_A to the value *true*, while staying neutral the rest of the time (see Figure 4).

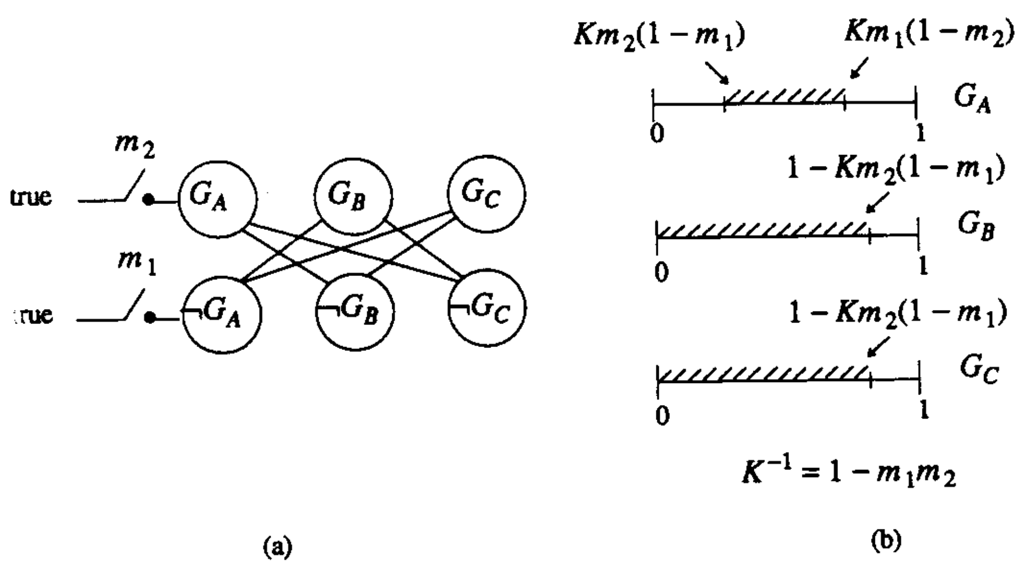


Figure 4. (a) Compatibility relations between the six elementary proprositions in the 3-prisoners puzzle. m_1 and m_2 represent the percentage of time each switch is closed.

(b) Belief intervals for the three propositions G_A, G_B and G_C.

Clearly, $m_1 m_2$ percent of the time the two switches are in conflict with each other; one is constraining G_A to true, the other constraining $\neg G_A$ to true, thus permitting no consistent extension. A fraction $(1 - m_1)m_2$ of the time G_A is true while switch-2 is neutral, rendering G_A, $\neg G_B$ and $\neg G_C$ provable. Similarly, a fraction $m_1(1 - m_2)$ of the time $\neg G_A$ is true while switch-1 is neutral, rendering $\neg G_A$ but no other proposition, provable. Summing up and normalizing by the no-conflict time $1 - m_1 m_2$ we have:

$$Bel(G_A) = Bel(\neg G_B) = Bel(\neg G_C) = \frac{m_2(1 - m_1)}{1 - m_1 m_2}$$

$$Bel(\neg G_A) = \frac{m_1(1 - m_2)}{1 - m_1 m_2}$$

$$Bel(G_B) = Bel(G_C) = 0$$

The assumption of evidence independence coupled with the normalization rule above, lead to an evidence pooling procedure known as *Dempster Rule of Combination*. The combined impact of several pieces of evidence could be calculated, again, by computing the fraction of time a given proposition A is compelled to be true by the combined action of all switches, assuming that they operate independently. Thus, the analysis of belief functions amounts to analyzing the set of extensions permitted by a network of static constraints (representing generic knowledge), subject to an additional set of externally imposed, fluctuating constraints, representing the impact of the available evidence. For any combination of the evidential constraints, we need to examine the set of extensions permitted by that combination and decide whether the proposition A is entailed by the set; i.e., if every extension contains A and none contains $\neg A$. The total time that a system spends under constraint combinations that compel A, divided by the total time spent in no-conflict combinations, yields $Bel(A)$.

The constraint network formulation of Dempster's combination rule is illustrated schematically in Figure 5.

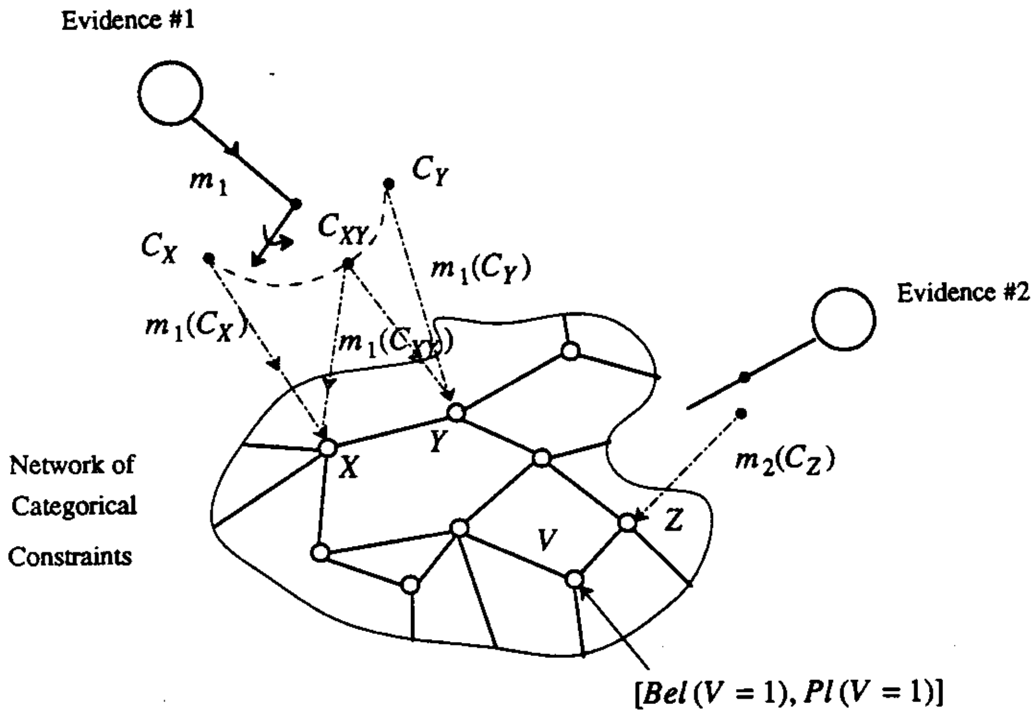


Figure 5. Multiple evidence modeled as random switches imposing additional constraints on a static network of compatibility relations.

It shows a static network of variables $X, Y, Z, V \cdots$ (the nodes) interacting via local constraints (the arcs), subject to the influence of two switches that impose additional time varying constraints on various regions of the network. To illustrate the analysis of the extension sets, let us assume that the static network represents the classical graph coloring problem: each node may take on one of three possible colors, 1, 2, or 3, but no two adjacent nodes may take on identical colors. The position of the switches represents additional constraints e.g., C_{XY}: either X or Y must contain the color 1, or, C_Z: Z cannot be assigned the color 2 etc. The relative time that a switch spends enforcing each of the constraints is indicated by the weight measures $m_1(C_X)$, $m_1(C_{XY})$, $m_2(C_Z)$, etc. Our objective is to compute $Bel(A)$ and $Pl(A)$, where A stands for the proposition $V = 1$, namely, variable V is assigned the color 1.

Figure 6 represents typical sets of solutions to the coloring problem under different combinations of the switches.

$$\begin{array}{lcl} & VXY \cdots & \\ \text{Type-1 positions} & \begin{bmatrix} 1\,2\,3 \cdots \\ 1\,1\,2 \\ 1\,3\,2 \end{bmatrix} & V = 1 \text{ in all solutions} \\ \text{Time} = \alpha & & \end{array}$$

Type-2 positions
Time = β

$$\begin{bmatrix} 1\,2\,1 & \cdots \\ 2\,3\,1 & \\ 2\,2\,3 & \cdot \end{bmatrix} \quad \begin{bmatrix} 3\,2\,1 & \cdots \\ 1\,2\,1 & \end{bmatrix}$$

$V = 1$ and $V \neq 1$ are compatible with each position

Type-3 positions
Time = γ

$$\begin{bmatrix} 2\,1\,3 & \cdots \\ 2\,3\,1 & \\ 3\,3\,3 & \end{bmatrix}$$

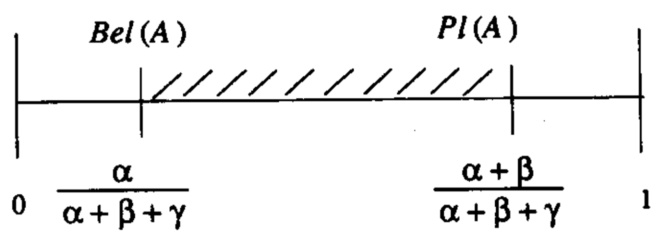
$V \neq 1$ in all solutions

Type-4 positions
Time = δ

$$\begin{bmatrix} \text{Nil} \end{bmatrix}$$

no solution

(a)



(b)

Figure 6. (a) Four types of constraints in the graph coloring problem and

(b) the resulting belief interval for the proposition $A : V = 1$

Each row represents one extension (or solution) where the entries indicate the value assigned to the variables (columns). The first set of solutions are characterized by having the value 1 assigned to V in each and every row. If the system spends a fraction α of the time in such combinations of switches, we say that $P[e \models V = 1] = \alpha$ namely, the proposition $A : V = 1$ can be proven true with probability α, given the evidence e. A type-2 position is characterized by the column of V containing 1's as well as alternative values e.g., 2 and 3. Each such position (or position combination) is compatible with both A and $\neg A$. Similarly, a type-3 position permits only extensions that exclude $V = 1$, while a type-4 position represents conflict situations; there exists no extension consistent with all the constraints. $Bel(A)$ and $Pl(A)$ are computed from the times spent in each type of constraint combination:

$$Bel(A) = \frac{\alpha}{\alpha + \beta + \gamma}$$

$$Pl(A) = 1 - Bel(V \neq 1) = 1 - \frac{\gamma}{\alpha+\beta+\gamma} = \frac{\alpha+\beta}{\alpha+\beta+\gamma} .$$

These are illustrated as a belief interval in Figure 6(b).

The proceeding analysis can be rather complex. The graph coloring problem, even with only three colors, is known to be NP complete. Moreover, if each piece of evidence is modeled by a 2-position switch and if we have n such switches, then a brute force analysis of $Bel(A)$ would require solving 2^n graph coloring problems. Listing the solutions obtained under all switch combinations and identifying those combinations yielding $e \models A$ seems hopeless. Fortunately, two factors help alleviate these difficulties: the sequential nature of Dempster's rule and the ability to exploit certain topological properties of the constraint network. The latter revolves around the idea of decomposing the network into a tree of clusters, where solutions can be obtained in linear time [Dechter & Pearl, 1987]. Adaptations of tree decomposition to belief function computations are reported in [Shafer et al, 1987b] and [Tung & Kong, 1987].

Dempster's rule, being associative and commutative, permits multiple evidence to be combined sequentially without enumerating all switch combinations encountered in the past. It is based on the fact that if two distinct switch combinations give rise to the same set of solutions, we can replace the two by a single equivalent constraint that allows exactly that set of solutions, for the total amount of time that the two combinations lasted. Thus, instead of recording all distinct switch combinations, it is sufficient to record all distinct solution sets induced by the combinations, and keep track of their weights.

