CmSc310 Artificial Intelligence

Expert Systems II

1. Reasoning under uncertainty

Knowledge is rarely absolutely certain. In expert systems we need a way to say that something is probably but not necessarily true.

One of the simplest methods to account for uncertainty is the so called Bayes approach.

1.1. Probabilities

P(X) – probability of event X to occur.

Three interpretations:
A. Computed as the ratio of the number of actual occurrences of X and the number of all possible cases when X may occur. E.G. tossing a coin – the probability to get “heads’ is ½, all possible cases are 2.
B. Computed as the frequency of event X occurring in repeated experiments. Based on statistical information.
C. “Computed” as degree of belief in X, this is the subjective interpretation, based on past experience and (usually) subconscious observations.

In the domain of expert systems P(X) is usually B or C.

Let H be our hypothesis.
P(H) is the degree of belief in H without any other evidence.

If we have noticed that H often happens to be true in the presence of some other evidence E, we talk about conditional probability:

\[ P(H|E) = \frac{P(H,E)}{P(E)} \]

where \( P(H,E) \) is the probability of both H and E to be true.

In order to compute \( P(H|E) \) we need to know \( P(H,E) \) and \( P(E) \).

\( P(E) \) can be computed using statistical data. (Say, for medical diagnosis, given 1000 people, how many of them exhibit symptoms E)
How do we compute $P(H,E)$? If two events are independent (no connection between them), the probability to occur at the same time is equal to the product of the probabilities for each of them to occur separately, i.e. $P(A,B) = P(A) \times P(B)$. However, we assume that there is some relation between the hypothesis and the evidence. So, in order to compute $P(H,E)$ we do the following:

From (1) we obtain

(2) $P(H,E) = P(H|E) \times P(E)$.

Since $P(H,E) = P(E,H)$ we can rewrite (2) thus:

(3) $P(H,E) = P(E,H) = P(E|H) \times P(H)$.

Now, we can compute $P(H)$ - given 1000 people, how many have been diagnosed with H. We can compute also $P(E|H)$ - given 1000 people diagnosed with H, how many of them exhibit symptoms E.

Now we have the expression

(4) $P(H|E) = \frac{P(E|H) \times P(H)}{P(E)}$

This expression is known as Bayes' theorem.

This expression can be used for multiple evidence $E_1, E_2,$ etc, provided that $E_1, E_2, \ldots$ are independent. Very often however the evidence exhibits some correlation. The model used to capture such correlations is called Bayesian belief networks, to be discussed in Week 10 under “Probabilistic reasoning” topic.

In expert system, especially in medical expert systems, there may be a large amount of evidence to be processed. The Bayesian probabilistic approach requires evaluating a large amount of conditional probabilities. In many cases this turns out to be an overwhelming task. Next we discuss the Certainty Factors model, a pragmatic approach for modeling reasoning under uncertainty.

1.2. Certainty factors (CF)

Certainty factors represent information about how certain the conclusion in a rule may be. Certainty factors can be attached both to the conditions in an if-then rule and to its conclusion. They are ad hoc values, given by the experts based on experience or by the users when providing initial data (as in the blank loan system).
Certainty factors are not probabilities, they represent beliefs about how strong a given evidence is, to what degree the evidence supports a hypothesis.

Certainty factors are measured using various scales bot numeric (0 – 100, 0 – 10, 0 – 1, -1 to -1) and linguistics ones (certain, fairly certain, likely, unlikely, highly unlikely, definitely not)

In the examples below, certainty factors range from -1 to +1.
- Higher certainty factors indicate strong confidence in a hypothesis.
- Certainty factors that approach -1 indicate confidence against a hypothesis.
- Certainty factors around 0 mean that we don’t have information either for or against a hypothesis.

**How to compute the certainty factor of a conclusion**

An IF-THEN rule consists of premises and a conclusion.

The rule has a certainty factor given by the experts. This certainty factor reflects how strongly the experts believe that the premises support the conclusion.

The premises also have certainty factors given by the user of the system. They reflect the degree of confidence in the premises.

