Journal of Automated Reasoning 1 (1985) 263–283. 0168–7433/85.15

Incidence Calculus: A Mechanism for Probabilistic Reasoning

ALAN BUNDY
Department of Artificial Intelligence, University of Edinburgh, Hope Park Square, Edinburgh EH8 9NW, Scotland

(Received: 10 June 1985)

Abstract. Mechanisms for the automation of uncertainty are required for expert systems. Sometimes these mechanisms need to obey the properties of probabilistic reasoning. We argue that a purely numeric mechanism, like those proposed so far, cannot provide a probabilistic logic with truth functional connectives. We propose an alternative mechanism, *Incidence Calculus*, which is based on a representation of uncertainty using sets of points, which might represent situations models or possible worlds. Incidence Calculus does provide a probabilistic logic with truth functional connectives.

Keywords: Incidence Calculus, probability, uncertainty, logic, expert systems, inference.

1. Introduction

Several mechanisms have been suggested for the automation of reasoning with uncertainty, e.g. Fuzzy Logic, [Zadeh 81], Shafer-Dempster theory, [Lowrance and Garvey 82], and the mechanisms proposed in MYCIN, [Shortliffe 76] and PROSPECTOR, [Duda *et al.* 78]. Most of these mechanisms involve assigning numbers to axioms (e.g. the facts and rules of an expert system), and assigning arithmetic functions to the rules of inference, so that new numbers can be calculated for the theorems that are derived from the axioms (e.g. the diagnoses of an expert system). We will call such mechanisms, *purely numeric*.

We will see that it is a desirable property of any uncertainty mechanism that the connectives should be *truth functional* with respect to the uncertainty measures. That is, it should be possible to calculate the uncertainty measures of a complex formula solely from the uncertainty measures of its subformulae. Note that the Propositional Logic connectives are truth functional with respect to the truth values true and false.

In some applications it is important to be able to assign *meaning* to the numbers so obtained, rather than use them merely to rank order some options. For instance, in medical diagnosis a user sometimes needs to be able to distinguish the situations where a diagnosis is very probably correct from the situation where it is just the best of an improbable batch. In the first case a surgeon might be prepared to perform a dangerous operation, in the second s/he might want to call for more tests and a re-diagnosis. In these situations we would like the numbers to represent probabilities.

Unfortunately, we will see that the logical connectives *cannot* be truth functional with respect to probabilities, or any other purely numeric uncertainty measure. We will propose a mechanism, called *Incidence Calculus*, in which the uncertainty measures are sets and for which the connectives *are* truth functional, and from which probabilities can easily be calculated.

2. Probabilistic Reasoning

What properties must probabilistic reasoning obey? These can be obtained from any testbook on mathematical probability, e.g. [Feller 68]. Adapting these properties to the needs of a logical calculus gives the set of equations given below.

We associate uncertainty values with *sentences*, i.e. formulas with no free variables, in some logic, e.g. Predicate Logic. The probability of a sentence being true is a real number, between 0 and 1. We will use upper case letters from the beginning of the alphabet to denote sentences, e.g. A, B, C, etc. and write $p(A)$ for the probability of A, etc. The probability of a true sentence is 1, and the probability of a false sentence is 0. That is,

$$p(t) = 1 \quad \text{and} \tag{i}$$

$$p(f) = 0, \tag{ii}$$

where t represents the universally true sentence, and f represents the universally false sentence. Values intermediate between 0 and 1 correspond to degrees of probability between these extremes.

The following equations assign arithmetic functions to the propositional connectives.

$$p(\sim A) = 1 - p(A) \tag{iii}$$

$$p(A \ \& \ B) = p(A).p(B) \quad \text{provided } A \text{ and } B \text{ are independent} \tag{iv}$$

$$p(A \vee B) = p(A) + p(B) - p(A \ \& \ B) \tag{v}$$

As we will see, the condition attached to equation (iv) is very important. It means that B is no more nor less likely given A than in general, and vice versa. It is this condition that will prevent purely numerical uncertainty mechanisms from having truth functional connectives, because the independence of two sentences cannot be coded in their numerical values.

3. The Limitations of a Purely Numeric Mechanism

If we ignore the independence condition on equation (iv) then we get a contradictory calculus. The contradiction can be derived by applying the rules, unconditionally, to two *dependent* sentences, for instance A and $\sim$A. Suppose, for the sake of definiteness, that $p(A) = 0.75$. Using the equations of the last section we have the following derivation.

$$p(\sim A) = 1 - p(A) = 0.25 \quad \text{(by (iii))}$$

$$p(A \,\&\, \sim A) \;=\; p(A).p(\sim A) \;=\; 0.1875 \qquad \text{(by (v))}$$

Similarly,

$$p(A \vee \sim A) \;=\; p(A) + p(\sim A) - p(A \,\&\, A) \qquad \text{(by (iv))}$$

$$= \; 0.8125$$

However,

$$p(A \,\&\, \sim A) \;=\; p(f) \;=\; 0 \qquad \text{(by (ii))}$$

and

$$p(A \vee \sim A) \;=\; p(t) \;=\; 1 \qquad \text{(by (i))}$$

but

$$0.1875 \neq 0 \quad \text{and} \quad 0.8125 \neq 1, \qquad \text{Contradiction!}$$

The contradiction cannot be avoided by modifying the arithmetic functions associated with the connectives. We would have to modify the equations (v) and (iii) so that they gave value 0 when calculating the probability of a sentence of the form $A \,\&\, \sim B$, where the probability of both A and B is 0.75. But not all such sentences are false. We can summarise this result by saying that a purely numeric probability mechanism cannot have truth functional connectives.

We need to design a mechanism which can take into account the degree of dependence of sentences when calculating their probabilities. In probability theory the *correlation*, $c(A, B)$, between two sentences, A and B, is used to measure their degree of dependence. It varies between the values -1 and 1. $c(A, B) = 0$ means A and B are independent. $c(A, A) = 1$ and $c(A, \sim A) = -1$. The correlation is so defined that:

$$p(A \,\&\, B) \;=\; p(A).p(B) + c(A, B).\sqrt{p(A).p(\sim A).p(B).p(\sim B)} \qquad \text{(vi)}$$

This equation★ provides an unconditional alternative to equation (iv), i.e. one without the condition that the conjuncts be independent. However, it does assume knowledge of both the probabilities *and* the correlations of the conjuncts. This assumption would not be unreasonable if we had a *correlation + probability calculus* with truth functional connectives, i.e. one which provided arithmetic functions for calculating the correlations of complex sentences from the correlations and probabilities of their subsentences, i.e. if we had functions which enabled the calculation of $c(A \,\&\, B, C)$, $c(A \vee B, C)$, $c(\sim A, C)$ and $c(A \rightarrow B, C)$ purely from $p(A)$, $p(B)$, $p(C)$, $c(A, B)$, $c(A, C)$ and $c(B, C)$.