The latter scheme will sometimes require much less space, especially under conditions of tight constraints where many switch combinations would yield no solution. Metaphorically, the set of recorded solution sets and their associated weights are equivalent to a single giant switch, with one position per distinct solution set. The impact of each new piece of evidence e'' can be calculated by first calculating the constraints accumulated by all the previous evidence e', then combining it with the constraints created by e'' itself, as if no other evidence was in existence. If the former induces a belief function Bel' and the latter Bel'', the result of combining the two by Dempster's rule is denoted by $Bel' \oplus Bel''$ and is called the *orthogonal sum* of Bel' and Bel''. Mathematically, for any proposition B, $Bel' \oplus Bel''(B) = Bel(B)$ can be computed from $m(A) = m' \oplus m''(A)$, using Eq.(3), where $m(A)$, reflecting the constraints imposed by both e' and e'', is given by

$$m(A) = m' \oplus m''(A) = K \sum_{A_1 \wedge A_2 = A} m'(A_1)\, m''(A_2) \qquad A \neq \varnothing \tag{5}$$

and

$$K^{-1} = \sum_{A_1 \wedge A_2 \neq \varnothing} m'(A_1)\, m''(A_2) \tag{6}$$

In other words, the weight $m(A)$ assigned to solution set A is the sum over all pairs of solution sets A_1, A_2 whose intersection is A. Multiplying $m'(A_1)$ by $m''(A_2)$ reflects the independence assumption; the probability that the two constraints will be enforced together is equal to the product of the probabilities that each holds separately. The intersection $A_1 \wedge A_2$ reflects the fact that the solution set resulting from the simultaneous imposition of two constraints is the intersection of the two solution sets obtained under the individual constraints. K is a normalizing factor guaranteeing that $\sum_A m(A) = 1$, and serves to discount the weights assigned to conflicting constraints (i.e., $A_1 \wedge A_2 = \emptyset$) and redistribute it equally among the remaining solution sets.

Figure 7 illustrates the essence of Dempster's rule using the two-switch model of Figure 6. Assume that the evidence e', represented by the two switches in Figure 6, permits the following set of four distinct solution sets:

$$A'_1 = \begin{Bmatrix} 1 & 2 & 3 \\ 1 & 1 & 2 \\ 1 & 3 & 2 \end{Bmatrix} \quad A'_2 = \begin{Bmatrix} 1 & 2 & 1 \\ 2 & 3 & 1 \\ 2 & 2 & 3 \end{Bmatrix} \quad A'_3 = \begin{Bmatrix} 1 & 2 & 1 \\ 3 & 2 & 1 \end{Bmatrix} \quad A'_4 = \begin{Bmatrix} 2 & 1 & 3 \\ 2 & 3 & 1 \\ 3 & 2 & 3 \end{Bmatrix},$$

with corresponding weights (m'_1, m'_2, m'_3, m'_4). As before, the columns represent the variables V, X, Y (with the other variables ignored for convenience) and the entries stand for the colors assigned to these variables. Set A'_1 is of type-1, relative the proposition $A: V = 1$, A'_2 and A'_3 are of type-2, while A_4 is of type-3. Now assume that a new piece of evidence e'' is obtained, represented by a 3-position switch which, in isolation, gives rise to the following three sets of solutions:

$$A''_1 = \begin{Bmatrix} 1 & 3 & 2 \\ 1 & 1 & 2 \\ 1 & 2 & 1 \end{Bmatrix} \quad A''_2 = \begin{Bmatrix} 1 & 2 & 3 \\ 2 & 2 & 3 \end{Bmatrix} \quad A''_3 = \begin{Bmatrix} 2 & 3 & 2 \end{Bmatrix}$$

with corresponding weights (m''_1, m''_2, m''_3). The combined effect of e' and e'' is shown in Figure 7. It displays the intersections of all pairs of solution sets, A'_i, A''_j where A'_i is taken from e' and A''_j from e''. A total of four distinct solution sets survive the intersection, while seven intersections turn out empty. The weight destined for these empty subsets is equal to

$$1 - K^{-1} = m'_4(m''_1 + m''_2 + m''_3) + m''_2 m'_3 + m''_3(m'_1 + m'_2 + m'_3) =$$

$$= m'_4 + m''_2 m'_3 + m''_3 - m''_3 m'_4$$

(using $\sum m(A) = 1$), and will serve to normalize the weights of the surviving solution sets.

To calculate $Bel(V = 1)$, we combine the weight of the three type-1 subsets, $\{1\ 2\ 1\}$, $\begin{Bmatrix} 1 & 3 & 2 \\ 1 & 1 & 2 \end{Bmatrix}$ and $\{1\ 2\ 3\}$, and divide by K^{-1}:

$$Bel(V = 1) = \frac{m''_1(m'_1 + m'_2 + m'_3) + m''_2 m'_1}{1 - (m'_4 + m''_2 m'_3 + m''_3 - m''_3 m'_4)}$$

$A'_4 = \begin{Bmatrix} 2 & 1 & 3 \\ 2 & 3 & 1 \\ 3 & 2 & 3 \end{Bmatrix}$ (m'_4)	$\varnothing$	$\varnothing$	$\varnothing$
$A'_3 = \begin{Bmatrix} 1 & 2 & 1 \\ 3 & 1 & 1 \end{Bmatrix}$ (m'_3)	$\begin{Bmatrix} 1 & 2 & 1 \end{Bmatrix}$ $(m'_3 m''_1)$	$\varnothing$	$\varnothing$
$A'_2 = \begin{Bmatrix} 1 & 2 & 1 \\ 2 & 3 & 1 \\ 2 & 2 & 3 \end{Bmatrix}$ (m'_2)	$\begin{Bmatrix} 1 & 2 & 1 \end{Bmatrix}$ $(m'_2 m''_1)$	$\begin{Bmatrix} 2 & 2 & 3 \end{Bmatrix}$ $(m'_2 m''_2)$	$\varnothing$
$A'_1 = \begin{Bmatrix} 1 & 2 & 3 \\ 1 & 1 & 2 \\ 1 & 3 & 2 \end{Bmatrix}$ (m'_1)	$\begin{Bmatrix} 1 & 3 & 2 \\ 1 & 1 & 2 \end{Bmatrix}$ $(m'_1 m''_1)$	$\begin{Bmatrix} 1 & 2 & 3 \end{Bmatrix}$ $(m'_1 m''_2)$	$\varnothing$
	$A''_1 = \begin{Bmatrix} 1 & 3 & 2 \\ 1 & 1 & 2 \\ 1 & 2 & 1 \end{Bmatrix}$ (m''_1)	$A''_2 = \begin{Bmatrix} 1 & 2 & 3 \\ 2 & 2 & 3 \end{Bmatrix}$ (m''_2)	$A''_3 = \begin{Bmatrix} 2 & 3 & 2 \end{Bmatrix}$ (m''_3)

Figure 7. Dempster's Rule of Combination

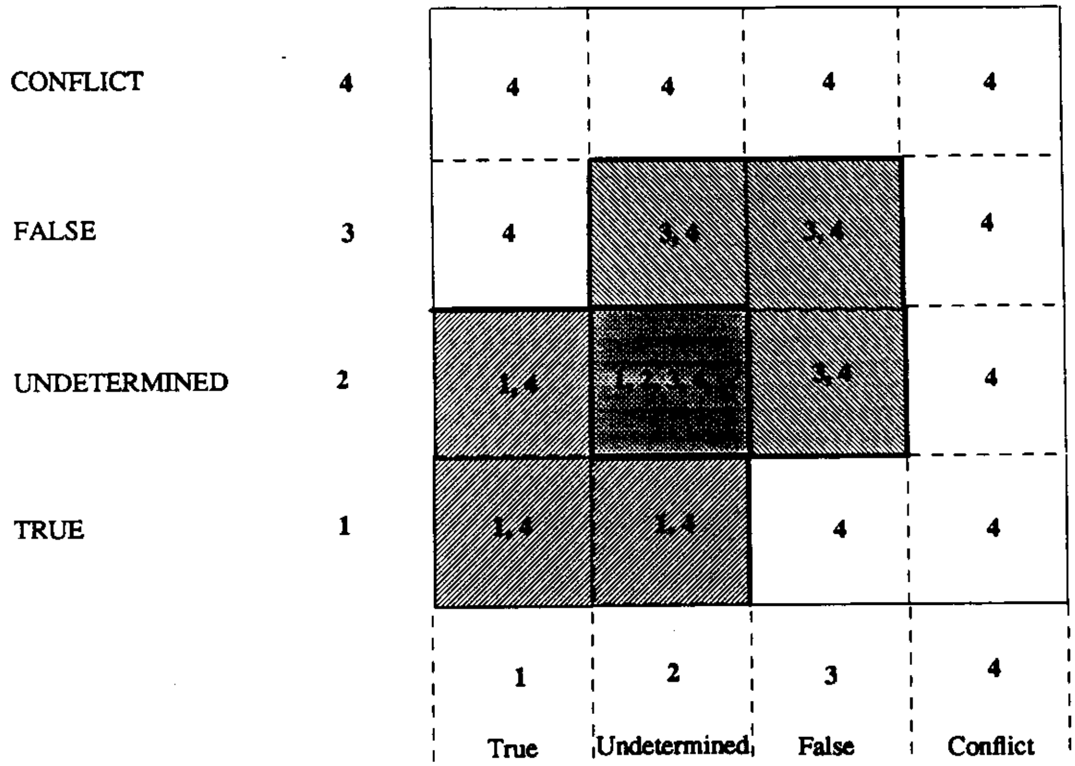


Figure 8. Set Intersection in Dempster's Rule of Combination

Figure 8 displays the type of solution sets that emerge from Dempster's rule. The solution sets of each evidence are grouped into four types and the type of the intersection is indicated in the table. In principle, the intersection of any two subsets can be empty, i.e., type-4. In addition, a type-1 set combined with either a type-1 or a type-2 set can yield only a type-1 set (solutions compatible with $\neg A$ are ruled out by type-1). Similarly, the intersection of type-3 with either type-3 or type-2 must yield a type-3 set (or nil). However, the intersection of two type-2 subsets may be of any type. Manifestly, type-2 solution sets can only emerge from the intersection of two type-2 sets. Hence, once type-2 sets become extinct (i.e., by exposure to a zero-interval evidence) they will remain extinct forever.

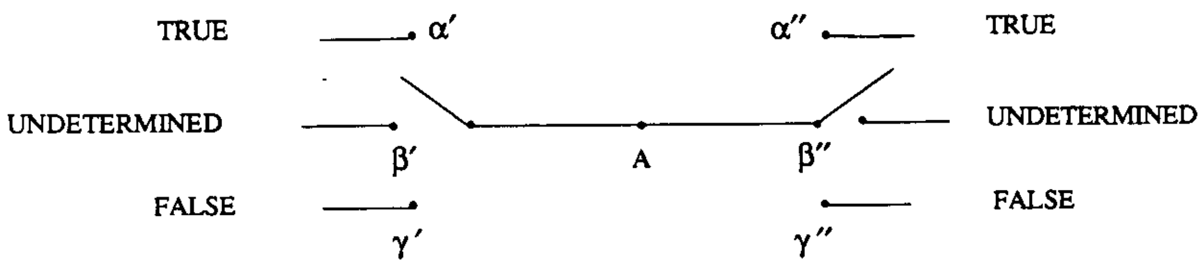


Figure 9. A 2-switch model for two pieces of evidence bearing on the same proposition, A.

