Acquisition of Uncertain Rules in a Probabilistic Logic

John G. Cleary
Dept. Computer Science,
University of Calgary,
2500 University Dr.,
Alberta T2N 1N4,
Canada.
uucp: cleary@calgary ARPA:cleary.calgary.ubc@csnet-relay CRNET:
cleary@calgary.cdn CSNET:cleary.calgary@ubc

ABSTRACT

The problem of acquiring uncertain rules from examples is considered. The uncertain rules are expressed using a simple probabilistic logic which obeys all the axioms of propositional logic. By using three truth values (true, false, undefined) a consistent expression of contradictory evidence is obtained. As well the logic is able to express the correlations between rules and to deal both with uncertain rules and with uncertain evidence. It is shown that there is a subclass of such rules where the probabilities of correlations between the rules can be directly computed from examples.

UNCERTAIN RULES

Uncertainty is an important part of many ruled based expert systems. For example, applications such as medical diagnosis do not allow anything but the weakest inferences to be made from available evidence. To report certain conclusions is both incorrect and misleading. A number of schemes for expressing and computing such uncertainties have been developed. For example, in MYCIN and EMYCIN "certainty factors" are used (Shortliffe, 1976). A certainty factor is a number between -1 and +1, -1 is intended to express sure knowledge that something is false and +1 sure knowledge that it is true. Intermediate values express varying degrees of ambivalence about the truth. For example 0 expresses a complete lack of knowledge about truth or falsity. Such certainty factors can be used in a number of ways. For example a rule such as:

will-rain ⇐ dark-cloud and falling-pressure with certainty 0.6;

says that it is almost certainly true that if there is dark cloud around and the barometric pressure is falling then it will rain, although, there will be some cases where this is not true. Certainty factors can be propagated through the system to evaluate the certainty of conclusions. For example the conclusion 'will-rain' would be given a certainty factor based on the certainty of the rule above and the certainties of 'dark-cloud' and 'falling-pressure'. Unfortunately there are grave problems with using certainty factors in this way. Consider the additional rule:

will-rain ⇐ lightning with certainty 0.4;

If both these rules fire then 'will-rain' is given a higher certainty factor than if only one of them fires. Unfortunately the second rule is merely another way of saying that storm clouds are present and the conclusion is not much more true as a result.

These problems can be seen more starkly by considering the expression '(dark-cloud or not dark-cloud)' or the expression '(dark-cloud and not dark-cloud)' which are respectively always true or false. However certainty factors ignore the fact that the two parts of the expression are correlated (the same) and report intermediate values for both expressions.

Another way to approach this is to use the probability that an expression is true rather than certainty factors. Again problems arise that are similar to those above. For example, one scheme used to compose such probabilities is obtained by assuming the various parts of an expression are uncorrelated:

$p \text{ and } q = p \times q$
$p \text{ or } q = p + q - p \times q$
$\text{not } p = 1-p$

Let p=1/2 then (p and not p) = 1/4 and (p or not p) = 3/4, neither of which is correct.

A second evaluation scheme (Zadeh, 1965) assumes that some correlation can occur between expressions and lets:

$p \text{ and } q = \min(p,q)$
$p \text{ or } q = \max(p,q)$
$\text{not } p = 1-p$

Again let p=1/2 then (p and not p) = 1/2 and (p or not p) = 1/2 which are both wrong. This scheme does not overweight rules which are similar but does underweight rules which are independent.

Another weaker scheme (Quinlan, 1983) uses intervals of probabilities. It is based on the principle that the following inequalities always hold:

$\max(0,1-p-q) \leq (p \text{ and } q) \leq \max(p,q)$
$\max(p,q) \leq (p \text{ or } q) \leq \min(1,p+q)$
$\text{not } p=1-p$

By only asserting that the probability for an expression lies in some range it never asserts anything which is false. For example if p = 1/2 it deduces that:

$0 \leq (p \text{ and not } p) \leq 1/2$

and $1/2 \leq (p \text{ or not } p) \leq 1.$

While true these inequalities are too weak to be generally useful.

A related problem is that some conclusions may have evidence both indicating that they are true and indicating that they are false. It seems, intuitively, that the situation where there is no evidence about something is different from the one where there is a known 0.5 probability that it is true and a known 0.5 probability that it is false. The schemes mentioned above also have problems in consistently accomodating both positive and negative evidence.

Considerations such as these show that it does not seem to be possible to provide a quantitative theory of truth and falsity using just probability values for expressions. In the next section an alternative is introduced which circumvents this by using (potentially infinite) sequences of bits to represent the truth or falsity of a statement. Probabilities can

be extracted after reasoning is complete but the calculations cannot take place using just the probabilites. Other attempts to provide a logical basis for uncertain reasoning have been made (Shapiro, 1983), (van Emden, 1986) but because they are based on the probability values of expressions they also are heir to the ills described above.