We will see in Section 6 that it is not possible to provide a correlation + probability calculus with truth functional connectives. So in order to use sentence (vi) it is necessary to be provided with the correlations of all the infinitely many, possible pairs

★Define a random variable V_A on points such that $V_A(I) = 1$ if A is true in I and $V_A(I) = 0$ otherwise. Then $E(A^2) = E(A) = p(A)$, and (vi) follows from the definition in [Feller 68] p. 236.

of sentences. This means that using correlations is not a feasible solution to the problems of dependency in probability theory.

4. A Set Theoretic Mechanism

To design a mechanism which *can* deal with this problem we need to go back to the set-theoretic roots of probability theory. The probability of a sentence is based on a *sample space* of *points* [Feller 68]. Each point can be regarded as a situation, Tarskian interpretation, or possible world in which a sentence will be either true or false. The sample space is intended to be an exhaustive and disjoint set of points. It will be denoted w.

Non-trivial theories often have an infinite number of possible interpretations. For computational reasons we will usually require the sample space to be finite. Thus each point must sometimes stand for a, possibly infinite, equivalence class of interpretations.

Let $i(a)$ be the subset of w, containing all those points in which sentence A is true. We will call $i(A)$, the *incidence*★ of A. In [Feller 68] what we call an incidence is called an event: however, the term event is also used, ambiguously, to refer to sentences. The distinction between incidences and sentences is crucial in the discussion below, so we will not use the ambiguous term 'event'.

The dependence or independence of two sentences is coded in the amount of intersection between their incidences. The amount of intersection of two independent sentences is no more or less than you would expect from a random assignment of the elements of their incidences.

For the rest of this paper we will assume the underlying logic to be Predicate Logic. The following axioms of Incidence Calculus associate a set theoretic function with each connective, propositional constant and quantifier of Predicate (Propositional) Logic so that the incidence of a complex sentence can be calculated from the incidences of its subsentences. We call the resulting system *Predicate (Propositional) Incidence Logic*.

$$i(t) = w \qquad \text{(vii)}$$

$$i(f) = \{\ \} \qquad \text{(viii)}$$

$$i(\sim A) = w \backslash i(A) \qquad \text{(ix)}$$

$$i(A \,\&\, B) = i(A) \cap i(B) \qquad \text{(xi)}$$

$$i(A \vee B) = i(A) \cup i(B) \qquad \text{(xi)}$$

$$i(\forall X\ A(X)) \subseteq i(A(s)) \subseteq i(\exists X\ A(X)) \qquad \text{(xii)}$$

where s is a term containing no variables not bound in $A(X)$. Note that there is no independence condition on equations (vii) to (xi), and hence that all the connectives

★Incidence: degree, extent or frequency of occurrence; amount. – *Collins English Dictionary*.

truth functional with respect to incidences. The axioms for the universal and existential quantifiers only set upper and lower bounds, respectively, on their incidences, so the quantifiers are not truth functional with respect to incidences. This is not surprising since they are not truth functional with respect to truth values either.

If J is a point, let $\varrho(J)$ be the probability of J occurring. If I is a set of points, let $wp(I)$ be the sum of the probabilities of the points in I, i.e.

$$wp(I) \;=\; \sum_{J \in I} \varrho(J)$$

$wp(I)$ is called the *weighted probability* of I. For computational reasons we will usually use finite I, but the theory does not require I to be finite or even discrete.

Since the sample space is meant to be an exhaustive and disjoint set of points, we will require that:

$$wp(w) \;=\; 1$$

If A is a sentence, let $p(A)$ be the *probability* of A occurring. We define

$$p(A) \;=\; wp(i(A))$$

If A and B are sentences, let $p(A|B)$ be the *conditional probability* of A given B. We define:

$$p(A|B) \;=\; wp(i(A) \cap i(B))/wp(i(B))$$

From these definitions it is easy to derive the probability equations of Section 2.

$$p(t) \;=\; wp(i(t)) \;=\; wp(w) \;=\; 1$$

$$p(f) \;=\; wp(i(f)) \;=\; wp(\{\ \}) \;=\; 0$$

$$\begin{aligned} p(\sim A) &= wp(i(\sim A)) \;=\; wp(w \backslash i(A)) \\ &= [wp(w) - wp(i(A))] \\ &= 1 - wp(i(A)) \;=\; 1 - p(A) \end{aligned}$$

If A and B are independent then B is true just as frequently if A is true as it is in general. This can be expressed mathematically as:

$$wp(i(A \,\&\, B))/wp(i(A)) \;=\; wp(i(B)) \qquad \text{hence}$$

$$\begin{aligned} p(A \,\&\, B) &= wp(i(A \,\&\, B)) \\ &= [wp(i(A)).wp(i(B))] \\ &= p(A).p(B) \end{aligned}$$

If the weighted probability of points in which A is true is added to the weighted probability of points in which B is true then the weighted probability of points in which both are true are counted twice. This can be expressed mathematically as:

$$wp(i(A)) + wp(i(B)) = wp(i(A \vee B)) + wp(i(A \& B)) \quad \text{hence,}$$

$$\begin{aligned} p(A \vee B) &= wp(i(A \vee B)) \\ &= [wp(i(A)) + wp(i(B)) - wp(i(A \& B))] \\ &= p(A) + p(B) - p(A \& B) \end{aligned}$$

It follows that we can represent the probability of a sentence, A, implicitly, by associating its incidence, $i(A)$, with it. If we need to know the probability we can calculate $wp(i(A))$.