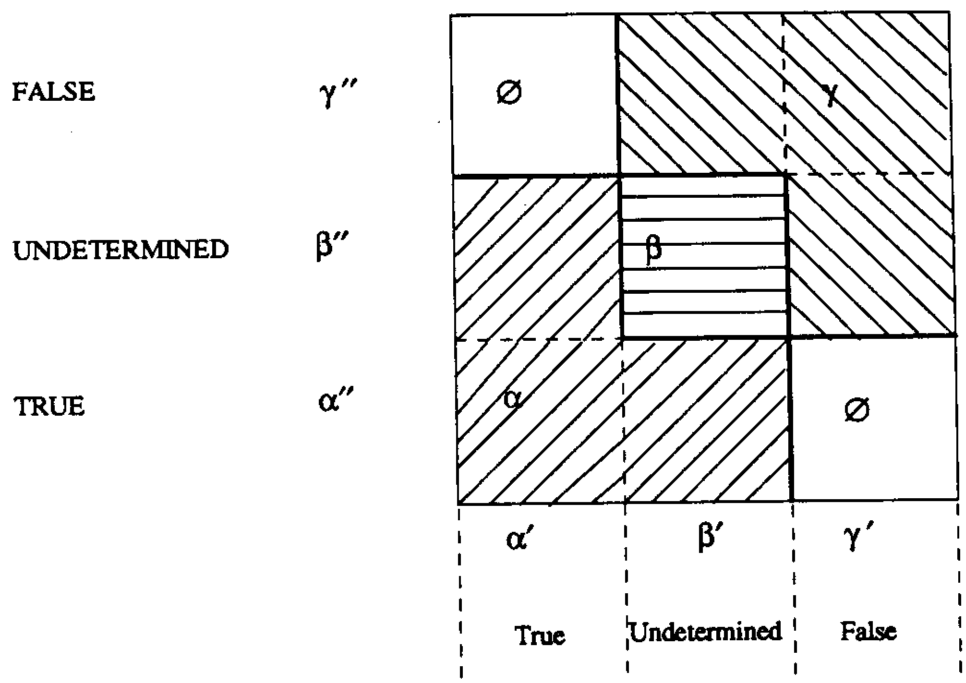


Figure 10. Dempster's Rule of Combination for the 2-Switch Model of Figure 9.9.

Dempster's rule assumes a particularly convenient form when several pieces of evidence bear on the same proposition (or its negation). This is illustrated in Figure 9, using, again, the two switch model. Each switch is characterized by three parameters (α, β, γ), indicating the fraction of the time that the switch spends in each state. In the first state, called TRUE, A is forced to true, in the UNDETERMINED state the evidence is compatible with both A and $\neg A$, while the third forces A to FALSE. The semantics of these states define what state is created by any combination of them. For example, a FALSE state combined with an UNDERTERMINED state yields a FALSE state because all extensions compatible with A would be excluded by the former. The weights assigned to the combined states are indicated by their corresponding areas in the diagram of Figure 10. For example, $Bel' \otimes Bel''(A)$ is given by the sum of the three areas labeled α, divided by the active areas $\alpha + \beta + \gamma$. These yield:

$$Bel(A) = Bel' \oplus Bel''(A) = \frac{\alpha}{\alpha + \beta + \gamma} = \frac{\alpha'\alpha'' + \alpha'\beta'' + \alpha''\beta'}{1 - \alpha'\gamma'' - \alpha''\gamma'}$$

$$Pl(A) = Pl' \oplus Pl''(A) = 1 - \frac{\gamma}{\alpha + \beta + \gamma} = 1 - \frac{\gamma'\gamma'' + \gamma'\beta'' + \beta'\gamma''}{1 - \alpha'\gamma'' - \alpha''\gamma'}$$

Expressed in terms of the belief parameters $[Bel, Pl]$, we use

$$b' = Bel'(A) \qquad p' = Pl'(A)$$

$$b'' = Bel''(A) \qquad p'' = Pl''(A)$$

and obtain

$$Bel(A) = \frac{p'p'' - (p' - b')(p'' - b'')}{1 - [b'(1 - p'') + b''(1 - p')]} \qquad Pl(A) = \frac{p'p''}{1 - [b'(1 - p'') + b''(1 - p')]} \tag{7}$$

This combination rule, constitutes a convenient calculus for D-S intervals [Ginsberg, 1984, Hajek, 1986]. To compute the interval $[Bel, Pl]$ of proposition A, it is only necessary to compute the parameters Bel, Pl associated with each separate piece of evidence, then combine them incrementally using Eq.(7).

1.4 More on the Nature of Probability Intervals

At this point, it is worthwhile reflecting on the significance of the interval $Pl(A) - Bel(A)$ in the D-S formalism. This interval is often interpreted to signify the degree of ignorance we have about probabilities, namely, the amount of information needed in order to construct a complete probabilistic model. If this were so, then the D-S approach would have an advantage over Bayes methods, which always provide point probabilities and so, might give one a false sense of security in the model.

Unfortunately, the D-S intervals have little to do with ignorance, nor do they represent *bounds* on the probabilities that would ensure once ignorance is removed. This was already demonstrated in the 3-prisoners puzzle (Section 1.1). We saw that despite our total ignorance regarding the process by which the jailer chose the letter recipient, the interval $Pl(G_A) - Bel(G_A)$ was zero, thus giving one the false impression that the answer $Bel(G_A) = 1/2$ is based on a complete model (with the jailer attempting to avoid C whenever possible). At the same time, knowledge of the selection process proved essential in Bayes analysis, as it could sway the posterior probability $P(G_A \mid e)$ all the way, from zero to $1/2$.

Figure 8 and Equation (7) reveal that the disappearance of the D-S interval is not an isolated incident but will occur whenever a piece of evidence imparts all its weight to a proposition and its negation. In other words, if e'' induces $Pl''(A) = Bel''(A)$, then regardless of the ignorance we possessed before (i.e., $Pl'(A) - Bel'(A)$) and regardless of any ignorance that might be conveyed by future evidence, $Pl(A) - Bel(A)$ will remain zero forever. In particular, whenever we start with a complete probabilistic model (where the belief interval is zero) no amount of conflicting evidence will ever succeed in opening that interval to reflect the conflict.

The ramification of this effect is that many sources of ignorance or uncertainty about probabilities are not represented in the D-S formalism. In particular the uncertainty caused by high sensitivity to unknown contingencies, which are perfectly reflected in the structure of causal networks, is not represented by belief intervals. Reiterating the example from [Pearl, 1987a], suppose we know that a given coin was produced by a defective machine -- precisely 49% of its output consists of double-head coins, 49% are double-tail coins, and the rest are fair. This description constitutes a complete probabilistic model which predicts that the outcome of the next toss will be head with probability 50%, and alerts us to the fact that the prediction is extremely susceptible to new information regarding the nature of the coin. While most people will hesitate to commit a point estimate of 50% to the next outcome of the coin, the D-S theory, nevertheless, assigns it a belief of 50%, with zero belief interval. Now imagine that we toss the coin twice and observe a tail and a head. This immediately implies that the coin is fair and, hence, most people would regain confidence to assign the next toss a 50% chance of turning up head. Yet, such narrowing of confidence interval would remain unnoticed in the D-S formalism; the theory will again assign the next outcome a belief of 50% with zero belief interval.

The vanishing of the difference $Pl - Bel$ in the 3-prisoners puzzle is a by-product of the normalization used in Dempster's rule. Refraining from this normalization would have yielded an interval $[1/3, 2/3]$ for G_A, reflecting the fact that both G_A and $\neg G_A$ can each be proven only one third of the time (assuming no proposition is truly provable from a contradiction). Indeed, the normalization by the no-conflict time stands at odds with the basic definition of *Bel* as the probability of being able to force a proof. The normalized version of *Bel* no longer reflects this intended probability but, rather, the probability of forcing a proof, conditioned upon having a non-empty set of extensions. Valuable information seems to get lost in the process of this conditionalization. A more reasonable approach would be to keep two intervals; one measuring the degree of conflict and one measuring the degree of evidence non-commitment. That would entail characterizing each proposition by four parameters, corresponding to the 4 types of solution sets (see Figure 10).

Another criticism of the normalization used in the D-S approach was advanced by Zadeh (1984), using the following example:

> Suppose that a patient, P, is examined by two doctors, A and B. A's diagnosis is that P has either meningitis, with probability 0.99, or brain tumor, with probability 0.01. B agrees with A that the probability of brain tumor is 0.01, but believes that it is the probability of concussion rather than meningitis that is 0.99. Ap-

plying the Dempster rule to this situation leads to the conclusion that the belief that P has brain tumor is 1.0 - a conclusion that is clearly counterintuitive because both A and B agree that it is highly unlikely that P has a brain tumor. What is even more disconcerting is that the same conclusion, (i.e., *Bel* (brain tumor) = 1) would obtain regardless of the probabilities associated with the other possible diagnoses.

Abstractly, this example involves two pieces of evidence bearing on a variable S with three values:

$$S = \{\text{meningitis, tumor, concussion}\}$$

The first rules out a concussion, and the second rules out meningitis, so taken together they leave a tumor as the only possibility.

[Shafer, 1987a] has argued that the conclusion is not unreasonable, given our assumptions. If our experts are absolutely reliable and if we accept the initial assumption that meningitis, tumor, and concussion are the only possibilities,

> "then it is a matter of logic, not merely probability, that the patient must have a tumor. As Sherlock Holmes put it, when you have eliminated the impossible, whatever remains, however improbable, must be the truth."

While Shafer's defense is justified in the example cited (Bayes analysis would yield the same result), nevertheless, normalization tends to disproportionally emphasize points of agreement and deemphasize disagreements between items of evidence. The net result being that belief intervals tend to narrow down much faster than they should, had we adhered to their original definition of beliefs as probabilities of provability.

1.5 Applications to Rule-Based Systems

Formulating the Dempster-Shafer theory in terms of constraint networks becomes more natural when the constraints are expressed in rule form. A rule, r, is a constraint among a group of propositions, having an *if-then* format:

$$r:\ a_1 \,\&\, a_2 \,\&\, \cdots\, a_n \Rightarrow c \qquad (8)$$

Propositions $a_1 \cdots a_n$ are called the antecedents (or justifications) of the rule, and c is its consequent. The semantics of the rule lies in forbidding any extension in which the antecedents are all true while the consequent is false, in other words, a rule is equivalent to the constraint

$$r:\ \neg[a_1\ \&\ a_2\ \&\ \cdots\ a_n\ \&\ \neg c] \tag{9}$$

Normally, rules are based on tacit *assumptions*, the failure of which (called *exceptions*) may invalidate the rule. For example, I may assert the rule r: "If it is Sunday John goes to the ball game", tacitly assuming the prerequisites: "John is still unmarried" "John is not sick" etc. Since such assumptions are too numerous to explicate, they are often summarized by giving the rule a measure of strength, m. For example, the rule above might be given a strength of $m = 0.8$ indicating a 80% assurance that none of the implicit exceptions will materialize.

In the D-S formalism, the strength m translates to a switch that spends a fraction m of the time imposing the constraint conveyed by the rule . The activity of the switch during the remaining time depends on the nature of the exceptions anticipated. Some exceptions (e.g., John being sick) lead to the negation of the conclusion while others (e.g., John is married) renders the conclusion unknown or uncommitted. In the former case the switch will force the negation of the conclusion while in the latter case it will spend the time in a neutral position. Thus, the rule author must be aware of the type of assumptions summarized by the rule strength. For the sake of simplicity we will characterize rules by a simple switch model, lending support to its consequence c but not to its negation. The more sophisticated model, such as the 3-position switches in Figure 9, yields essentially the same results.