A PROBABILISTIC LOGIC

The technique used here is to assign each expression an infinite sequence of true/false values rather than just one true/false possibility. This is trivially different from the normal propositional calculus. All theorems hold, for example, (p or not p) = (true, true,...) and (p and not p) = (false,false, ...). In order to make this useful a new family of logical constants are introduced. Each constant is some infinite sequence of true/false values with a fixed probability that it will be true. The constants are written in the form $\tau(x)$ where x is the probability that an item in the sequence is true. So, it is always true that $\tau(1)$ = (true, true, ..) and $\tau(0)$ = (false, false, ...).

When evaluating a logical expression with these constants some actual value has to be assigned to them. There are a number of ways of doing this and it is convenient to assume that different constants are uncorrelated. That is, the probability of a true in the sequence ($\tau_1(x)$ and $\tau_2(y)$) is always $x \times y$. In practice these infinite uncorrelated sequences are likely to be approximated by finite sequences of pseudo-randomly generated bits.

To express an uncertain rule the constants can be used as follows:

will-rain $\Leftarrow$ dark-cloud and falling-pressure and $\tau(0.6)$;

This says that if there is dark cloud and falling barometric pressure then in 0.6 of the cases there will be rain. The second rule can be expressed as:

will-rain $\Leftarrow$ lightning and $\tau(0.4)$;

This has still not solved the problem that the two rules are highly correlated but they can be reformulated as:

$$\left.\begin{array}{l}\text{will-rain} \Leftarrow \text{dark-cloud and falling-pressure and } \tau_1(0.75)\\ \text{and}\\ \text{will-rain} \Leftarrow \text{lightning and } \tau_2(0.5)\end{array}\right\} \Leftarrow \tau_3(0.8)$$

This reformulation says that the two rules have a common cause which says that they are true 80% of the time. As a result of this the probability of will-rain will only be weakly augmented when both rules fire, solving the original problem of expressing the fact that the two rules are not independent of each other. By rewriting these rules in the equivalent form below it can be seen that the original $\tau(0.6)$ has been replaced by ($\tau_1(0.75)$ and $\tau_3(0.8)$) and $\tau(0.4)$ by ($\tau_2(0.5)$ and $\tau_3(0.8)$):

will-rain $\Leftarrow$ dark-cloud and falling-pressure and $\tau_1(0.75)$ and $\tau_3(0.8)$

will-rain $\Leftarrow$ lightning and $\tau_2(0.5)$ and $\tau_3(0.8)$

In this way sets of rules with arbitrary correlations between them can be expressed. However, it is not clear that all possible ways of using the logical constants are useful. The form used above where sets of rules are enabled seems to be an intuitive way of expressing such realtionships and, as will be seen later, it has some advantages when the probabilities of the constants are acquired from experience. However, the acid test for such formalisms

is whether computer naive experts can express their intuitions in this form. No tests of this have been done nor has there been exploration of possible ways of "sugaring" the syntax to make it more palatable.

Probabilities can be extracted from our logical sequences by counting the number of true values in the sequence and taking the limit. The result of all this is a probability logic (Gaines, 1984), (Rescher, 1963). The logic obeys all the usual logical axioms including the tautology (x or not x).

Rule Sets

As is usual in rule based systems, it is necessary in practice to restrict attention to the Horn clause subset of the logic. In the next section the logic and the types of allowable rules will be extended to cater for negation and the possibility of evidence both for and against a proposition.

The normal situation in an expert system is a fixed body of rules (Horn clauses) which encodes the invariant knowledge about the problem at hand, and a set of facts which describes the current situation. The existence of a probabilistic logic allows some useful extensions to this view. For example, a fact can be stated as an additional "rule" of the form:

$$\text{dark-clouds} \Leftarrow \tau(1)$$

and additionally uncertain evidence (I think there is a 50% chance that those are dark clouds) can be accomodated by a rule of the form:

$$\text{dark-clouds} \Leftarrow \tau(0.5)$$

There is also nothing to stop the head of such a "factual" rule from from being an intermediate deduction which is also computed elsewhere by a set of rules. For example the user might say "it is going to rain with probability 0.5 although there is nothing in the rules to support this." and enter this as a "factual" rule:

$$\text{rain} \Leftarrow \tau(0.5)$$

In order to effectively compute the consequences that result from such rule sets it seems to be necessary to restrict them to those with finite derivation trees (van Emden, 1986). That is, no recursive rules are permitted. This accords well with the normal situation in rule-based expert systems. The use of the α–β heuristic in such evaluations has been discussed for truth-functional logics of uncertainty (ibid), and can also be used here.