We can also derive the standard definition of conditional probability.

$$\begin{aligned} p(A|B) &= wp(i(A) \cap i(B))/wp(i(B)) \\ &= wp(i(A \& B))/wp(i(B)) \\ &= p(A \& B)/p(B) \end{aligned}$$

Hence,

$$p(A \& B) = p(B).p(A|B)$$

so conditional probability provides an alternative to correlation for calculating the probability of a conjunction from the probability of its conjuncts. However, a conditional probability + probability calculus with truth functional connectives is no more possible than a correlation + probability one, as we shall see in Section 6.

5. The Representation of Incidences

One of the advantages of a purely numeric mechanism for uncertainty is that computers are particularly efficient at representing and manipulating numbers. They are not so efficient at representing and manipulating sets.

However, Incidence Calculus can be implemented reasonably efficiently by representing the incidences of sentences as bit strings and manipulating them with logical operations. Each incidence can be represented by a bit string of a fixed length, say 100 bits, each bit corresponding to an element of w. The longer the string, the greater the accuracy, but the greater the cost in terms of space and time. Each bit in a string is 1 or 0 according to whether the element it corresponds to is or not in the incidence being represented.

The incidence of $A \& B$ can then be calculated by taking the logical *and* of the incidences of A and B; the incidence of $A \vee B$ can be calculated by taking the logical *or* of the incidences of A and B; and the incidence of $\sim A$ can be calculated by taking the logical *not* of the incidence of A.

We will usually use uninterpreted points, i.e. points without any intended meaning. In this case we can chose their probabilities to be whatever is convenient. To simplify the calculation of sentence probabilities, the points of w can be taken as equiprobable, then:

– let $n(S)$ be the number of elements in set S.
Since the points are equi-probable, for each point I, $\varrho(I) = 1/n(w)$,
hence, for each subset, S, of w,

$$wp(S) = n(S)/n(w)$$

– so, for each sentence, A,

$$p(A) = n(i(A))/n(w) \qquad \text{(xiii)}$$

We can now redo the calculations of Section 3, but using incidences rather than probabilities. Let $w = \{0, 1, \ldots, 99\}$ which might be internally represented by a bit string, as described above. Using a w of size 100 will enable us to calculate probabilities to 2 decimal places.

Suppose A is a sentence with probability 0.75. We will assign to A the incidence $\{0, 1, \ldots, 74\}$. Now using the incidence equations of Section 4:

$$i(\sim A) = \{75, 76, \ldots, 99\} \qquad \text{(by (ix))}$$

$$i(A \,\&\, \sim A) = \{\ \} \qquad \text{(by (x))}$$

$$i(A \vee \sim A) = \{0, 1, \ldots, 74, 75, \ldots, 99\} \qquad \text{(by (xi))}$$

hence,

$$p(A \,\&\, \sim A) = 0 \qquad \text{(by (xiii))}$$

and

$$p(A \vee \sim A) = 1 \qquad \text{(by (xiii))}$$

which is as desired.

However, if B is a sentence, independent of A, with probability 0.25, $p(A \,\&\, B)$ is different from $p(A \,\&\, \sim A)$. Suppose we assign to B the incidence $\{0, 4, 8, \ldots, 96\}$, then:

$$i(A \,\&\, B) = \{0, 4, 8, \ldots, 72\} \qquad \text{(by (x))}$$

hence,

$$p(A \,\&\, B) = 0.19 \qquad \text{(by (xiii))}$$

$$i(A \vee B) = \{0, 1, 2, \ldots, 74, 76, \ldots 96\} \qquad \text{(by (xi))}$$

hence,

$$p(A \vee B) = 0.81 \qquad \text{(by (xiii))}$$

which is correct to 2 decimal places, as desired.

If the domain supports a more meaningful representation of points, then these can be used instead of the equi-probable, uninterpreted points proposed above. For instance, in a domain involving time, it might be possible to specify a number of hypothetical futures, e.g. it rains, it snows, there is sunshine, etc. It may then be possible to assign incidences to axioms in a principled manner. Note that it would still be possible to represent the incidences with bit strings. If these hypothetical futures are not equi-probable, then we must assign a probability to each of them to enable $wp(I)$ to be calculated for each incidence. When the incidences are meaningful to the user we might expect him/her to input them directly, rather than use probabilities. In this case we may not be interested in calculating probabilities at all.

Incidences can be a more efficient method of storing probabilistic information than probabilities. Consider the Propositional Incidence Calculus situation and suppose that we want to assign the probabilities of a set of sentences. Let this set contain n different propostions. If we are to conduct inference with these sentences then we may have to add new sentences to the set during inference, so we had better consider the set as containing all sentences constructable from the n propositions. If all sentences are put in conjunctive normal form we can see that there are 2^{2^n} different sentences altogether whose probability needs to be stored. However, I am reliably informed★ that the probabilities of all these sentences can be recovered provided the probabilities of all the 2^n clauses are known. To store a decimal number of m digits requires $10.m$ bits, thus $10.m.2^n$ are required altogether. To record a probability of m digits in an incidence requires a sample space of size 10^m, i.e. 10^m bits. But the incidences (and hence probabilities) of all sentences of the set can be recovered from the incidences of the n propositions, so only $n.10^m$ bits are required altogether. In a typical expert system m will be very small compared with n (uncertainly measures are usually subjective so a value of $m > 2$ would be spurious accuracy, whereas n will have the same order of magnitude as the number of production rules), so $10^m.n \ll 10.m.2^n$. Therefore, the minimum storage required for the incidences is significantly less than that required for probabilities.

6. A Correlation + Probability Calculus Cannot Have Truth Functional Connectives

We are now in a position to redeem the promise of Section 3 to show that a correlation + probability calculus cannot have truth functional connectives. In particular, we will show that it is not possible to give equations which enable calculation of $c(A \,\&\, B, C)$ purely from $p(A)$, $p(B)$, $p(C)$, $c(A, B)$, $c(A, C)$ and $c(B, C)$.

To do this we need only exhibit two situations in each of which the values of $p(A)$, $p(B)$, $p(C)$, $c(A, B)$, $c(A, C)$ and $c(B, C)$ are identical but where $c(A \,\&\, B, C)$ has different values. Such a pair of situations is exhibited in the Venn diagrams of Figure. 1.

In these situations $w = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ and each of these points is equi-probable. The incidences of A, B, C, $A \,\&\, B$, etc. are assigned as in the diagram, e.g. $i(A) = \{0, 3, 4\}$ and $i(A \,\&\, B) = \{3\}$ in situation 1. From these assignments we

★D. M. Titterington, personal communication.