Combining Belief Functions in Rule Networks

Assume we have a system of rules, a list F of observational facts called *premises*, and we wish to find the belief $Bel(\cdot)$ attributable to some proposition c. This amounts to computing the probability that a proof exists between the premises in F and the conclusion c. Each proof consists of a sequence of rules $r_1, r_2, \cdots r_m$ such that the antecedents in each r_i are either premises or are proven and the consequence of r_m is the desired conclusion c. Graphically, a proof can be represented by a directed acyclic graph, like the one shown in Figure 11, where the root nodes are all premises, the leaf node is c, and each bundle of converging arrows represent a given rule. The arcs connecting the arrows represent the logical AND function between the antecedents of each rule.

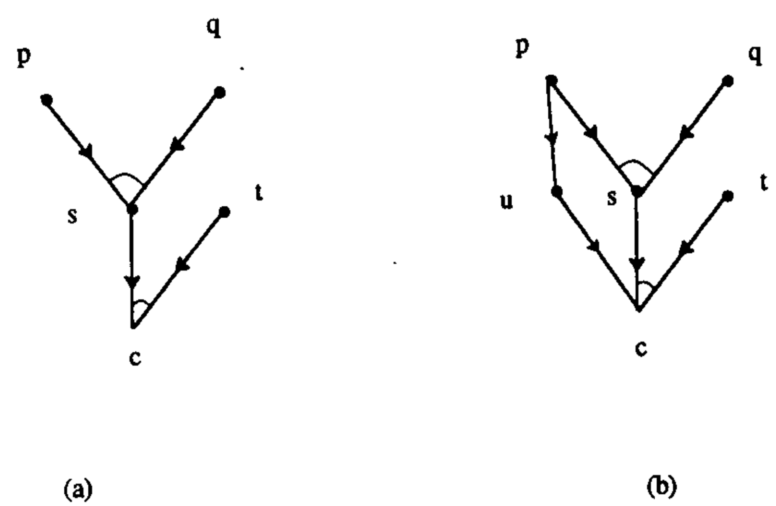


Figure 11 (a) A proof graph for proposition c, representing two rules

$p \,\&\, q \rightarrow s$, $s \,\&\, t \rightarrow c$ and the premises p, q, t.

(b) An AND/OR graph representing four rules

The collection R of all rules available to a system can be represented by an AND/OR graph like the one in Figure 11(b), where an OR function is understood to exist between any two parent-bundles converging toward the same node. The graph in Figure 11(b) contains two proofs for c, (r_1, r_2) and (r_3, r_4). In case t can no longer be asserted as a premise, the proof (r_1, r_2) is no longer valid and c can be proven via (r_3, r_4).

We are now in a position to calculate $Bel(c)$, namely, the probability that c is provable in a system of uncertain rules, where each rule r_i is characterized by a strength measure m_i. In the D-S formalism, a system of uncertain rules is equivalent to an AND/OR graph intercepted by a collection of random switches, as shown in Figure 12. The task of computing $Bel(c)$, then, amounts to calculating the percentage of time that some proof graph remains unintercepted, between the premises F and the conclusion c. In the special case where every rule has a single antecedent, the problem reduces to that of finding the percentage of time that an unintercepted path exists between some premise and the conclusion. Such problems have been studied extensively in the area of network reliability and, in general, they turn out to be rather complex, even under the assumption that the interruptions are independent of each other [Grnarov, 1979].

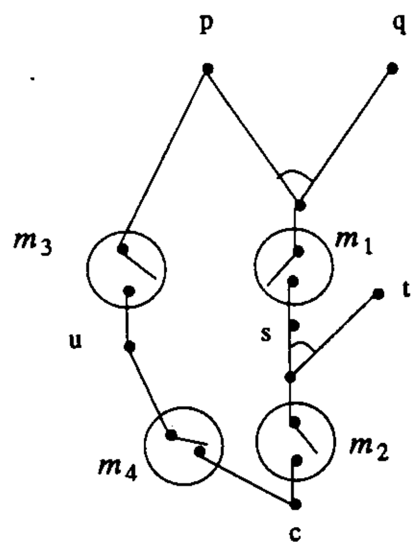


Figure 12. Random switch model for the rule network of Fig. 9.9(b).

A brute-force way of calculating $Bel(c)$ would be to enumerate all switch combinations, test if a proof exists in each combination, then total the times spends in combinations that pass the test. For a system with n rules, this would require the enumeration of 2^n combinations. Fortunately, the simple nature of the network in Figure 12, permits the calculation to be done without enumerating all combinations. Since the network contains two disjoint proofs, the active times of the two proof graphs are independent, hence the time that c is non-provable is equal to the product of the times that each of the two proof graphs is inactive, i.e., $(1 - m_3 m_4)(1 - m_1 m_2)$. The rest of the time c is provable, hence

$$Bel(c) = 1 - (1 - m_3 m_4)(1 - m_1 m_2) \tag{10}$$

Note that in our example, instead of enumerating all 2^4 switch positions, it was necessary to enumerate only the two proof paths (in general, proof graphs), (r_1, r_2) and (r_3, r_4), calculate the active time t_i of each path, then calculate $Bel(c)$ using the formula

$$Bel(c) = 1 - \prod_i (1 - t_i) \tag{11}$$

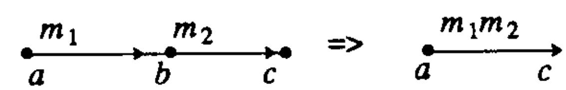
Such shortcuts will not be feasible in general rule networks. For example, if we add to the system of Figure 11 the rule, r_5: $s \rightarrow u$ (m_5), an additional proof graph is added $\{r_1, r_5, r_4\}$ whose activation time is dependent on the other proof graphs and we can no longer calculate $Bel(c)$ by multiplying together the inactive times of the three proof graphs separately. Rather, we would have to enumerate all distinct ways that at least one proof graph remains active, e.g.,

$$Bel(c) = 1 - (1 - m_3 m_4)(1 - m_1 m_2) + m_1 m_5 m_4 (1 - m_2)(1 - m_3)$$

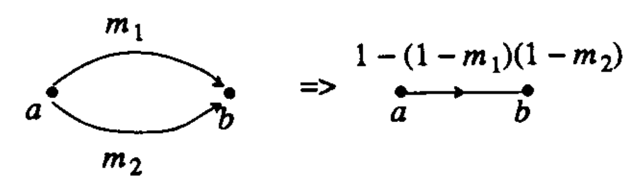
The first term represents the condition that at least one of the proofs, (r_1, r_2), (r_3, r_4), is active, while the second represents the proof remaining under the complementary condition.

The topological feature that permits shortcuts as in the network of Figure 11(b) is a property called *series-parallel.* This features enables recursive computations of network flow, network reliability and, naturally, belief functions and probabilities in rule-based system. Formally, a rule network is said to be series-parallel if it can be reduced to a single rule by repeated application of the following two operations:

1. Series reduction -



2. Parallel reduction -



It is clear from this definition that series-parallel rule networks permit the calculation of belief functions in time proportional to the number of rules (as opposed to the number of switch combination), since each reduction operation reduces the number of rules by one. For example, the network of Figure 13 can be reduced in three operations, yielding

$$Bel(c) = m_1(m_2 || m_3 \cdot m_4) = m_1[1-(1-m_2)(1-m_3 m_4)] \tag{13}$$

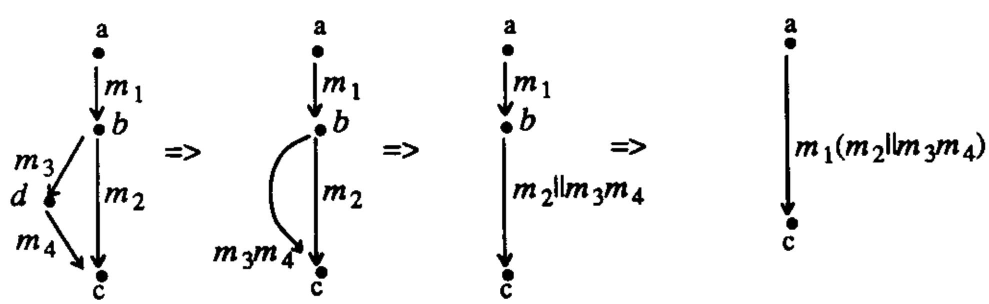


Figure 13. Reducing a series-parallel rule network to a single rule ($m_1 || m_2$ stands for $1-(1-m_1)(1-m_2)$).

However, in general network of rules, the calculation of $Bel(c)$ may require exponential time.

The Limits of Extensional Systems

The foregoing analysis delineates the class of D-S systems where extensional techniques are valid. In an extensional system the uncertainty associated with the consequent of each rule, is solely a function of the uncertainties associated with the antecedents of the rule and that associated with the rule itself. In the D-S versions of such systems these uncertainties are represented by a pair of supports $[b, p]$ and, so, the pair associated with the consequent of each rule is presumed to be a function of only the pair which characterizes the rule and those which characterize the antecedents [Ginsberg, 1984, Baldwin, 1987]. Moreover, when two rules converge toward the same conclusion the pair associated with the latter is determined from those of the individual rules via the interval calculus of Eq.(7). This is precisely where extensional systems deviate from the principles dictated by the D-S theory. The combination rules of Eq.(7) were derived under the assumption that the two items of evidence are independent of each other (i.e., the two switches in Figure 9 work independently). Applying the rule (or any uniform combination rule) to every pair of converging arrows in a large network may violate this independence assumption, especially if the proof paths overlap. For example, given the truth of a in the initial network of Figure 13, an extensional analysis will compute $Bel(c)$ as follows:

$$Bel(b) = m_1$$

$$Bel(d) = m_3 Bel(b) = m_3 m_1$$

$$\begin{aligned} Bel(c) = m_4 Bel(d) \| m_2 Bel(b) &= m_4 m_3 m_1 \| m_2 m_1 \\ &= 1 - (1 - m_4 m_3 m_1)(1 - m_2 m_1) \\ &= m_1(m_4 m_3 + m_2 - m_4 m_3 m_2 m_1) \end{aligned} \quad (14)$$

The correct result should be

$$Bel(c) = m_1(m_2 \| m_3 m_4) = m_1(m_2 + m_3 m_4 - m_2 m_3 m_4),$$

as in Eq. (13). The difference between the two expressions is equal to $m_2 m_3 m_4 (m_1 - m_1^2)$ and, clearly, it stems from counting the arc m_1 twice. An extensional system is too local to realize that the beliefs at b and d originate from the same evidence.

Such analysis enables us to quickly come up with conceptual examples that amplify the discrepancy between the two approaches and thus, highlight the conditions under which using extensional systems leads to paradoxical conclusions. To maximize the difference $m_1 - m_1^2$), we let $m_1 = ½$ and $m_2 = m_3 = m_4 = 1$, and assemble the following system of rules:

r_1: If I flip the coin (a), then it will turn up head (b), ($m_1 = ½$)

r_2: If the coin turns up head (b), then you win (c), ($m_2 = 1$)

r_3: If the coin turns up head (b), then I lose (d), ($m_3 = 1$)

r_4: If I lose (d), then you win (c), $(m_4 = 1)$

Suppose I flip the coin (a = true), what is the belief attributable to your winning (c)? The correct answer is clearly ½, since the path $b \rightarrow d \rightarrow c$ is superfluous. Yet, the answer computed by an extensional system would be $Bel(c) = 3/4$, as if my loss contributes an extra piece of evidence toward your winning.

