IMPLEMENTATION OF AND EXPERIMENTS WITH A VARIABLE PRECISION LOGIC INFERENCE SYSTEM*

Peter Haddawy
Intelligent Systems Group
Department of Computer Science
University of Illinois
Urbana, Illinois 61801

ABSTRACT

A system capable of performing approximate inferences under time constraints is presented. Censored production rules are used to represent both domain and control information. These are given a probabilistic semantics and reasoning is performed using a scheme based on Dempster-Shafer theory. Examples show the naturalness of the representation and the flexibility of the system. Suggestions for further research are offered.

I INTRODUCTION

It is a Sunday afternoon and your fully autonomous car is taking you for a drive. Suddenly a truck pulls out into the road ahead of you. Your car has 5 seconds to decide what to do. If your car were powered by current reasoning technology, chances are it would never reach a decision because while trying to determine the best course of action it would hit the truck, destroying both itself and you. In this situation any decision, even a rough guess, is better than indecision. What is needed is a system capable of producing the best decision possible within a given time limit.

In any practical reasoning process there are extra-logical cost constraints such as time and resource limitations which must be taken into account. The tradeoff between the cost, certainty, and specificity of inferences can be used to flexibly adjust to these constraints. Certainty refers to the degree of belief in a statement, while specificity refers to the degree of detail of a description. The idea of a logic in which the certainty of an inference could be varied to conform to cost constraints was presented by Michalski and Winston [1985]. This Variable Precision Logic (VPL) used censored production rules to encode both domain and control information. The rules take the form

P -> D ⌊ C
read *If P then D unless C.*

The unless part of the rule is called the censor. Censors represent exception conditions and as such are considered to be false most of the time. Therefore, the determination of their truth values is given a lower priority than that of rule antecedents. Whereas unlimited resources are devoted to checking the antecedents, only a limited amount of resources is devoted to checking the censors in time critical situations. The unless symbol is logically interpreted as an exclusive-or operator between the censor and the consequent. Thus, given the rule

Sunday -> Go to the park ⌊ Weather is bad,

we can conclude that if it is Sunday and the weather is good, I will go to the park; and if it is Sunday and the weather is bad, I will not go to the park. Formally,

$$P_1 \wedge P_2 \wedge \cdots \rightarrow D \lfloor C_1 \vee C_2 \vee \cdots$$

is logically interpreted as

$$P_1 \wedge P_2 \wedge \cdots \wedge \neg(C_1 \vee C_2 \vee \ldots) \rightarrow D$$

and

$$P_1 \wedge P_2 \wedge \cdots \wedge (C_1 \vee C_2 \vee \ldots) \rightarrow \neg D$$

* This work was supported in part by the Defense Advanced Research Project Agency under grant N00014-K-85-0878, in part by the National Science Foundation under grant NSF DCR 84-06801, and in part by the Office of Naval Research under grant N00014-82-K-0186.

In order to make the exceptions quantitative, numerical parameters are associated with each rule, representing the strength of inference when the truth value of the censor is known and when the value is unknown. These values allow the precision of inferences to be varied.

This paper presents a formalization of the notion of uncertainty in censored production rules and an implementation of an inference system capable of varying the certainty of inferences to conform to given time limits.

II FORMALISM AND THEORY

Uncertain inference in the VPL system is performed using a scheme based on Dempster–Shafer theory [Shafer 1976]. Domain information is represented in the form of rules and facts. A fact is assigned a certainty represented by a Shafer interval, [s p]. The s value indicates the support for a proposition, while the p value indicates its plausibility. The intervals for A & ¬A are related by p(A) = 1 – s(¬A). The s and p values can also be thought of as the minimum and maximum probabilities of the proposition. The amount of uncertainty in a proposition is defined as the difference of the values. A value of 'unknown' is represented as the interval [0 1]. The value of a conjunction or disjunction of facts is calculated by applying the formulas for probabilistic product or sum respectively to the support and plausibility values separately.