Bit Sequences

Infinite sequences of bits can be made computationally tractable by approximating them with finite pseudo-random sequences of bits. The probabilities are in turn approximated by counting over these finite sequences. The major question this approximation raises is the accuracy with which the finite sequences represent the probabilities which would be generated by infinite length sequences. The accuracy is dependent on both the number of bits used and on the ability of the system to generate a large number of uncorrelated pseudo-random sequences. It is easily shown that the standard deviation of the estimated probability is given by $\sqrt{p(1-p)/N}$, where N is the number of bits and p is the probability of

a bit being true. So for N=32 and p=1/2 the results will be accurate to ±10%, for N=1024 this is reduced to ±1.5%. Even the ±10% figure is well within the errors acceptable in existing systems (Shortliffe, 1976, p183). Conversely, 1024 bits is only 32 32-bit words so the logical operations required should not be too burdensome amongst all the other activities of an expert system. It is necessary to generate a number of uncorrelated random sequences for the different τ sequences. As noted in (Gaines, 1969, Sec. 4.16) it is easy to generate a large number of uncorrelated sequences with p=1/2 using suitably long shift registers. For example, with N=1024 a single 33 bit shift register can deliver 2^{23} independent sequences. These are readily combined to deliver sequences with appropriate probabilities. The stochastic computing systems described in (Gaines, 1969) are well suited to performing the types of computations needed here.

NEGATION_

Negation poses problems for Horn clause logics in general. The essential problem is that the statement of facts about particular situations must include information that some things are true, that others are false and that some are just not known. The usual way of handling this is to make the closed world assumption that *if something cannot be proven then it is false*. This is far too draconian for the current purposes as it makes it impossible to express that a fact is unknown. It is ridiculous to conclude that because I do not know whether it is sunny therefore it must definitely be cloudy.

A resolution of this is to extend the truth values to include *undefined* as well as true and false and to introduce two distinct forms of negation. As above, the truth values are infinite sequences (this will be ignored whenever convenient). The interpretation of true is "provably true" and of false is "provably false". The two forms of negation are denoted by ~ and ¬. Their truth tables are:

	t	u	f
~	f	u	t

	t	u	f
¬	f	t	t

~ should be interpreted as "provably not" and ¬ as "not provable". So ¬ corresponds to the normal closed world notion of negation. The truth tables for the various connectives are:

∧	t	u	f
t	t	u	f
u	u	u	f
f	f	f	f

∨	t	u	f
t	t	t	t
u	t	u	u
f	t	u	f

⇐	t	u	f
t	t	u	f
u	t	u	u
f	t	t	t

Note that as a result of these definitions a ⇐ b is equivalent to a ∨ ~b.

We would like the expression (x or not x) always to be true and (x and not x) always to be false. These need to be restated carefully in the new logic but there do indeed exist

statements with the correct properties. For example, $x \vee \neg x$ is always true (x is provably true or not provably false) and $\sim x \vee x$ is never false. Similarly, $\neg(\neg x \wedge x)$, $\sim(\neg x \wedge x)$ and $\neg(\sim x \wedge x)$ are always true.

Extended Horn Clauses

To accommodate these new notions the form of clauses allowed in the rule set can be extended as follows. The general form of rules is:

$$a \Leftarrow b_1, b_2, b_3, \ldots b_n$$

where a, the head of the clause, can be a term of the form x or $\sim x$ and the b_i can be of the form $\sim x$, $\neg x$ or x where x is some atomic formula. Also, the b_i can be constants of the form $\tau(p)$ or $\sim\tau(p)$. $\neg x$ is not permitted in the head of a clause. Because the τ constants can only occur on the right-hand side of rules which will not "fire" if any of their terms are undefined, there is no need to allow for undefined values in the infinite constants. There is no logical reason to exclude them, they just serve no useful purpose in this Horn clause logic.

The operational interpretation of these rules is that whenever x appears in the head of a rule whose body evaluates to true, then x is forced to true. Similarly if $\sim x$ appears in the head of a rule then this forces the truth value of x to false. A weaker form of the closed world assumption is needed to completely define this procedure, that is, "if a value cannot be proven true or proven false then it is undefined". This seems much more palatable than the original form. These procedures ensure that none of the rules is provably false.