Relations to Bayes Analysis

It is interesting to note that a Bayesian analysis will produce the same result as (10) and (11) under the assumptions that:

1. A rule $r: a \rightarrow b \ (m)$ is interpreted as two conditional probability statements: $P(b \mid a) = m, P(b \mid \neg a) = 0$

2. Converging rules interact disjunctively, via the noisy-OR model (see [Pearl, 1987b]).

These two assumptions permit the construction of a complete probabilistic model (i.e., a Bayes network) for any acyclic rule network. The probabilities $BEL(A) = P(A \mid e)$ calculated from such models are identical to the belief functions $Bel(A)$ calculated from the D-S model, for any proposition A in the rule set. However, the negations of these propositions obtain the probabilities $BEL(\neg A) = 1 - BEL(A)$ while in the D-S model they are assigned zero $Bel(\cdot)$ values ($\neg A$ cannot be proven by a rule set unless $\neg A$ appears as a consequent of at least one rule).

1.6 Bayesian vs. Dempster-Shafer: A Semantic Clash

The essential difference between the Bayesian and D-S interpretations of the rules shows up in systems involving a mixture of conflicting rules; some supporting a proposition A and some supporting its negation, $\neg A$. In such systems the semantic clash between the two approaches leads to qualitatively different conclusions. Whereas the D-S scheme resolves conflicts by the bold and uniform mechanism of Dempster's normalization (see Figure 9), the Bayes approach resolves them by a more cautious mechanism, appealing to their semantics. As a result, the D-S approach will inherent all the problems of classical monotonic logic when applied to situations requiring belief revision. We shall demonstrate these problems using a simple, 3-rule example, called the ***penguin triangle***.

Consider the rule set:

R: r_1: $p \rightarrow \neg f \quad (m_1)$, meaning "Penguins normally don't fly";

r_2: $b \rightarrow f \quad (m_2)$, meaning "birds normally fly";

r_3: $p \rightarrow b \quad (m_3 = 1)$, meaning "penguins are birds"

To emphasize our strong conviction in these rules, we make m_1 and m_2 close to unity and write

$$m_1 = 1 - \varepsilon_1 \quad m_2 = 1 - \varepsilon_2$$

where ε_1 and ε_2 are small positive quantities. Assume we find an animal, called Tweety, that is categorically classified as a bird and a penguin, and that we wish to assess the likelihood that Tweety can fly. In other words, we are given the premises p and b, and we need to compute $Bel(f)$, using the D-S approach, or $P(f \mid p, b)$, using the Bayesian approach.

The Bayesian approach immediately yields the expected result, namely, that Tweety's birdness does not render Tweety a better flyer than an ordinary penguin. The reason is that the entailment $p \supset b$ permits us to replace $P(f \mid p, b)$ by $P(f \mid p)$, giving

$$P(f \mid p, b) = P(f \mid p) = 1 - P(\neg f \mid p) = 1 - m_1 = \varepsilon_1 \tag{15}$$

In the D-S approach, on the other hand, if we treat the rules as a system of switches randomly oscillating between a TRUE and NEUTRAL position (as in the rule network of Figure 12), a counter-intuitive result obtains; birdness seems to endow Tweety with extra flying power. This is shown in the following table, where the four states of the rules r_1 and r_2 are enumerated together with their associated probabilities and the provability state of the proposition *fly*.

Probabilities	r_1	r_2	*fly*	$\neg$*fly*
$\varepsilon_1\varepsilon_2$	neutral	neutral	not provable	not provable
$(1-\varepsilon_1)\varepsilon_2$	true	neutral	not provable	provable
$\varepsilon_1(1-\varepsilon_2)$	neutral	true	provable	not provable
$(1-\varepsilon_1)(1-\varepsilon_2)$	true	true	conflict	conflict

Summing over the states where *fly* is provable and normalizing, we obtain

$$Bel(fly) = \frac{\varepsilon_1(1-\varepsilon_2)}{1-(1-\varepsilon_1)(1-\varepsilon_2)} = \frac{\varepsilon_1 - \varepsilon_1\varepsilon_2}{\varepsilon_1 + \varepsilon_2 - \varepsilon_1\varepsilon_2} \approx \frac{\varepsilon_1}{\varepsilon_1 + \varepsilon_2} \tag{16}$$

We see that the belief attributable to Tweety's flying, critically depends on whether she is a penguin bird, or just a penguin. In the latter case, rule r_1 dictates $Bel(fly) = \varepsilon_1$, which is negligibly small. In the former case, adding the superfluous information that all penguins are birds and birds normally fly, renders $Bel(fly)$ subtantially higher, as in (16). It does not go to zero with ε_1 and ε_2, but depends on the relative magnitudes of these quantities. If the proportion of non-flying birds (ε_2) is smaller than the proportion of flying penguins (ε_1), Tweety's flying will be assigned a belief measure greater than 0.5. Equipping the switches with FALSE-TRUE positions or with FALSE-NEUTRAL-TRUE positions, as in Figure 9, would yield essentially identical results.

Identical results will also obtain when rule r_3 is not asserted with absolute certainty ($m_3 = 1$) but is subject to exceptions, i.e., $m_3 = 1 - \varepsilon_3 < 1$. The Bayesian analysis yields:

$$P(f \mid p, b) \leq \frac{\varepsilon_1}{1 - \varepsilon_3} \tag{17}$$

meaning that, as long as ε_3 remains small, bird penguins have very small chance of flying; regardless of how many birds can't fly (ε_2). The D-S analysis, on the other hand, would still yield the paradoxical result

$$Bel(f) \cong \frac{\varepsilon_1}{\varepsilon_1 + \varepsilon_2}, \tag{18}$$

meaning that if non-flying birds are very rare, i.e., $\varepsilon_2 \approx 0$, then bird penguins have very high chance of flying.

The clash with intuition does not revolve around the exact numerical value of $Bel(f)$ but, rather, around the unacceptable phenomenon that rule r_3, stating that penguins are a subclass of birds, plays no role in the D-S analysis. Once we know that Tweety is both a penguin and a bird, *Bel*(Tweety flies) is only a function of m_1 and m_2, regardless of how penguins and birds are related. The same results would obtain had Tweety been an elephant bird, a broiled bird or a penguin with massive wings.

In common discourse, class properties are expected to be overridden by properties of more specific subclasses. Yet, the D-S analysis yields a substantial increase in the belief that penguins can fly if one adds the superfluous information that penguins are birds and birds normally fly.

This paradoxical result stems from the D-S interpretation of if-then rules as randomized logical formula of the material implication type, as opposed to statements of conditional probabilities. While in classical logic the three rules in our example will yield an unforgivable contradiction, the uncertainties attached to these rules, together with Dempster's normalization, now render them manageable. However, they are managed in the wrong way, because the material-implication interpretation of if-then type rules is so fundamentally wrong, that it cannot be rectified by allowing exceptions in the form of randomization. The source of the problem lies in the property of transitivity, $(a \rightarrow b, b \rightarrow c) \Rightarrow a \rightarrow c$, which is inherent to the material-implication interpretation. There are occasions where rule transitivity must be totally suppressed, not merely weakened, or else strange results will surface. One such occasion occurs in property inheritance, where subclass specificity should override superclass properties. Randomization, in this case, only weakens the flow of inference through rules in tandem but does not bring it to a dead halt, as it should.

This phenomenon is not unique to taxonomic property inheritance but is pervasive in common everyday reasoning. For example, consider the rules:

r_1: If I am sick, then I can't answer the door (m_1)
r_2: If I am home, then I can answer the door (m_2)
r_3: If I am sick, then I stay home (m_3 1)

Rule r_3 tells us that exceptions to rule r_2, due to sickness, were already taken into account by the measure m_2 and, moreover, exceptions to rule r_1, including those emanating from staying home, were already summarized in the measure m_1. Thus, given that I am sick, the conclusion is that I can't answer the door with confidence m_1; given that I am both sick and at home, the same conclusion applies and the same confidence as well.

The phenomena is further amplified in abductive tasks, where rule transitivity may lead to truly strange results. Consider the rules

r_4: If the ground is wet, then it rained last night (m_4)

r_5: If the sprinkler was on, then the ground is wet ($m_5 \approx 1$)

If we find that the ground is wet, rule r_4 tells us that $Bel(\text{rain}) = m_4$. Now, suppose we further observe that the sprinkler was on. Instead of decreasing (by virtue of explaining away the wet ground), $Bel(\text{rain})$ will remain the same. More seriously, suppose we first observe the sprinkler. Rule r_5 would correctly predict that the ground ought to get wet and, without even inspecting the ground, r_4 will conclude that it rained last night, with $Bel(\text{rain}) = m_4 m_5$.

These difficulties have haunted non-monotonic logic for years (see [Pearl, 1987d] and [Ginsberg, 1988] for more detail) and will be transferred over to the D-S analysis as long as if-then rule are treated as material implications, however weakened by randomization. They can be circumvented by two methods, none of which is truly satisfactory. One method requires the rule author to explicitly state the exceptions (or assumptions) underlying each rule. For example rule r_2 will be phrased: r'_2: If I am at home, I can answer the door, unless I am sick, or asleep or under, a gun threat ... in which case I will not be able to answer the door. This method would work well under D-S analysis, however, the enormous number of exceptions to each rule prevents it from being practical. The second method, used for example, in inheritance systems [Touretzky, 1986] [Etherington, 1987] is to recognize that rule sets such as $\{r_1, r_2, r_3\}$ lead to multiple extensions, and use extra logical criteria to decide which of the extensions should be preferred. For example, our rule set $\{r_1, r_2, r_3\}$ together with the facts: "I am sick" and "I am home" would give rise to two extensions:

$$E_1 = \{\text{"I am sick", "I am home", "I cannot answer the door"}\}$$

$$E_2 = \{\text{"I am sick", "I am home", "I can answer the door"}\},$$

depending on which rule, r_1 or r_2, is activated first. E_1 is preferred to E_2 because the rule r_1 *preempts* the path (r_3, r_2) [Touretzky, 1986]. Ginsberg [1984] has proposed to handle preemptions by special-purpose meta rules which "never apply a rule to a set when there is a corresponding rule which can be applied to a subset". In [Geffner & Pearl, 1987] we show that this notion of preemption can be integrated directly into the logic (thus preventing the generation of multiple extensions) if only the rules are given their proper interpretation, namely, conditional probability statements with probabilities arbitrarily close to one, short of actually being one.

2 TRUTH MAINTENANCE SYSTEMS

Introduction

Truth Maintenance Systems [Doyle, 1979] provide means of keeping track of beliefs and their justifications developed during an inference process. Since our reasoning habits are built largely on default assumptions and educated guesses, some of these assumptions may have to be retracted in the light of new information. Moreover, all conclusions that were derived from these assumption would have to be retracted as well, unless they can be supported by new arguments. To manage this retraction process, a truth-maintenance system must maintain a dependency record with each inferred fact indicating its justification in terms of both the presence and absence of information.

This retraction job is similar to the belief revision task of [Pearl, 1987b] with the exceptions that the interrelationships between propositions is not probabilistic but logical; a conclusion is either justified by a set of facts or unjustified, with no grey levels of support strength. As a result, the abductive task of finding the "most likely explanation" cannot be accomplished by numerical means, as in [Pearl, 1987b], but, rather, relies on explicit diagnostic rules suggesting hypotheses that account for observations. Second, uncertainty in these systems is represented, not by numerical degrees of belief but rather, by symbolic annotations called *assumptions,* which identify, by names, those uncertain facts which, if true, would justify our belief in a given proposition (we referred to such entities as *contingencies* in [Pearl, 1987a]. In other words, instead a numerical quantity, the truth value of a proposition is represented by a Boolean expression that identifies the assumptions needed for believing in that proposition.

Although lacking numerical criteria for deciding among opposing hypotheses, these systems are nevertheless popular because they are compatible with the symbolic style of logical inferences, and because, unlike monotonic logic, they do not take advantage of the rule of detachment which permits one to dispose of the derivation once the conclusion is established. On the contrary, by recording the history of derivations, these systems can retrace the source of beliefs, a feature necessary for generating explanations and for resolving contradictions.