Rules are interpreted as expressing conditional probabilities. Beliefs are propogated across the rules using an approach similar to that employed by Ginsberg [1984] and derived by Dubois & Prade [1985]. Suppose we have the rule A –> B, where $prob(B \mid A) \in [S_r P_r]$ and $prob(A) \in [S_A P_A]$. Then it can be shown that $prob(B) \in [S_A S_{r,}\ 1 - S_A + S_A P_r]$. Now if $prob(\neg B \mid A) \in [S_{\bar{r}} P_{\bar{r}}]$ then $P_r = 1 - S_{\bar{r}}$, from which it follows that $prob(B) \in [S_A S_{r,}\ 1 - S_A S_{\bar{r}}]$. To use this scheme, the certainty of a rule is represented by four values: $\alpha\ \beta\ \gamma\ \delta$, where

$$
\begin{aligned}
\alpha &= S_r \text{ for } P \wedge \neg C \text{ –> } D \\
\beta &= S_{\bar{r}} \text{ for } P \wedge C \text{ –> } \neg D \\
\gamma &= S_r \text{ for } P \text{ –> } D \\
\delta &= S_{\bar{r}} \text{ for } P \text{ –> } \neg D
\end{aligned}
$$

These values are constrained by the following restrictions:

$$
\begin{aligned}
\alpha + \beta &\leq 1 \\
\gamma + \delta &\leq 1
\end{aligned}
$$

When the value of the censor is known, the Shafer interval for the conclusion D is computed according to

$$
\begin{aligned}
s(D) &= s(P)[1 - p(C)]\alpha \\
p(D) &= 1 - s(P)s(C)\beta.
\end{aligned}
$$

When the censor value is unknown, the formulas are

$$
\begin{aligned}
s(D) &= s(P)\alpha \\
p(D) &= 1 - s(P)\beta.
\end{aligned}
$$

Evidence for multiply argued conclusions is combined using Dempster's orthogonal sum rule. This requires the assumption that the evidence events are conditionally independent. The formula used to combine two Shafer intervals is similar to that in [Ginsberg 1984]:

$$
[a\ \ b] \oplus [c\ \ d] = \left[1 - \frac{\bar{a}\,\bar{c}}{1 - (a\bar{d} + \bar{b}c)} \quad \frac{bd}{1 - (a\bar{d} + \bar{b}c)} \right]
$$

The above discussion has presented an approximate inference scheme for propositional logic, but the VPL system uses a typed predicate logic representation. The type information enumerates the elements of a finite domain for each predicate argument. In this representation, terms containing only ground instances are equivalent to propositional logic and thus present no additional problems. However, a semantics for expressions with free variables is needed. Rules of the form A(x,y) –> B(x), with an associated certainty [s p] are interpreted as ∀x,y p(B(x)¦A(x,y)) = [s p]. This is essentially a short–hand for listing rules over the entire domain of x and y. Similarly, a fact A(x) with certainty [s p] is interpreted as ∀x p(A(x)) = [s p].

III SYSTEM OVERVIEW

The VPL system consists of six main components: the user interface, the parser, the knowledge base, the unifier, the inference engine, and the rule-base analyzer. The system is implemented in Common Lisp and runs on a Symbolics 3640.

The system is designed to be fully interactive for incremental rule base development. The user may assert or retract rules and facts, define new types and predicates, and make queries. Once a rule base is complete, the user may perform an analysis of inference times. The rule-base analyzer determines for each possible query the inference time required for all uniform depths of censor chaining. A censor chain is a rule chain in the search tree leading from a censor. A list of times with associated depths for each query is stored in the time data base.

A user query may have an optional time limit associated with it, in which case the system searches the time data base to determine the maximum censor chaining depth which will guarantee a response in the requried time. If no time limit is specified, chaining depth is unlimited. The system performs backward chaining inference, with possibly limited search depth on censors. Inference is performed in two stages: search and calculation. The search strategy is breadth first and exhaustive. The exhaustive search is achieved by generating all consistent ground instances of any free variables after unification. To satisfy a goal, the system searches for a fact which unifies with the goal. If none is found, it tries to find rules which unify with the goal. If both of these attempts fail, the query is given a value of [0 1]. During the search process, instructions for performing the certainty calculations are put on a calculation stack. When the search terminates, the entries on the calculation stack are evaluated and put on a value list. The computation on the bottom of the stack corresponds to the user query.