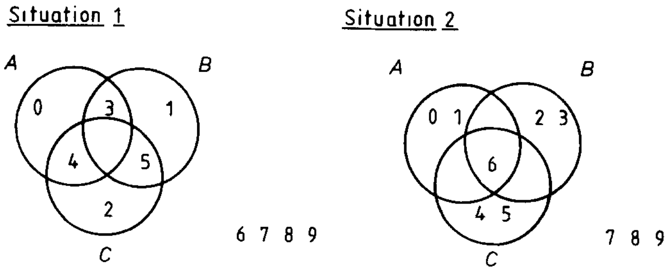

Fig. 1. A Correlation + Probability Calculus cannot have truth functional connectives.

can use sentences (xiii) and (vi) to calculate that:

$p(A) = p(B) = p(C) = 0.3$ in both situations and that

$p(A \,\&\, B) = p(B \,\&\, C) = p(A \,\&\, C) = 0.1$ in both situations.

Hence,

$$c(A, B) = c(B, C) = c(A, C)$$

$$= (0.1 - 0.3 \times 0.3)/\sqrt{0.3 \times 0.7 \times 0.3 \times 0.7}$$

$$= 0.047619 \quad \text{in both situations.}$$

But in situation 1 $p(A \,\&\, B \,\&\, C) = 0$, so

$$c(A \,\&\, B, C) = (0 - 0.1 \times 0.3)/\sqrt{0.1 \times 0.9 \times 0.3 \times 0.7}$$

$$= -0.21822$$

and in situation 2 $p(A \,\&\, B \,\&\, C) = 0.1$, so

$$c(A \,\&\, B, C) = (0.1 - 0.1 \times 0.3)/\sqrt{0.1 \times 0.9 \times 0.3 \times 0.7}$$

$$= -0.50918$$

Considered study of this example should convince you that having different values of $c(A \,\&\, B, C)$ for the same values of $p(A)$, $p(B)$, $p(C)$, $c(A, B)$, $c(A, C)$ and $c(B, C)$ is not an exceptional situation, but rather the norm. So we cannot hope for a rough correlation calculus which would merely work most of the time.

Neither can we hope for a calculus based on a modifed form of correlation. To be useful any such function, $d(X, Y)$, would have to enable $p(X \,\&\, Y)$ to be calculated purely from itself and $p(X)$ and $p(Y)$, as in sentence (vi). Thus $d(X, Y)$ would be expressible purely in terms of $p(X)$, $p(Y)$ and $p(X \,\&\, Y)$. Hence, $d(X \,\&\, Y, Z)$ would be expressible purely in terms of $p(X \,\&\, Y)$, $p(Z)$ and $p(X \,\&\, Y \,\&\, Z)$. The counter-example given in Figure 1 makes the values of $p(A)$, $p(B)$, $p(C)$, $p(A \,\&\, B)$, $p(B \,\&\, C)$ and $p(A \,\&\, C)$ identical in the two situations, but makes the values of $p(A \,\&\, B \,\&\, C)$ different. Thus $d(A, B)$, $d(B, C)$ and $d(A, C)$ would be the same and $d(A \,\&\, B, C)$ would be different, however $d(X, Y)$ was defined.

For instance, suppose d were conditional probability. Now $p(A|B) = p(A|C) = p(B|A) = p(B|C) = p(C|A) = p(C|B) = 0.333\ldots$ in both situations in Figure 1, but $p(A \& B|C) = 0$ in situation 1 and $0.333\ldots$ in situation 2. Thus a conditional probability + probability calculus would not have truth functional connectives either.

Note that the example also shows that the probabilities of all the sentences in the set cannot be calculated from the probabilities and correlations (or conditional probabilities) of just the propositions in the set. Thus the recording of correlations (or conditional probabilities) together with probabilities will not require storage space less than that for incidences.

7. Inference Under Uncertainty

Any calculus for uncertainty reasoning needs to provide some mechanims for inheriting uncertainty values from the hypothesis of an inference step to the conclusion, i.e. if we know the uncertainty of A and $A \rightarrow B$ then we need to be able to calculate the uncertainty of B when modus ponens is applied. By analogy with the truth functionality of Propositional Logic, we will call this property of a calculus *proof functionality*.

However, this proof functional criterion is too strong and must be relaxed in general. Consider the case of modus ponens, B may be derivable in several ways, e.g. from C and $C \rightarrow B$. Any particular derivation can only provide a lower bound on the certainty of B. But we would like to make this lower bound as tight as possible, i.e. to make the calculus as proof functional as possible.

In Incidence Calculus the uncertainty of a sentence is its incidence. In general, if $A \vdash B$ is a rule of inference of a logical system then all we can infer is that $i(A) \subseteq i(B)$; if A is true in some point then B will be true in that point. Each derivation of B provides a new lower bound for its incidence. The greatest lower bound is found by taking the union of these lower bounds. This is legitimised by the set theoretic theorem.

$$L_1 \subseteq I \,\&\, L_2 \subseteq I \rightarrow L_1 \cup L_2 \subseteq I$$

where I is the incidence and L_1 and L_2 two lower bounds.

The maintenance of a lower bound of the true incidence is in the same spirit that MYCIN amalgamates the certainty factors calculated from different derivations of the same conclusion, except that the MYCIN amalgamation algorithm is ad-hoc whereas ours is justified by set theory.

In Shafter–Dempster theory both a lower and an upper bound are maintained, [Lowrance and Garvey 82]. The lower bound, Spt(A), represents the degree to which the evidence supports A; the upper bound, Pls(A), represents the degree to which the evidence fails to refute A.

In some cases inference steps *are* proof functional, e.g. when deriving $A \& B$ from A and B then $i(A \& B) = i(A) \cap i(B)$. Note that in a probability calculus

$$p(A \,\&\, B) = p(A).p(B) + c(A, B).\sqrt{p(A).p(\sim A).p(B).p(\sim B)}$$

where $c(A, B)$ can vary between -1 and $+1$, so for given $p(A)$ and $p(B)$, $p(A \,\&\, B)$ can take a range of values. Thus a probability calculus would not be proof functional in this case. We can rephrase this observation by saying that in order for an uncertainty mechanism to be proof functional for this rule of inference it is necessary that & be truth functional with respect to the uncertainty measure. & is truth functional with respect to incidences but not probabilities. Similar remarks will hold for any rule of inference in which the hypotheses are all subformulae of the conclusion. This emphasises the importance of the connectives being truth functional with respect to the uncertainty measure.