The reason for introducing Truth Maintenance Systems in this report is two fold. First, these systems can be viewed as symbolic engines for computing D-S belief functions. Second, like the D-S theory, inferences in truth-maintenance systems are generated by hard, categorical compatibility relations and, hence, the notion of evidential support is based on the notion of *provability;* the semantic clash of the proceeding section will have to be dealt with separately.

2.1 Naming the Assumptions

Truth-Maintenance Systems (TMS) use rules as their elementary units of knowledge and, similar to our treatment in Section 1.5, conclusions are drawn by piecing together rules to form proofs. Likewise, rules may have exceptions that may cause the expected conclusion of the proof to clash with observed facts or with other deductions. However, whereas the exceptions and/or assumptions in Section 1.5 were summarized numerically, using the rule weight m, the TMS approach maintains an explicit list of the main assumptions and exceptions that are involved in each rule. For example,

$$\text{TURN–KEY} \rightarrow \text{START–ENGINE} \quad (m) \tag{19}$$

will be written

$$\text{TURN–KEY \& [GOOD–STARTER, BATTERY–NOT–DEAD, etc.]} \rightarrow \text{START–ENGINE} \tag{20}$$

where the terms in the square brackets are the assumptions behind the rule in (19). Thus, each rule in a TMS consists of two types of antecedents called *justifications* (e.g. TURN-KEY) and *assumptions* (e.g. BATTERY-NOT-DEAD). The difference between the two is only that assumptions are presumed to be true under normal conditions while justifications may be true or false, depending on whether they can be proven or refuted from observed facts, called *premises* in TMS's terminology.

This distinction introduces a bias that would permit us to temporarily ignore the assumptions altogether; proofs constructed via the justification part of the rules are considered legitimate, and their conclusions are tentatively adopted as firmly held beliefs. Indeed, in the original TMS proposed by [Doyle, 1979], assumptions are manipulated only when an observation is obtained that conflicts with proofs based on previously held assumptions, at which time, the TMS produces an alternative set of assumptions consistent with the observation. For example, if in addition to rule (20), we also have the facts (premises) that the key is turned and the engine does not start, then the TMS will issue a new assumption set {BAD-STARTER} or, alternatively, {BATTERY-DEAD}. rendering the observation consistent with the premises.

In a subsequent version of TMS, called ATMS (Assumption-Based TMS, [de Kleer, 1986]) the system maintains, not just one, but a whole list of assumption sets (called *environments*), any of which, if realized, could support our currently held beliefs. Each assumption set is *minimal*, or non-redundant, in the sense that no assumption can be moved from the set without destroying its support of the current belief. Minimality, parallels the notion of most-probable-explanation (see [Pearl, 1987b]) if exceptions (i.e., negation of assumptions) are assumed to be rare, equiprobable and independent events. Additionally, similar to the cautious attitude displayed by the belief revision scheme of [Pearl, 1987b], the system also maintains a list of assumption sets to support the negations of our current beliefs, in preparation for conflicting observations sometime in the future. In our example above, the list of assumption sets {{BAD-STARTER} , {BATTERY-DEAD}} will label the proposition ¬ (ENGINE-START), even prior to such an unfortunate observation. Each assumption set in this list is sufficient to support a proof of $\neg(ENGINE-START)$.

Example. To see the relation between the ATMS mode of reasoning and the one analyzed in Section 1.4 let us illustrate the former using the example of Figure 11, with a fifth rule added, $r_5 : s \rightarrow u$. Instead of switches and rule weights as in Figure 12, exceptions are formulated in terms of propositional symbols $A_1, A_2, \cdots$ (connoting assumptions) that are added to the antecedents of the corresponding rules. Altogether, we have the following 5-rule system as shown in Figure 14:

$r_1 : p \ \& \ q \ \& \ [A_1] \rightarrow s$
$r_2 : s \ \& \ t \ \& \ [A_2] \rightarrow c$
$r_3 : p \ \& \ [A_3] \rightarrow u$
$r_4 : u \ \& \ [A_4] \rightarrow c$
$r_5 : s \ \& \ [A_5] \rightarrow u$

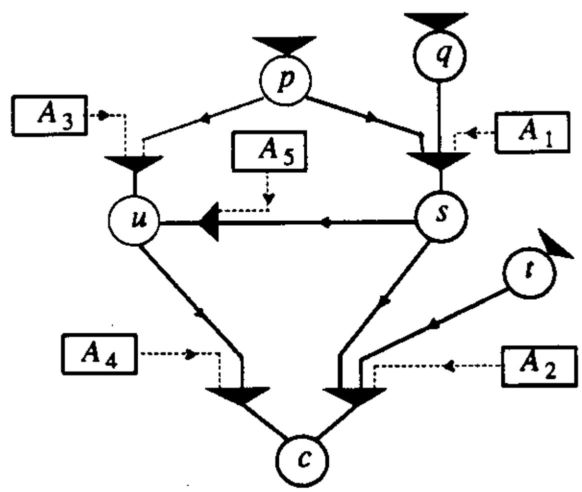


Figure 14. A graphical representation of a five-rule ATMS with their associated assumptions, $A_1, \cdots A_5$

Note the striking similarity between Figure 14 and Figure 12; the switches and their weights $m_1, m_2, \cdots$ are merely replaced by the proposition labels $A_1, A_2, \cdots$ that represent the assumptions. Accordingly, each assumption A_i can be viewed as a valve which controls the position of a switch and asserts "the switch corresponding to rule r_i is ON." However, unlike the switches in the D-S formalism, each A_i can be a complex Boolean formula of propositions which, in themselves, can be consequents of other rules.

Under normal operation, the ATMS is given a set of premises, e.g., $\{p, q, t\}$, and maintains for each proposition in the system a list of minimal sets of assumptions under which that proposition can be proven. Such a list is called a *label*. In our example, the label of u would be $L(u) = \{\{A_3\}, \{A_1, A_5\}\}$. The set $\{A_3\}$, for instance, indicates that assumption A_3 (together with the premises) is sufficient to activate rule r_3, which constitutes a proof of u. Proposition c, however, will have a more extensive label:

$$L(c) = \left\{ \{A_3, A_4\}, \{A_1, A_5, A_4\}, \{A_1, A_2\} \right\}, \tag{21}$$

where each set corresponds to the assumptions that enable one distinct proof-graph for c.

A ATMS keeps separate labels for the negations of the propositions in the system. In our system these labels will be empty because there is no rule with either $\neg s$, $\neg u$, or $\neg c$ as a consequent and, so, there is no way to prove any of these propositions. However, had the system been augmented with such a rule, say,

$$r_6 = v \rightarrow \neg c,$$

we would then label $\neg c$ with:

$$L(\neg c) = \left\{ \{\neg A_1, \neg A_3\}, \{\neg A_3, \neg A_5, \neg A_2\}, \{\neg A_4, \neg A_2\}, \{\neg A_1, \neg A_4\} \right\} \tag{22}$$

These sets come from negating the disjunction of the sets in $L(c)$, and correspond, as the reader might have expected, to the four (minimal) cutsets of the network in Figure 14.

Labels represent the *contexts* or *environments* under which propositions can be safely believed. The purpose of maintaining these labels explicitly is to propagate them quickly from one proposition to another and to be able to retract the proper set of assumptions when a contradiction arises. For example, in case $\neg c$ is asserted as a premise, assumptions which underlie c are no longer consistent with the premise set and at least one assumption must be retracted from each of the three sets of $L(c)$. This is accomplished by keeping a global stack of "nogood" assumption sets to be removed from any label in which they appear. In our example, the "nogood" assumption sets resulting from observing $\neg c$ would simply be the sets contained in $L(c)$. These, as well as their supersets, should be removed from the label of every proposition in the system, in particular those derived prior to the observation of $\neg c$.

In summary, the ATMS can be viewed as a symbolic algebra system that produces a Boolean expression $L(c)$ for every given proposition c. $L(c)$ contains a list of nonredundant sets of assumptions called environments, each of which is sufficient to support a proof of c, given the available premises. In logical terms, $L(c)$ enumerates the prime implicants entailing c [Reiter and de Kleer, 1987]. In graphical terms, assuming that each rule contains a single justification and a single assumption, $L(c)$ enumerates all the paths (proofs) leading from the premises to c.

2.2 Uncertainty Management in ATMS

From a purely logical viewpoint, the label $L(c)$ that the ATMS attaches to a proposition yields only three possible truth values for c: believed, disbelieved and unknown. If any environment in $L(c)$ is believed, then c is believed as well, if we believe any environment in $L(\neg c)$, then c is disbelieved. Otherwise, if we cannot confirm $L(c)$ nor $L(\neg c)$, then c is unknown.

These three-value logic lacks the facility of rating the degree of uncertainty attributable to unknown propositions and, thus, may lead to a stalemate whenever a decision is to be made whose outcome depends critically on the truth of these propositions. However, there are three very useful functions that one can perform, even within this limitation:

1) *Produce Explanation*. Once a proposition c is believed, the ATMS can retrace back the justification paths and identify the argument (proof) justifying that belief as well as the assumptions upon which it is founded. This is similar to tracing back the π and λ messages in probabilistic belief networks and is facilitated by recording the justification part of the rules next to each datum which amounts, essentially, to keeping a network structure in memory.

2) *Managing Conflicts* - Contradictions between expectations and reality are viewed as a signal for modifying the currently held set of assumptions. New sets of assumptions are generated automatically which are consistent and maximal (i.e., contains a minimal set of exceptions)

3) *Guiding the Acquisition of New Information* - If a certain proposition is in an "unknown" state, then the label $L(c)$ provides clues as to what information is required to render it known, i.e., believed or disbelieved. For example, if a confirmation of assumption A is all that is missing from one set in $L(\neg c)$ while the confirmation of $\neg A$ is missing from some set in $L(c)$, then a test leading to the confirmation or denial of A should be devised.

Introducing Numerical Uncertainties. The three features above do not fully replace the facilities provided by numerical measures of uncertainty. For example, a pending decision may depend crucially on the likelihood that a given "unknown" proposition c will turn out true, and this, in turn, depends on how certain we are in the assumption sets of $L(c)$ compared with those of $L(\neg c)$. Likewise, there could be several diagnostic tests that one can perform, each having the potential of confirming or denying a crucial proposition c, yet some tests may be judged more likely to yield that determination than others.

The information needed for calculating numerical measures of certainty can easily be introduced within the ATMS setting. If we have the knowledge to assess the relative likelihood of the various assumptions in the system, then the ATMS provides a symbolic facility for translating this knowledge into a certainty measure for any conclusion of interest. The semantics of this certainty measure will be identical to that of belief functions in the D-S formalism, i.e., the probability of establishing a proof for the conclusion.

Since $L(c)$ constitutes a Boolean expression whose truth signifies the existence of a proof for c, $Bel(c)$ can be obtained by simply computing the probability of $L(c)$, i.e.,

$$Bel(c) = P[L(c)]. \tag{23}$$

Moreover, since the atoms which make up $L(c)$ are all assumption-type propositions, $P[L(c)]$ can be computed from the probabilities assigned to the assumptions. In particular, if one assumes that assumptions are independent of each other, the computation of $P[L(c)]$ can be done symbolically and amounts to the same computation we have conducted for belief functions in rule-based systems (see Section 1.4).