Contradiction

This opens the possibility that two rules will attempt to force the same conclusion to be both true and false. In systems with a single truth value such a contradiction is catastrophic, the rule set has to be rejected (or debugged). If an infinite sequence of values is available the situation is not as bad. Any position along the sequence which generates a contradiction on any value at the head of a rule will cause all conclusions at that position to be ignored. In a rule set where this happens a lot, the precision of the inferred probabilities will drop as a smaller number of positions are available to count from. Although any high probability of contradiction will be untenable the system is at least graceful enough to allow some degree of contradiction and not fail catastrophically.

ACQUISITION

Deriving Constants From Observations

Putting logic aside briefly, we can consider the normal task of statistics which is exemplified by the question "predict whether it will rain given the current observations". Statistics tells us that what should be done is to count the number of times it has rained when the values of all the observations are the same as the current situation. The fraction of times that it has rained in these circumstances is the probability that it will rain again. However, typically there are many observations that can be made and it is unlikely that exactly the current situation has ever occurred in the past. This is now a form of the zero frequency problem, "what is the probability that something which has never occurred in the

past will occur now". Statistics *per se* can give us no answer to this question; it must be sought from other knowledge about the problem.

One way to do this is to select some subpart of the current situation which is judged to be particularly relevant. If this subpart is sufficiently small then the current values will have occurred before and statistics can now be brought into play. What results is effectively a rule. For example, if the features judged to be relevant to rain are dark clouds and falling barometric pressure then statistics tell us how often the rule:

will-rain $\Leftarrow$ dark-cloud and falling-pressure;

is true. More can be gained by reformulating this into a rule in probabilistic logic:

will-rain $\Leftarrow$ dark-cloud $\wedge$ falling-pressure $\wedge$ $\tau_1(p)$

p can be estimated by counting all the observations on which the statistics are based. To be more precise the value of τ_1 can be computed using the inverse rules:

$$\sim \tau_1 \Leftarrow \text{dark-cloud} \wedge \text{falling-pressure} \wedge \sim \text{will-rain} ;$$

and $$\tau_1 \Leftarrow \text{dark-cloud} \wedge \text{falling-pressure} \wedge \text{will-rain} ;$$

The first of these is needed to avoid contradictions. The second is adopted to maximize the utility of the rule and to avoid the trivial situation where τ_1 is always false. The number of instances where τ_1 is (provably) true or false can be used to estimate p (the cases where it is undefined need not be counted).

These two rules for computing τ_1 are particulalry useful because they are in the Horn clause form introduced earlier and so can be readily computed. Unfortunately, not all uses of logical constants can be so readily inverted. I will first consider some useful cases where the inversion is possible and then discuss some of the difficulties which arise in other cases.

The general case is that which arises when using nested rules of the form

$$\left.\begin{array}{l}\text{will-rain} \Leftarrow \text{dark-cloud} \wedge \text{falling-pressure} \wedge \tau_1 \\ \\ \text{will-rain} \Leftarrow \text{lightning} \wedge \tau_2\end{array}\right\} \Leftarrow \tau_0$$

Following the same reasoning as above the requirement that the rules fire as often as possible gives:

$$\tau_1 \wedge \tau_0 \Leftarrow \text{dark-cloud} \wedge \text{falling-pressure} \wedge \text{will-rain}$$
$$\tau_2 \wedge \tau_0 \Leftarrow \text{lightning} \wedge \text{will-rain}$$

These two implications readily expand to the four Horn clauses

$$\tau_1 \Leftarrow \text{dark-cloud} \wedge \text{falling-pressure} \wedge \text{will-rain}$$
$$\tau_2 \Leftarrow \text{lightning} \wedge \text{will-rain}$$
$$\tau_0 \Leftarrow \text{dark-cloud} \wedge \text{falling-pressure} \wedge \text{will-rain}$$
$$\tau_0 \Leftarrow \text{lightning} \wedge \text{will-rain}$$

The requirement that the rules not be contradictory gives the two implications:

$$\sim(\tau_1 \wedge \tau_0) \Leftarrow \text{dark-cloud} \wedge \text{falling-pressure} \wedge \sim\text{will-rain}$$
$$\sim(\tau_2 \wedge \tau_0) \Leftarrow \text{lightning} \wedge \sim\text{will-rain}$$

These cannot be so readily handled as there is ambiguity as to which of the constants should be set false when a rule fails. This ambiguity can be resolved by again saying that the rules should fire as often as possible and so only setting τ_0 false when both rules fail that is:

$$\sim\tau_1 \Leftarrow \text{dark-cloud} \wedge \text{falling-pressure} \wedge \sim\text{will-rain}$$
$$\sim\tau_2 \Leftarrow \text{lightning} \wedge \sim\text{will-rain}$$
$$\sim\tau_0 \Leftarrow \sim\tau_1 \wedge \sim\tau_2$$

Thus the original two Horn clauses give seven inverted Horn clauses for computing the probabilities of the constants.