In general, the lower bounds provided by Incidence Calulus are much tighter than those provided by a purely numeric probability calculus because the connectives are truth functional with respect to incidences but not to probabilities.

The Rules of Inference of Predicate Calculus preserve truth, i.e. if the hypothesis is true in some interpretation then the conclusion is true in that interpretation. Thus we might have a rule $A \,\&\, B \vdash A$, but not a rule $A \vdash A \,\&\, B$. However, in an uncertainty calculus, we can extend the notion of rule of inference to include both of these rules: while the first, given $i(A \,\&\, B)$, provides a new lower bound for $i(A)$, the second, given $i(A)$ provides a new upper bound for $i(A \,\&\, B)$. If we are only given upper and lower bounds for the hypotheses then both rules provide both new upper and new lower bounds for their conclusions.

An inference engine for an Incidence Logic can and should exploit this extended notion of rule of inference operating on upper and lower bounds of the incidences of the sentences of the logic. The Legal Assignment Finder of section 8 provides such an inference engine for Propositional Incidence Logic. It is a extension of the Beth's Semantic Tableau and is shown to be complete for Propositional Incidence Logic in [Bundy 85].

The Legal Assignment Finder incorporates rules of inference corresponding to each of the connectives, for inheriting both upper and lower bounds and going in both directions. That these connectives are truth functional with respect to incidences is a direct contribution to the proof functionality of the inference engine.

The Legal Assignment Finder is input an initial assignment of upper and lower bounds for the incidences of a set of sentences in Propositional Logic. It uses rules of inference to specialize this initial assignment to just those legal assignments of incidences to these sentences. Suppose F is an assignment, then it defines the functions $\sup_F$ and $\inf_F$ from sentences to incidences. The former gives the upper bound and the latter the lower bound.

For instance, there is a rule of inference

$$\text{And1:} \quad \sup_G(A) = [\sup_F(A \,\&\, B) \cup w\backslash\inf_F(B)] \cap \sup_F(A)$$

where assignment G is a specialization of assignment F. The justification of this rule is that since

$$i(A \,\&\, B) \subseteq \sup_F(A \,\&\, B) \quad \text{and}$$

$$w\backslash i(B) \subseteq w\backslash \inf_F(B) \quad \text{then}$$

$$i(A \mathbin{\&} B) \cup w\backslash i(B) \subseteq \sup_F(A \mathbin{\&} B) \cup w\backslash \inf_F(B)$$

But

$$i(A) \subseteq i(A \mathbin{\&} B) \cup w\backslash i(B)$$

and

$$i(A) \subseteq \sup_F(A)$$

Therefore

$$\sup_F(A) = [\sup_F(A \mathbin{\&} B) \cup w\backslash \inf_F(B)] \cap \sup_F(A)$$

The Legal Assignment Finder has 4 such rules for negation and 6 each for conjunction, disjunction and implication.

8. The Legal Assignment Finder

The purpose of the Legal Assignment Finder is to find all the legal specializations of an initial assignment. Only a consistent initial assignment will have any.

This could be done by finding all specializations by case analysis and then rejecting those that are not legal, but this would be unnecessarily inefficient. Instead, it interleaves the legality test with the case analysis. The legality test is done by the set of rules of inference mentioned above. They are used to infer the consequences of an assignment around the sentences, tightening the sup and inf bounds of each. Sometimes this inference specializes the initial assignment into a legal or contradictory assignment without case analysis but, in general, case analysis is required.

DEFINITION 1: *Types of Assignment.* An assignment F is called *total* iff for all A $\inf_F(A) = \sup F(A)$. In this case we introduce a third mapping i_F from $sf(S)$ to w, where $sf(S)$ is the set of sentences S and their subsentences, such that

$$i_F(A) = \inf_F(A) = \sup_F(A) \quad \text{for all} \quad A \in sf(S).$$

$i_F(A)$ is called the *incidence* of A in F.

An assignment F is called *legal* iff it is total and i_F obeys the axioms of Incidence Calculus for $sf(S)$ and w.

An assignment F is called *contradictory* iff for some A $\inf_F(A) \not\subseteq \sup_F(A)$.

An assignment F is a *specialization* of an assignment G iff

$$\inf_G(A) \subseteq \inf_F(A) \quad \text{and}$$

$$\sup_F(A) \subseteq \sup_G(A) \quad \text{for all} \quad A \in sf(S)$$

An assignment F is a *proper specialization* of an assignment G iff

F is a specialization of G and

$\inf_G(A) \subset \inf_F(A)$ or

$\sup_F(A) \subset \sup_G(A)$ for some $A \in sf(S)$

An assignment F is *consistent* iff it has a legal specialization.

Note that specialization and proper specialization are transitive and reflexive.

DEFINITION 2: *The Legal Assignment Finder.* To apply the Legal Assignment Finder to assignment F:

1. Let F be the current assignment.
2. Until the current assignment is contradictory or there are no rules in the queue then:
   (a) Pick a rule from the queue and apply it to the current assignment.
   (b) Let the resulting assignment be the new current assignment.

   Let the current assignment be G.
3. If G is contradictory then exit with result $\{ \}$,
4. Otherwise, if G is total then exit with result $\{G\}$.
5. Otherwise, split G into cases $G1$ and $G2$.
6. Apply the Legal Assignment Finder to $G1$ with result set1.
7. Apply the Legal Assignment Finder to $G2$ with result set2.
8. Exit with result set1 $\cup$ set2.

The results of inference are given below. There are more rules than strictly required to test for legality, but the additional results are valuable in helping to avoid cases analysis.

DEFINITION 3: *The Rules of Inference.* A rule of inference is a mapping from assignments to assignments. Let F be the assignment before the rule fires and G be the assignment afterwards. In each case, G is the same as F except for the changes, on some particular formula, defined below for each rule.