To illustrate the computation, consider Figure 14, and let p_i stand for the probability that assumption A_i is true. To calculate $Bel(c)$, the ATMS provides the label $L(c)$ from (21) and we have:

$$Bel(c) = P[L(c)] = P[A_3A_4 \vee A_1A_4A_5 \vee A_1A_2] \tag{24}$$

Probability calculus permits us to calculate the probability of a disjunction or conjunction of any two expression (in terms of the probabilities of the individual expressions) if the expressions are either mutually exclusive or independent. Unfortunately the subexpressions in (24) are neither. We therefore substitute

$$A_3A_4 = A_3A_4(A_1 \vee \neg A_1)$$

and obtain

$$Bel(c) = P[A_3A_4\neg A_1 \vee A_1(A_2 \vee A_4(A_3 \vee A_5))] \tag{25}$$

The first term in the square bracket is disjoint of the second, while the second is in series-parallel format. This yields

$$\begin{aligned} Bel(c) &= p_3p_4(1-p_1) + p_1[p_2 \| p_4(p_3 \| p_5)] \\ &= p_3p_4(1-p_1) + p_1[p_2 + p_4(p_3 \| p_5) - p_2p_4(p_3 \| p_5)] \\ &= p_3p_4(1-p_1) + p_1[p_2 + p_4(1-p_2)(p_3 + p_5 - p_3p_5)] \end{aligned} \tag{26}$$

This expression is identical to the one in Eq.(12), identifying p_i with m_i. Mechanical procedures for computing the probability of an arbitrary Boolean expression of independent propositions have been reported in the literature on network reliability (e.g., [Grnarov et al, 1979]) and can be easily applied to the labels returned by the ATMS.

2.3. Incidence Calculus

Incidence calculus [Bundy, 1985] is a method of computing belief functions by logical sampling, similar in spirit to the method of stochastic simulation, [Pearl, 1987e]. A probabilistic knowledge base is used to generate random samples of truth values for a select set of propositions and these are presented as input facts, or axioms, to a theorem prover. Different sets of facts give rise to different theorems and $Bel(c)$ is given by that fraction of the time that c can be proven.

Consider again the rule network of Figure 14, where it is required to compute $Bel(c)$, given the probabilities on the assumptions $A_1, \cdots A_5$. Instead of computing a symbolic expression for $Bel(c)$ in terms of $p_1, \cdots p_5$, as in the preceding subsection, one can actually *simulate* the real-time behavior of the network under a random assignment of truth values to the assumptions, then count the frequency at which c is proven true. The truth value of each assumption A_i will be represented by a random bit string of ones and zeros, in which the frequency of ones is p_i. In every time step, each assumption selects the next bit from its bit string, sets its value accordingly, and a theorem-prover attempts to prove the truth of c. The frequency at which a proof is established is equal to $Bel(c)$.

This scheme is a physical embodiment of the random switch model described in Section 1.3. The random position of each switch is replaced by the random bit string assigned to each proposition (e.g., assumption) whose degree of certainty we wish to assert. The theorem prover can be general purpose (e.g., First Order Logic), not limited to propositional constraint networks. The scheme is not limited to simulating independent switches; dependencies can be simulated by having the bit strings generated by a complete probabilistic model (e.g., a causal network) in which these dependencies are encoded.

3 NILSSON'S PROBABILISTIC LOGIC

While Bayesian theory requires the specification of a complete probabilistic model, and the D-S theory sidesteps the missing specifications by compromising its inferences, probabilistic logic [Nilsson, 1986] considers the space of all models consistent with the specifications available, and computes bounds, instead of point values, for the probabilities required. Probabilistic logic (PL) addresses the following problem. Suppose we are given a collections S of logical sentences, some representing facts (e.g., "Socrates is a dog") and some representing generic laws (e.g., "all dogs bark"), and suppose someone attaches probability measures to some of the sentences, representing the degree of belief in their truth. Our task is to deduce the probability of other sentences in the language whose probabilities were not specified explicitly (e.g., "Socrates barks").

In some way this problem resembles the evidential reasoning tasks of [Pearl, 1986]. There, too, we started with probabilistic assessments on a small set of sentences (i.e., those used in the construction of the Bayes network) and we were able to deduce the probability of every query phrased in propositional form (see [Pearl, 1986]). However, in Bayes networks, the assignment of probabilities was done in a principled way, guaranteeing consistency and completeness; merely assigning probabilities to a set of logical sentences does not, in general, define a complete probabilistic model, even when the assignment is consistent. The logical relationships between the sentences in S will, in general, admit a high number of truth value assignments, called *extensions,* and, unless one assigns a probability rating to each such extension, the model remains, in a probabilistic sense, incomplete.

To illustrate this point let us examine a simple example involving the following three sentences:

$S_1 = p$
$S_2 = p \supset q$

$S_3 = q$

If we regard these sentences as binary variables that may take on true-false values, then out of the $2^3 = 8$ such value combinations, the four consistent ones are extensions, and these are given by the rows of the following table:

Extensions	S_1: p	S_2: $p \supset q$	$S_3 = q$
W_1	true	true	true
W_2	true	false	false
W_3	false	true	true
W_4	false	true	false

Assigning probability measures to any two sentences does not fully specify a model and does not yield a unique probability measure for the third sentence. The best way to see it, using the Bayesian style of representing dependencies, is to view S_1 and S_3 as parent variables of S_2. Since S_2 is a Boolean function of S_1 and S_3, its value is completely determined by the values assigned to the latter two. This means that a complete probabilistic model can be defined by assigning arbitrary weights to each of the four possible truth values of (S_1, S_3), just making sure that they sum to unity. This requires a specification of three parameters, e.g., $P(p, q), P(p, \neg q)$ and $P(\neg p, q)$. If we only specify $P(p)$ and $P(q)$, the model remains underspecified.

More interesting difficulties surface when someone assigns probability measures to S_1 and S_2 and seeks to deduce the probability of S_3. This is a typical occurrence in rule-based systems, where S_2 is (falsely) taken to be the logical representation of the English sentence "if p then q". Since S_3 is not a Boolean function of S_1 and S_2 (for $S_1 = \text{false}$ and $S_2 = \text{true}$, S_3 can attain either a true or a false value), even specifying the joint probability on the pair (S_1, S_2) will not suffice; one still need to assess $P(S_3 = \text{true} \mid S_1 = \text{false}, S_2 = \text{true})$ before the model is completely specified. Moreover, since S_2 and S_1 cannot both be false, one cannot specify the joint probability on (S_1, S_2) by an arbitrary selection of three parameters; caution must be exercised to ensure that the selection is consistent with the requirement $P(S_1 = \text{false}) + P(S_2 = \text{false}) \leq 1$.

Violations of such consistency requirements may be very troublesome in rule-based systems [Duda et al, 1976]. $P(S_2 = \text{false})$ is often perceived as measuring one's doubt in the validity of some generic rule, (e.g., exceptions to the rule S_2: "dogs bark"), and hence, it is kept constant. At the same time, since $P(S_1 = \text{false})$ is allowed to vary, depending on the amount of support that S_1 receives from other rules, the consistency requirement above will occasionally be violated. It is also not uncommon to find an expert providing the assessments: $P(S_2 = \text{false}) = .1$, reflecting a rule $p \supset q$ with 10% exceptions, together with $P(S_1 = \text{true}) = .001$, reflecting some rare event p. The two assessments are clearly inconsistent because S_2 must be true whenever S_1 is false, i.e., at least 99.9% of the time. Aside from the obvious conclusion that the material implication S_2: $p \supset q$ is the wrong interpretation of the conditional sentence "if p then q", the example illustrates the ease of introducing inconsistencies while assigning probability measures to individual logical sentences.

Probabilistic logic provides a formal way of testing when probability assignments are consistent with each other. Once we have a criterion for testing consistency, we also possess a device for determining *bounds* on the permissible probabilities that any given sentence may assume, given the probabilities assigned to other sentences. Such bounds are said to be *probabilistically entailed* by the other assignments.

Conceptually, probabilistic consistency and entailment have simple semantics; the probability $P(S_i)$ associated with a sentence S_i in $\mathbf{S}$ constitutes a *constraint* on any probability distribution P that can be assigned to the extensions of $\mathbf{S}$. Since those constraints are not sufficient to determine one unique distribution, we seek a description of the set $\mathbf{P}$ of all probability distributions that comply with the given constraints. Each distribution $P \in \mathbf{P}$ defines a probability value $P(\phi)$ for any arbitrary sentence ϕ; $P(\phi)$ is simply the sum total of the probabilities of those extensions in which ϕ is true. Given $\mathbf{P}$, we can determine the range of admissible $P(\phi)$ values, letting P span the entire set $\mathbf{P}$.

Let $\mathbf{W} = \{W_1, W_2 \cdots\}$ stand for the set of all extensions of $\mathbf{S}$, let $P(W_i)$ be the weight assigned to W_i by some distribution $P \in \mathbf{P}$, and let w_{ij} be a binary variable taking the value 1 if sentence S_j is in extension W_i, zero otherwise. The constraints that the probabilities $P(S_j)$ impose on the distribution P are given in the form of a linear equation

$$P(S_j) = \sum_{W_i \supset S_j} P(W_i) = \sum_i w_{ij} \, P(W_i) \tag{27}$$

In our 3-sentence example above, we had four feasible extensions, each corresponding to a row in the truth table, hence w_{ij} are given by the (i, j)th entry of the table. If we demand that our three sentences be given the probability vector $\pi = (P(S_1), P(S_2), P(S_3))$, then the three equations in (27) constitute constraints on the distributions $P = (P(W_1), P(W_2), P(W_3), P(W_4))$ that could possibly be assigned to our extensions. Given any P satisfying (27) and an arbitrary sentence ϕ, $P(\phi)$ can be computed simply by adding the weights $P(W_j)$ over the extensions where ϕ holds.

Eq.(27) also constitutes a description of the set of consistent probabilities, π, that can be assigned to the sentences in $\mathbf{S}$. Consistency means that Eq.(27) should have at least one solution for $P(W_j)$. Since it is a linear mapping between $P(W_j)$ and $P(S_i)$, Eq. (27) maps extreme distributions $P(W_j)$ to extreme assignments $P(S_i)$, and the convex hull of these extreme assignments defines the set of consistent assignments π. Moreover, since each extreme distribution on $\mathbf{W}$ selects one row from the w_{ij} table, we conclude that the set of consistent assignments π to the sentences in $\mathbf{S}$ is bounded by the convex hull generated by the extensions of $\mathbf{S}$.

To illustrate how probability bounds are entailed, assume that we start with a single sentence S_2 to which we attribute the probability $P(S_2) = \pi_2$ and we wish to find what values are permitted for either $P(S_1)$ or $P(S_3)$ (For more elaborate examples, together with geometrical representations of the convex hulls, see [Nilsson, 1986]).