This procedure is readily extended to any set of such nested rules where each constant appears only once. The rule sets may also be nested more than one deep as in:

$$\left(\begin{array}{ll} \left(\begin{array}{l} a \Leftarrow b \wedge \tau_2 \\ c \Leftarrow d \wedge \tau_3 \end{array} \right) \Leftarrow \tau_1 & \\ & \Leftarrow \tau_0 \\ e \Leftarrow f \wedge \tau_4 & \end{array} \right)$$

The relationships allowed by such rules are tractable in that their inverses give Horn clauses. They also seem to be able to encompass a useful set of inter-relationships between rules. It may be that a larger tractable class of rules is available but some simple examples will illustrate the problems involved. Consider the following three rules:

$$a \Leftarrow b \wedge \tau_1$$
$$a \Leftarrow c \wedge \tau_2$$
$$d \Leftarrow e \wedge \tau_1$$

The naive inverse rules for this case are:

i) $\tau_1 \Leftarrow b \wedge a$
ii) $\sim\tau_1 \Leftarrow b \wedge \sim a$
iii) $\tau_2 \Leftarrow c \wedge a$
iv) $\sim\tau_2 \Leftarrow c \wedge \sim a$
v) $\tau_1 \Leftarrow e \wedge d$
vi) $\sim\tau_1 \Leftarrow e \wedge \sim d$

Because τ_1 occurs twice in the head of a rule it may be forced into a contradiction. For example, if a, b, and e are true and d is false then τ_1 should be set to both true and false. Presumably this is a signal that the rules should be debugged. However, the situation is slightly more complex than this. Because a occurs twice at the head of two of the original rules, i) and iii) together are too strong. If c is also true then the second rule will fire and it is not necessary to force τ_1 to be true. This situation can be resolved by the new weaker inverse rule:

i) $\tau_1 \Leftarrow b \wedge a \wedge \neg\tau_2$

Because of the very complex interelationships that can exist between the constants it is not clear in general how to resolve such situations and in particular how to guarantee that there are not circular realtionships in the inverse rules.

Examples

A particularly interesting way of constructing the τ constants is to use example cases provided by the expert. These are presumably carefully chosen synthetic cases which are important in practice. They can be handled in the same way as normal observations with the extra possibility that the counts can be more heavily weighted than cases encountered in day to day experience.

Observations where some of the facts are uncertain can also be handled in a similar way. For example, it may be known only that it is cloudy with probability 0.4 and sunny with probability 0.3 (leaving an uncertainty of 0.3). When cloudy is set true, one set of values for the constants will be computed; if it is set false (or undefined) then another set will be computed. So if a number of different pseudo-observations are made where cloudy is true 40% of the time and false 30% of the time then the counts for the various constants can be incremented by a suitably small value each time. That is the same procedure can be used for evaluating the inverse rules as is used for evaluating the original rules.

SUMMARY

The major contribution of this paper is to provide a logical framework within which uncertain rules can be expressed and their uncertainties assessed from experience. The logic on which the rules are based does not suffer from many of the problems which other more *ad hoc* schemes are heir to. Most importantly the logic allows an automatic way of updating uncertainties including the correlations and redundancies between different rules.

There are a number of unresolved questions. It is not clear how easily such rules and their relationships can be expressed and understood by experts. Similarly, the subset of rule

types which can be easily inverted needs to be extended where possible and evaluated in practice.

ACKNOWLEDGEMENTS

This work was supported by the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

van Emden, M.H. (1986) "Quantitative Deduction and its Fixpoint Theory," *J. Logic Programming,* **3**(1) 37-53.

Gaines, B.R. (1969) "Stochastic Computing Systems," in *Advances in Information Science* Vol. II. Tou, J. (Ed). 37-173. Plenum.

Gaines, B.R. (1984) "Fundamentals of Decision," *Studies in the Management Sciences,* **20**, 47-65.

Rescher, N. (1969) *Many-Valued Logic.* New York, McGraw-Hill.

Shapiro, E.Y. (1983) "Logic Programs with Uncertainties," in *Proc. 8th I.J.C.A.I.* Bundy, A. (Ed). 529-532. William Kaufman.

Shortliffe, E.H. (1976) *Computer-Based Medical Consultations.* New York, Elsevier.

Quinlan, J.R. (1983) "Inferno: a Cautious Approach to Uncertain Inference," *The Computer Journal,* **26**(3) 255-269.

Zadeh, L.A. (1965) "Fuzzy Sets," *Information and Control,* **8**, 338-353.