IV EXAMPLES

This section presents two simple examples to demonstrate the system's capabilities. The first is intended to highlight the approximate inference methods. The idea is that a bird can fly unless it is a special kind of bird such as a penguin or is in an unusual condition such as dead. The input file shows domain type declarations, followed by predicate declarations, followed by rules.

Following the input file is a log of some example runs. After the rules are loaded, the system is told that tweety is a dead bird, from which it concludes that tweety cannot fly. Next, changing our certainty in tweety's death changes the certainty in his ability to fly proportionately. When the system is told that tweety may be a kiwi, this information combines with the possibility of his death to further decrease our belief in his ability to fly. Finally, if tweety is neither in an unusual condition nor a special bird, he is able to fly.

```
type
(animal (spot rover jane tweety road-runner))
(bird (tweety road-runner))

pred
(is-bird (animal))
(flies (animal))
(is-special-bird (bird))
(is-in-unusual-condition (animal))
(is-penguin (bird))
(is-ostrich (bird))
(is-emu (bird))
(is-kiwi (bird))
(is-domestic-turkey (bird))
(is-dead (animal))
(is-sick (animal))
(has-broken-wing (bird))

assert
((is-bird $x) => (flies $x)
 ⌊ (is-special-bird $x)
 (is-in-unusual-condition $x) 1.0 1.0 .9 .05)

((is-dead $x) =>
 (is-in-unusual-condition $x) 1.0 0.0)

((is-sick $x) =>
 (is-in-unusual-condition $x) 0.9 0.06)

((has-broken-wing $x) =>
 (is-in-unusual-condition $x) 1.0 0.0)
```

```
((is-penguin $x) => (is-special-bird $x) 1.0 0)
((is-ostrich $x) => (is-special-bird $x) 1.0 0)
((is-emu $x) => (is-special-bird $x) 1.0 0)
((is-kiwi $x) => (is-special-bird $x) 1.0 0)
((is-domestic-turkey $x) =>
 (is-special-bird $x) 1.0 0)
```

Example Runs

---- Tweety is a dead bird ----

```
ENTER Command
> assert
ENTER Command or assertion
> ((is-bird tweety) 1 1)
ENTER Command or assertion
> ((is-dead tweety) 1 1)
ENTER Command or assertion
> ?
ENTER Command or make query of system
> (flies tweety)
using censor chaining depth of UNLIMITED
0.103377685 seconds elapsed time

<RESULT> [0.00 0.00]
```

-- reduce certainty in Tweety's death --

```
ENTER Command or make query of system
> (assert ((is-dead tweety) .7 .8))
ENTER Command or assertion
> (? (flies tweety))
using censor chaining depth of UNLIMITED
0.1013306 seconds elapsed time

<RESULT> [0.00 0.30]
```

---- suspect that Tweety is a kiwi ----

```
ENTER Command or make query of system
> (assert ((is-kiwi tweety) .3 .5))
ENTER Command or assertion
> (? (flies tweety))
using censor chaining depth of UNLIMITED
0.10849539 seconds elapsed time

<RESULT> [0.00 0.21]
```

---- Tweety is healthy and normal ----

```
ENTER Command or make query of system
> (assert ((is-in-unusual-condition tweety) 0 0))
ENTER Command or assertion
> ((~ is-special-bird tweety) 1 1)
ENTER Command or assertion
> (? (flies tweety))
using censor chaining depth of UNLIMITED
0.027635619 seconds elapsed time

<RESULT> [1.00 1.00]
```