There are three feasible extensions permitted by S_1 and S_2, given by the table below:

Extensions	S_1: p	S_2: $p \supset q$
W_1	0	1
W_2	1	0
W_3	1	1

The three extreme distributions on these extensions, (1, 0, 0) (0, 1, 0) and (0, 0, 1), map into the following values for the pair $[P(S_1), P(S_2)]$: (0, 1) (1, 0) and (1, 1). Each extension defines an extreme point in the space $[P(S_1), P(S_2)]$, and the convex hull generated by these points defines the permissible region for the pair of probability assignments $[P(S_1), P(S_2)]$ (see Figure 15). Clearly, if $P(S_2)$ is set to π_2, then $P(S_1)$ is bounded by the inequality

$$1 - \pi_2 \leq P(S_1) \leq 1 \tag{28}$$

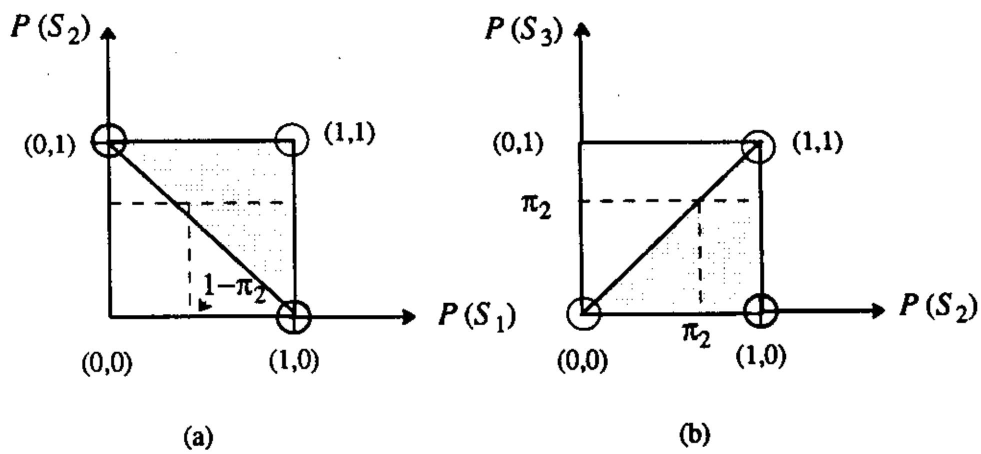


Figure 15. The convex hull formed by the extensions of S determines the region of consistence probability assignments to the sentences in S

The bounds on $P(S_3)$ can be found in the the same way. The three legitimate extensions are shown in Figure 15(b), which yields

$$\pi_2 \leq P(S_3) \leq 1 \tag{29}$$

Summary: Probabilistic entailment is a method of dealing with partially specified probabilistic models, where the specification is in the form of probability assignments to a select set of logical sentences. Any such assignment defines a region of permissible complete models and, by describing the boundaries of this region, one can deduce bounds on the probabilities of new sentences.

The way probabilistic logic (PL) deals with partially specified models is opposite to that of D-S theory. Both methods accept specifications in the form of logical sentences and a probability assignment to a subset of these sentences. However, whereas PL treats probabilistic models as the object-level theories and logical relations as meta-level constraints, the D-S theory reverses these role; logical constraints serve as the object-level theory, within which deduction takes place, and probabilistic information serves to govern the inputs that the object-level theory obtains. This is described schematically in Figure 16

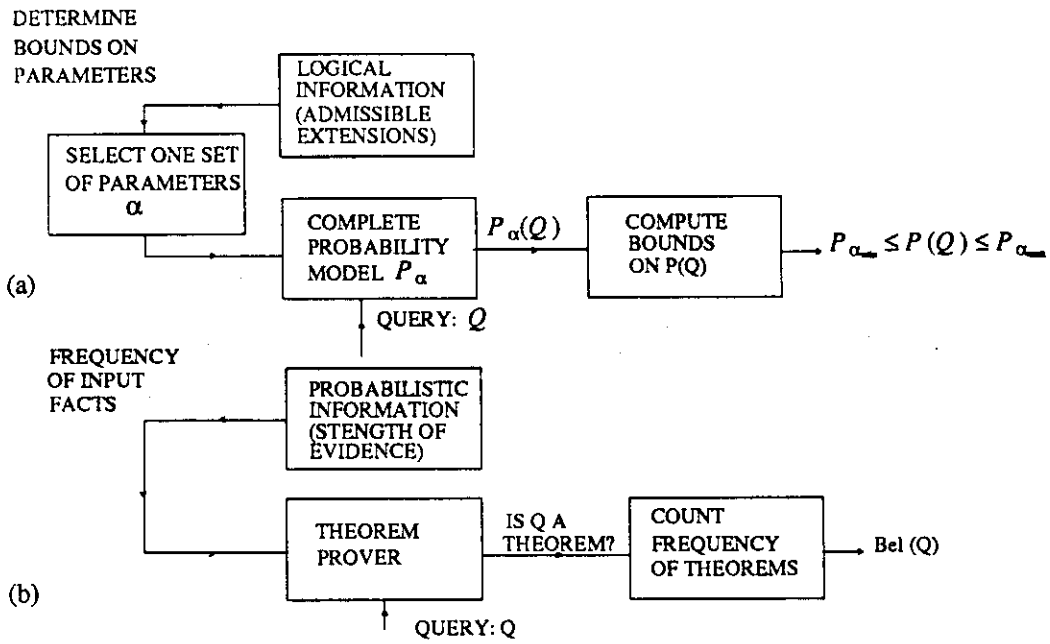


Figure 16. Meta-Level and Object-Level Theories in (a) Probabilistic Logic and (b) Dempster-Shafer's Formalism

PL is shown in Figure 16(a) to be using logical information for controlling (or bounding) the set of parameters α which characterizes a complete probability model P_α. Object level deductions are done in the probabilistic domain; computing $P_\alpha(Q)$ for an arbitrary query sentence Q, then deducing bounds on $P(Q)$ by considering the logical bounderies of α. The D-S scheme is depicted in Figure 16(b). It uses probabilistic information regarding the strength of evidence to route certain facts (those supported by the evidence) as inputs to a logical inference engine, i.e., a theorem prover. The frequency at which various input facts are presented to the engine is controlled by the strength of evidence supporting these facts. For every set of input facts, the inference engine verifies whether Q is a theorem, and $Bel(Q)$ is computed as the frequency with which this theorem is proved by the engine. This description of the D-S approach best matches its Incidence Calculus implementation (Section 2.3).

References

[Baldwin, 1987] Baldwin, J.F., "Evidential Support Logic Programming," *Fuzzy Sets and Systems*, 24, 1987, pp. 1-26.

[Bundy, 1985] Bundy, A., "Incidence Calculus: A Mechanism for Probabilistic Reasoning", *Journal of Automated Reasoning, 1* (1985), pp. 263-283.

[de Kleer, 1986] de Kleer, J., "An Assumption-Based Truth Maintenance System", *Artificial Intelligence,* Vol. 29, 1986, pp. 241-288.

Dechter, R., & Pearl, J., "Tree-Clustering Schemes for Constraint-Processing," UCLA Cognitive Systems Laboratory, *Technical Report 870054 (R-92), June 1987.*

[Doyle, 1979] Doyle, J., "A Truth Maintenance System," *Artificial Intelligence,* 12:3, 1979.

[Duda, Hart & Nilsson, 1976] Duda, R.O., P.E. Hart, and N.J. Nilsson, "Subjective Bayesian Methods for Rule-Based Inference System," *Proceedings, National Computer Conference (AFIPS Conference Proceedings),* 45, 1976, pp. 1075-1082.

[Geffner & Pearl, 1987] Geffner, H., & Pearl, J., "A Sound Framework for Reasoning with Defaults," UCLA Cognitive Systems Laboratory, *Technical Report (R-94),* October 1987.

[Ginsberg, 1984] Ginsberg, M. L., "Non-monotonic Reasoning Using Dempter's Rule", *Proc. AAAI-84 Conference,* Austin, Texas, 1984, pp. 126-129.

[Ginsberg, 1987] Ginsberg, M., (ed). *Readings in Non-monotonic Reasoning,"* Morgan and Kaufmann, Los Altos, CA. 1987.

[Gmarov, et al, 1979] Gmarov, A., Kleinrock, L., and Gerla, M., "A New Algorithm for Network Reliability Computation," *Comp. Networking Symposium: Proceedings,* Dec. 1979, Gaithersburg, MA., pp. 17-20.

[Hajek, 1987] Hajek, P., "Logic and Plausible Inference in Expert Systems", *Proc. of AI-Workshop on Inductive Reasoning,* April, 1987, RISO, Denmark.

[Heckerman, 1986] Heckerman, D., "Probabilistic Interpretations for MYCIN Certainty Factors" *Uncertainty in Artificial Intelligence,* Kanal, L. N. and Lemmer, J. F. (eds.), North Holland, 1986, pp. 167-196.

[Hummel & Manevitz, 1987] Hummel, R., and Manevitz, L.M., "Combining Bodies of Dependent Information," *Proceedings,* IJCAI-87, Milan, Italy, pp. 1015-1017.

[Nilsson, 1986] Nilsson, N., "Probabilistic Logic," *Artificial Intelligence,* 28:1, 1986, pp. 71-87.

[Pearl, 1982] Pearl, J., "Reverend Bayes on Inference Engines: a Distributed Hierarchical Approach," *Proceedings,* AAAI National Conference on AI, Pittsburgh, PA. August 1982, pp: 133-136.

[Pearl, 1986] Pearl, J., "Fusion, Propagation and Structuring in Belief Networks, *Artificial Intelligence,* Vol. 29(3), September 1986, pp. 241-288.

[Pearl, 1987a] Pearl, J., "Do we Need Higher-Order Probabilities and, If so, What do they Mean"? *Proceedings,* AAAI Workshop on Uncertainty in AI, Seattle, WA. July 1987, pp. 47-60.

[Pearl, 1987b] Pearl, J., "Distributed Revision of Composite Beliefs," *Artificial Intelligence, Vol. 33(2), 1987, pp.* 173-215.

[Pearl, 1987c] Pearl, J., "Bayes Decision Methods," *Encyclopedia of AI,* Wiley Interscience, New York, 1987, pp. 48-56.

[Pearl, 1987d] Pearl, J., "Embracing Causality in Formal Reasoning," *Proceedings,* AAAI Conference, Seattle, WA. July 1987, pp. 369-373.

[Pearl, 1987e] Pearl, J., "Evidential Reasoning Using Stochastic Simulation of Causal Models," *Artificial Intelligence,* Vol. 32(2), 1987, pp. 245-258.

[Reiter & de Kleer, 1987] Reiter, R. and de Kleer, J., "Foundations of Assumption-Based Truth Maintenance Systems", *Proc. AAA-87,* Seattle, WA, 1987, pp. 183-188.

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

[Shafer, 1987a] Shafer, G., "Belief Functions and Possibility Measures," in J. Bezdek (Ed.), *Analysis of Fuzzy Information,* Vol. 1: Mathematics & Logic, CRC Press, 1987, pp. 51-84.

[Shafer et al, 1987b] Shafer, G., Shenoy P. P., and Mellouli, K., "Propagating Belief Functions in Qualitative Markov Trees." The University of Kansas, Lawrence, School of Business, working paper no. 190, June 1987. To appear in International Journal of Approximate Reasoning.

[Spieglehalter, 1986] Spieglehalter, D., "Probabilistic Reasoning in Predictive Expert Systems", in Kanal, L.N. & Lemmer, J. (Eds.) *Uncertainty in Artificial Intelligence,* North-Holland, Amsterdam , 1986, pp. 47-68.

[Touretzky, 1986] Touretzky, D. S., *The Mathematics of Inheritance Systems.* Morgan Kaufmann, Los Altos, CA 1986.

[Tung & Kong, 1986] Tung, C. and Kong, A., "Multivariate Belief Functions and Graphical Models," *Research Report S-107,* Harvard University, Cambridge, MA, October 1986.

[Zadeh, 1981] Zadeh, L., "Possibility Theory and Soft Data Analysis," *Mathematical Frontier of the Social and Policy Sciences* (Cobb, L. and Thrall, R. M., eds.) Westview Press, Boulder, 1981, pp. 69-129.

[Zadeh, 1984] Zadeh, L., "Review of a Mathematical Theory of Evidence," by Glen Shafer, *The AI Magazine, 5, No. 3, 81, 1984.*

[Zadeh, 1986] Zadeh, L., "A Simple View of Dempster-Shafer Theory of Evidence and Its Implication for the Rule of Combination", *AI Magazine,* Summer 1986, pp. 85-90.