Not1: $\sup_G(A) = [w \backslash \inf_F(\sim A)] \cap \sup_F(A)$

Not2: $\inf_G(A) = [w \backslash \sup_F(\sim A)] \cup \inf_F(A)$

Not3: $\sup_G(\sim A) = [w \backslash \inf_F(A)] \cap \sup_F(\sim A)$

Not4: $\inf_G(\sim A) = [w \backslash \sup_F(A)] \cup \inf_F(\sim A)$

And1: $\sup_G(A) = [\sup_F(A \,\&\, B) \cup w \backslash \inf_F(B)] \cap \sup_F(A)$

And2: $\inf_G(A) = \inf_F(A \,\&\, B) \cup \inf_F(A)$

And3: $\sup_G(B) = [\sup_F(A \,\&\, B) \cup w \backslash \inf_F(A)] \cap \sup_F(B)$

And4: $\inf_G(\mathrm{B}) = \inf_F(A \,\&\, B) \cup \inf_F(B)$

And5: $\sup_G(A \,\&\, B) \;=\; \sup_F(A) \cap \sup_F(B) \cap \sup_F(A \,\&\, B)$

And6: $\inf_G(A \,\&\, B) \;=\; [\inf_F(A) \cap \inf_F(B)] \cup \inf_F(A \,\&\, B)$

For conciseness we omit the rules for disjunction and implication, but these can be easily derived from the ones above.

DEFINITION 4: *Contribution of a Rule*. Note that each rule has the form:

$\sup_G(A) \;=\; \ldots \cap \sup_F(A)$ or

$\inf_G(A) \;=\; \ldots \cup \inf_F(A)$

We call the . . . part of the rule its *contribution*.

These rules are applied to an assignment using the following procedure.

DEFINITION 5: *Rule Firing*. To apply a rule to assignment F to produce assignment G.

1. Remove the rule from the queue.
2. Evaluate the right hand side of the rule.
3. If the result gives a new value for the sup or inf of A, then find all rules which can be instantiated to have the new value in their contributions, and queue those instances.
4. Exit with assignment G.

Before the inference process can begin all the relevant rules must be queued. This is done by the following procedure.

DEFINITION 6: *Initialization*. Let S be the set of sentences we are interested in, and $sf(S)$ be the set of these sentences and all their subsentences.

To initialize $sf(S)$ with assignment F:

Set up an empty queue.

For each sentence $A \in sf(S)$:

1. If $\inf_F(A) \neq \{\ \}$ then queue all instances of rules with $\inf_F(A)$ in their contributions.
2. If $\sup_F(A) \neq w$ then queue all instances of rules with $\sup_F(A)$ in their contributions.

Case splitting is necessary when the inference process runs out of rules to fire without specializing the assignment to a total or contradictory one. It is done by picking a point and considering the two cases (1) that it is not and (2) that it is in the incidence of a sentence.

DEFINITION 7: *Case Splitting*. To split G into cases $G1$ and $G2$

1. Since G is not total and not contradictory there exists an $A \in sf(S)$ such that $\inf_G(A) \subset \sup_G(A)$
2. Choose $j \in \sup_G(A) \backslash \inf_G(A)$ non-deterministically.

3. Let $G1$ and $G2$ be the same as G except that:

$$\sup_{G1}(A) = \sup_G(A)\backslash\{j\}$$

$$\inf_{G2}(A) = \inf_G(A) \cup \{j\}$$

4. Find all rules which can be instantiated to have $\sup_{G1}(A)$ in their contributions and queue those in case $G1$.
5. Find all rules which can be instantiated to have $\inf_{G2}(A)$ in their contributions and queue those instances in case $G2$.

This completes the definition of the Legal Assignment Finder. It has been implemented in Prolog by the author. Various theoretic properties of this algorithm are proved in [Bundy 85]. In particular, we show it complete for Propositional Incidence Logic. We also show that it degenerates into a tautology checker when w is a singleton, and that, therefore, it is at least exponential in the size of the sentences in the input set, and that a complete polynomial algorithm is highly unlikely to exist.

The following illustrates the performance of the algorithm.

Suppose $w = \{j, k, l\}$ and $i(\sim(a \;\&\; \sim b)) = \{j, k\}$.

Assign all proper subsentences, B, of $\sim(a \;\&\; \sim b)$
the default assignment of $\sup(B) = w$ and $\inf(B) = \{\ \}$.

Then:

rule Not1 assigns $\{l\}$ to $\sup(a \;\&\; \sim b)$.

rule Not2 assigns $\{l\}$ to $\inf(a \;\&\; \sim b)$.

rule And2 assigns $\{l\}$ to $\inf(a)$.

rule And4 assigns $\{l\}$ to $\inf(\sim b)$.

rule Not1 assigns $\{j, k\}$ to $\sup(b)$.

No further inference is possible without case splitting.
For instance, $\{l\} \subseteq i(a) \subseteq \{j, k, l\}$.
We could consider the cases
1. $\{l\} \subseteq i(a) \subseteq \{j, l\}$
2. $\{l, j\} \subseteq i(a) \subseteq \{j, k, l\}$

and then continue the inference process in each case.

In Predicate Incidence Logic no complete terminating algorithm exists. Such an algorithm would have to degenerate into a complete terminating algorithm for detecting contradictions in standard Predicate Calculus, and this task is known to be semi-decidable. However, we can extend the Legal Assignment Finder to an incomplete inference engine for the Predicate Incidence Logic with the following rules of inference.

DEFINITION 8: *Predicate Incidence Logic Rules of Inference*:

All1: $\sup_G(\forall X\ A(X)) = \sup_F(A(s)) \cap \sup_F(\forall X\ A(X))$

All2: $\inf_G(A(s)) = \inf_F(\forall X\ A(X) \cup \inf_F(A(s))$

Exist1: $\sup_G(A(s)) = \sup_F(\exists X\ A(X)) \cap \sup_F(A(s))$

Exist2: $\inf_G(\exists X\ A(X)) = \inf_F(A(s)) \cup \inf_F(\exists X\ A(X))$

where s is a term containing no variables not bound in $A(X)$
Note that some mechanism is required to control the introduction of new terms s, i.e. to introduce new sentences of the form $A(s)$. In contrast to the Propositional case the set of sentences may grow during the course of the inference.