The next example is the one described in the introduction. It shows the ability of the system to vary the depth of censor chaining in response to time limits. The input file shows rules for determining if a car can stop in time to avoid an obstacle based on the condition of its brakes and the road. A log of the sample run shows the effect of varying the time limit. With a censor chaining depth of 1 or less the system cannot determine the truth values of the road-condition censor and thus uses the more approximate version of the rule.

```
type
(level (low medium high))
(rating (good fair poor))
(substance (gravel ice))
(place (road ground))
(temp-type (below-freezing moderate hot))
(looks (shiny rough))

pred
(speed-distance-ratio (level))
(can-stop-in-time())
(road-condition (rating))
(brake-condition (rating))
(on (substance place))
(temperature (temp-type))
(road-appearance (looks))
(construction-site ())
(sound-of-pebbles-hitting-underside-of-car ())

assert
; rules
( (~ speed-distance-ratio high) =>
  (can-stop-in-time)
  | (road-condition poor)
    (brake-condition poor) 1.0 1.0 .85 .1)

( (on ice road) => (road-condition poor) 1.0 0)

( (on gravel road) => (road-condition poor) .9 .1)

( (temperature below-freezing)
  (road-appearance shiny) => (on ice road) .9 .1)

( (construction-site)
  (sound-of-pebbles-hitting-underside-of-car)
```

```
=> (on gravel road) .9 .1)

; facts
((speed-distance-ratio high) .05 .15)
((temperature below-freezing) .2 .3)
((road-appearance shiny) 0 0)
((construction-site) 1.0 1.0)
((sound-of-pebbles-hitting-underside-of-car)
 .8 .85)
```

Example Runs

—— time limit of 1 second ——

```
ENTER Command or input file
> (? ((can-stop-in-time) 1))
using censor chaining depth of 2
0.08085977 seconds elapsed time

<RESULT> [0.00 0.45]
```

—— time = .05 second ——

```
ENTER Command or make query of system
> (? ((can-stop-in-time) .05))
using censor chaining depth of 1
0.031729784 seconds elapsed time

<RESULT> [0.72 0.92]
```

—— time = .03 second ——

```
ENTER Command or make query of system
> (? ((can-stop-in-time) .03))
using censor chaining depth of 0
0.017470836 seconds elapsed time

<RESULT> [0.72 0.92]
```

—— time limit too low ——

```
ENTER Command or make query of system
> (? ((can-stop-in-time) .01))
Cannot perform inference in requested time.
Minimum guaranteed time is 0.018423745 sec
```

V CONCLUSIONS

It has been shown that the formalism of censored production rules when given a probabilistic semantics allows a system to adjust the certainty of inferences to conform to time constraints. Such a system has numerous applications in situations where decisions must be made in real time and with uncertain information. Examples range from medical expert systems for operating rooms to domestic robots.

In the current system, if the value of a rule's censor is known, the α and β certainty values are used. If the value is unknown, the γ and δ values are used. A better approach would be to look at the degree to which the censor value is known and use this to interpolate between the α & β and γ & δ rule certainty values.

Much world knowledge is best expressed in the form of taxonomies. Taxonomies carry more information than simple collections of rules. To make the system more effective, I am working on incorporating specialized inference rules for reasoning in taxonomies.

This paper has only investigated the trade-off between inference time and certainty. The trade-offs between cost and specificity and certainty and specificity need yet to be explored. This is a direction in which the results of machine learning research hold much promise.

ACKNOWLEDGEMENTS

I would like to thank Prof. Michalski for his guidance, Lisa Wolf for comments on a draft of this paper, and John Wiegand and Jim Kelly for help on an earlier implementation.

REFERENCES

Dubois, D., Prade, H. "Combination and Propogation of Uncertainty with Belief Fuctions" In *Proc. IJCAI-85.* Los Angeles, California, August, 1985, pp. 111-113.

Ginsberg, M.L. "Non-monotonic Reasoning Using Dempster's Rule" In *Proc. AAAI-84.* Austin, Texas, August, 1984, pp. 126-129.

Michalski, R.S., Winston,P.H., "Variable Precision Logic" MIT AI Memo 857, Artificial Intelligence Laboratory, MIT, August, 1985 (accepted for AI Journal, 1986).

Shafer, G. *A Mathematical Theory of Evidence.* Princeton University Press, Princeton, New Jersey, 1976.