9. Maintaining Consistency

If the user of an expert system based on Incidence Calculus is able to assign incidences in an uncontrolled manner, then it is possible to make an inconsistent assignment. For instance, it follows from the equations of Incidence Calculus that:

$$w \backslash i(A) \subseteq i(A \rightarrow B)$$

so $A \rightarrow B$ cannot be assigned an incidence independently of A. Suppose that $w = \{j, k, l\}$ and the user assigns $i(A) = \{j\}$ and $i(A \rightarrow B) = \{j, k\}$, then:

$$\begin{aligned} \{j\} &= i(A) \\ &\subseteq i(A) \cap i(\sim B) \\ &= i(A \ \& \sim B) \\ &= i(\sim(A \rightarrow B)) \\ &= w \backslash i(A \rightarrow B) \\ &= \{j, k, l\} \backslash \{j, k\} \\ &= \{l\} \end{aligned}$$

so, $\{j\} \subseteq \{l\}$, which is a contradiction.
This possibility of building an inconsistent theory is true of any theory, but is particularly easy to do unintentionally in Incidence Calculus.

In Propositional Incidence Calculus, if w is finite, these inconsistencies can be detected by the Legal Assignment Finder.

On the problem above this algorithm makes the following calculations:*

$\sup(A) = \{j\}$
$\inf(A) = \{j\}$

*For simplicity we have omitted the assignment subscripts on sup and inf. Successive groups of lines represent successive assignments.

$$\sup(A \to B) = \{j, k\}$$
$$\inf(A \to B) = \{j, k\}$$

$$\sup(B) = w$$
$$\inf(B) = \{\ \}$$

$$\sup(A) = [w \backslash (\{j, k\} \cap w \backslash w)] \cap \{j\} = \{j\}$$
$$\inf(A) = (w \backslash \{j, k\}) \cup \{j\} = \{l, j\}$$

$$\inf(A) \not\subseteq \sup(A)$$

Therefore, exit with result { }, indicating an inconsistent initial assignment.

Note that this example does not involve case analysis. In [Bundy 85] we have proposed that a version of the Legal Assignment Finder without case analysis be used for consistency checking. We call this algorithm the *Inconsistency Detection Algorithm*. This is computationally much cheaper and picks up all of the straightforward inconsistencies.

10. Assigning Incidences

To initialize an incidence based system, sentences must be given as axioms and incidences must be assigned to them. In expert systems like MYCIN the initial assignment of numerical 'uncertainty factors' is made subjectively by the user.

If we assume that users are prepared to assign probabilities (numbers) to axioms, but not incidences (sets), then our task is to convert probabilities into incidences. Since incidences incorporate more information about the sentences than probabilities, namely the degree of independence of the sentences, we must either make assumptions about this extra information or provide a mechanism for the user to input it. For instance, we would assume that each axiom is as independent of the others as is allowed by the equations of Incidence Calculus. This problem is discussed and an algorithm proposed in [Corlet and Todd 85]. Alternatively, we could allow the user to supply sufficient correlations of conditional probabilities between sentences to uniquely determine the assignment of incidences.

If assumptions are made in the assignment of incidences then an inconsistent assignment of incidences may be made, even though the probability assignment is consistent. An inconsistent assignment of incidences can be detected by running the Legal Assignment Finder on the current assignment or the cheaper, but incomplete, Inconsistency Detection Algorithm. In the latter case we may not detect inconsistency even though it is present. If consistency is detected then it is necessary to back up and remake one of the incidence assignments, relaxing one of the assumptions (e.g. of independence). This is computationally very expensive since we must save back-up points during incidence assignment. When all back-up points are exhausted then the original probability assignment must have been inconsistent, so the user must be informed and asked to remake it.

Although, initially, some sentences may be assumed independent, the Legal Assignment Finder will still detect dependences between sentences and will assign incidences to reflect this dependence. For instance, consider the example at the end of Section

5; the assignments to A and B reflect their independence, but the assignment to $\sim A$ reflects its dependence on A. Once the assignment to A had been made the assignment to $\sim A$ would be automatically made by rules Not3 and Not4 of the Legal Assignment Finder.

Note that to determine uniquely the incidences of all sentences in some set it is not enough to know the probabilities and correlations (or conditional probabilities) of just the propositions in the set. For instance, consider the situations described in Figure 1 and suppose that the following information were all available:

$$p(A) = p(B) = p(c) = 0.3 \quad \text{and}$$

$$c(A, B) = c(B, C) = c(C, A) = 0.47619 \quad \text{or equivalently}$$

$$p(A|B) = p(A|C) = p(B|A) = p(B|C) = p(C|A) = p(C|B) = 0.333..$$

then both the assignments given in Figure 1 are legal, i.e. they both fit the numbers above. However, they yield different values for $p(A \,\&\, B \,\&\, C)$, namely 0 and 0.1, respectively. If situation 1 is chosen, but 0.1 is later assigned to $p(A \,\&\, B \,\&\, C)$ then the Inconsistency Detection Algorithm will detect an inconsistency, even though the probabilities, correlations and conditional probabilities assigned are not, in fact, contradictory. Backtracking would be required to find the consistent assignment, situation 2.

It is unclear how much information is required to determine uniquely the incidences in a set of sentences. It may well be that the correlations (or conditional probabilities) are required of all pairs of sentences in the set – even of those sentences whose probability is to be determined. Note that in the Predicate Incidence Logic case, this set is potential infinite, so even the storage of these correlations would be impossible, let alone the acquisition.

An alternative solution is to assign incidences directly, rather than via probabilities and correlations. These incidences might be subjective or objective. Subjective incidences would be estimated by the expert or user, just as uncertainty numbers usually are in expert systems. Objective incidences would be provided by data from field studies, just as objective probabilities are provided in Bayesion decision aids.★

The human provider of subjective incidences would have to be presented with some sample space, w, of incidents from which to chose. In practice, w would have to be small and might be presented in some graphic form, e.g. a screen divided up into an array. It would help if the incidents were meaningful to the human and related to the domain, e.g. days of the week. The Inconsistency Detection Algorithm could track the assignment and warn of any inconsistency. Empirical studies are required to see whether this is feasible. It is known that humans often object to providing uncertainty numbers; would incidences be better or worse?

The provision of objective incidences involves less statistical analysis than the provision of objective probabilities. Each set of identical experiments becomes an incident and is stored in the incidence of those formulae that were true in the

★I am indebted to Richard O'Keefe for suggesting using objective incidences.

experiments. One does not need to calculate the probability of the formulae, but one does need to estimate the probability of the individual incidents. If each experiment is considered equi-probable then the probability of each incident can be calculated from the number of experiments in its set. One also needs to assume that these incidents cover all the cases.

Whether subjective or objective incidents are used, the assumption of equi-probable incidents will almost certainly need to be dropped. This does not entail a great cost. The representation of incidences as bit-strings can still be maintained, but the calculation of probabilities from them is a little more complicated (see Section 4 for the details). This calculation does not need to be done very often, for instance, only when the final diagnosis is being output, if then.

11. Related Work

It is interesting to compare the Incidence Calculus with the *Probabilistic Logic* of Nilsson, [Nilsson 84]. In Probabilistic Logic, points are partially-specified Herbrand Models, i.e. they are vectors of truth assignments to a finite sequence of sentences. Thus the incidence of each sentence consists of those points with the value 'true' in the vector slot corresponding to that sentence. The following calculation is used to find the probability of a new sentence, given the probabilities of some old sentences.

(a) Construct all the points for the set of old and new sentences.
(b) Make an assignment of probabilities to each point which will yield the given probabilities of the old sentences.
(c) Use this assignment to calculate the probability of the new sentences.

Step (b) above is computational impractical in general. Nilsson suggests various ways to circumvent it in special cases. We avoid this problem by using uninterpreted points rather than using Herbrand Models, as Nilsson does. The probability of each of our points is set to $1/n(w)$, and hence an impractical calculation is avoided. Instead of fixing the probabilities of the sentences by adjusting the probability of the points, it is done by adjusting the size of their incidences, and this is a much less expensive calculation.

The storage of $\sup(A)$ and $\inf(A)$ rather than $i(A)$ is very similar to the storage of the interval $[\mathrm{Spt}(A),\mathrm{Pls}(A)]$ in Shafer–Dempster theory rather than the probability, $p(A)$. In fact,

$$\mathrm{Spt}(A) = n(\inf(A))/n(w) \quad \text{and}$$

$$\mathrm{Pls}(a) = n(\sup(A))/n(w)$$

So Incidence Calculus can easily be adapted to provide a mechanism for dealing with the problem of dependent sentences in Shafer–Dempster theory, instead of in probability theory. Instead of assigning $i(A)$ on the basis of $p(A)$ we would want to assign $\sup(A)$ and $\inf(A)$ on the basis of $\mathrm{Pls}(A)$ and $\mathrm{Spt}(A)$, and instead of calculating and outputting the probability of any inferred conclusion we would output a range of probabilities.

12. Conclusion and Further Work

In this note we have described a mechanism, Incidence Calculus, for incorporating probabilistic reasoning in an inference system. This mechanism is based on the assignment of sets of points to sentences, rather than the normal technique of assigning numbers to sentences. Our mechanism provides a probabilistic logic with truth functional connectives. A purely numeric mechanism cannot provide this. Because of this property, Incidence Calculus provides tighter bounds on the probabilities of inferred sentences than could be provided by such a numberic mechanism.

Incidence Calculus can be implemented reasonably efficiently using bit strings. It is then a more efficient technique for storing the probabilities of sentences than a purely numeric mechanism, especially if the dependencies between sentences must also be stored.

There is a problem over the initial assignment of incidences to sentences. An initial assignment of probabilities is not enough to determine uniquely the incidences. Pairwise correlations or conditional probabilities are also required between all sentences in the input set. In the Predicate Incidence Logic case this set is potentially infinite. Either one must make an initial assignment and be prepared to remake it when further information is available or incidences must be input directly.

We have presented a complete inference engine for Propositional Incidence Logic, the Legal Assignment Finder, and outlined the extensions required to extend this into an inference engine for Predicate Incidence Logic. More work is required to explore the theoretical properties of this extended engine, e.g. completeness and soundness, and to implement and test it.

Incidence Calculus can be readily adapted to Shafer–Dempster Theory, i.e. to returning an interval of probabilities rather than a single value.

Some of the mechanisms described in this paper have been implemented. Further implementation and testing is required.

Acknowledgements

I am grateful for comments and advice on this paper from Roberto Desimone, Richard O'Keefe, David Spiegelhalter, Bernard Silver, Steve Owen, M.J.R. Healy, D.M. Titterington, Allen White, Chris Robertson, Peter Fisk and Alberto Petterossi.

An earlier draft of this paper appeared in the proceedings of the International Conference on Fifth Generation Computer Systems, Tokyo, 1984.

This research was supported by SERC/Alvey grant, number GR/C/20826.

References

[Bundy 85] Bundy, A., *Correctness criteria of some algorithms for uncertain reasoning using Incidence Calculus*. Research Paper, Dept. of Artificial Intelligence, Edinburgh, 1985. Submitted to the *Journal of Automated Reasoning*.

[Corlett and Todd 85] Corlett, R. A. and Todd, S. J., 'A Monte-Carlo approach to uncertain inference'. In Ross, P. (editor). *Proceedings of AISB-85*, pp. 28–34 (1985)

[Duda *et al.* 78] Duda, R. O., Hart, P. E., Nilsson, N. J. and Sutherland, G. L., 'Semantic network representations in rule-based inference systems'. In Waterman, D. and Hayes-Roth, F. (editors), *Pattern-directed Inference Systems*, pp. 203–221. Academic Press (1978).

[Feller 68] Feller, W. *An Introduction to Probability Theory and Its Applications*. John Wiley & Sons, Third Edition (1968).

[Lowrance and Garvey 82] Lowrance, J. D. and Garvey, T. D., 'Evidential reasoning: A developing concept'. In *Proceedings of the International Conference on Cybernetics and Society*, pp. 6–9. IEEE (1982).

[Nilsson 84] Nilsson, N., *Probabilistic Logic*. Technical Note 321, SRI International (1984)

[Shortliffe 76] Shortliffe, E. H., *Computer-based Medical Consultations: MYCIN*. North Holland (1976).

[Zadeh 81] Zadeh, L. 'PRUF – a meaning representational language for natural languages'. In Mamdani, E. and Gaines, B. (editors) *Fuzzy reasoning and its applications*. Academic Press (1981).