Premises are logical expressions combining conditions (facts, propositions) with AND and OR logical operators.

Rules for computing certainty factors of the premises and the conclusion:
Let P and Q are two conditions,
\[ CF(P \text{ and } Q) = \min(CF(P), CF(Q)) \]
\[ CF(P \text{ or } Q) = \max(CF(P), CF(Q)) \]

Let R be the rule with certainty factor CF(R),
Let CF(premises) be the certainty factor of the premises computed using the two rules above.

\[ CF(\text{conclusion}) = CF(\text{premises}) \times CF(R) \]

Certainty factors of the rules are stored in the knowledge base, certainty factors of the premises and the conclusions are stored in the working memory when a particular session is being run.
Example:

IF (high credit rating AND (high balance OR excellent banking history))
THEN superior loan with confidence 0.8

Assume the user enters “high credit rating” with CF 0.9, “high balance” with CF 0.6, and “excellent banking history” with CF 0.7

CF(high balance OR excellent banking history) = max(0.6, 0.7) = 0.7
CF(high credit rating AND (high balance OR excellent banking history)) = min(0.9, 0.7) = 0.7

CF(superior loan) = 0.7 * 0.8 = 0.56

Certainty factors of conclusions that result from multiple rules

Suppose two rules R1 and R2 give same conclusion with different CFs. Let CF_{R1}(C) and CF_{R2}(C) be the certainty factors of conclusion C obtained by R1 and R2.

To compute the new CF of C we use the following rules depending on the signs of CF_{R1}(C) and CF_{R2}(C):

a. CF_{R1}(C) > 0, CF_{R2}(C) > 0
   \[ CF(C) = CF_{R1}(C) + CF_{R2}(C) - CF_{R1}(C) * CF_{R2}(C) \]

b. CF_{R1}(C) < 0, CF_{R2}(C) < 0
   \[ CF(C) = CF_{R1}(C) + CF_{R2}(C) + CF_{R1}(C) * CF_{R2}(C) \]

c. CF_{R1}(C) and CF_{R2}(C) have opposite signs
   \[ [CF_{R1}(C) + CF_{R2}(C)]/[1 - min(|CF_{R1}(C)|, |CF_{R2}(C)|)] \]

Sources:
Part of the material presented in this section is adapted form

Papers on certainty factors:
The Certainty-Factor Model, by David Heckerman:

2. Case Studies

2.1. MYCIN - rule based approach

MYCIN - developed at Stanford in the 1970s to illustrate AI techniques.

Task: to diagnose and recommend treatment for certain blood infections.

Knowledge is represented as a set of IF-THEN rules with certainty factors. The following is an English version of one of Mycin's rules:

IF the infection is primary-bacteremia
AND the site of the culture is one of the sterile sites
AND the suspected portal of entry is the gastrointestinal tract
THEN there is suggestive evidence (0.7) that infection is bacteroid.

MYCIN was written in Lisp, and its rules are formally represented as Lisp expressions.

MYCIN is a (primarily) goal-directed system, using the basic backward chaining reasoning
Uses various heuristics to control the search for a solution (or proof of some hypothesis).
- to be more efficient
- not to ask too many unnecessary questions

Heuristics:
- First ask preset questions to rule out some hypothesis. Then concentrate on the likely hypothesis and go into full backward chaining mode to try and prove each one.
- Preview the premises of a rule to see if there is a false premise, that would make the rule inapplicable.
- Use first rules with higher certainty factors

The dialogue

- Collect initial data to find a set of plausible hypotheses.
- Ask specific questions to test each hypothesis
- Propose a diagnosis and ask questions to determine appropriate treatment
- Propose treatment
At any stage the user can ask **why** a question was asked or **how** a conclusion was reached, and when treatment is recommended the user can ask for alternative treatments if the first is not viewed as satisfactory.

Other developments from the MYCIN project:
- EMYCIN: expert system shell
- PUFF - developed using EMYCIN shell; for heart disorders.
- GUIDON - a tutoring system based in MYCIN.
- NEOMYCIN - rewritten version of MYCIN, used explicit taxonomy of diseases

### 2.2. Internist

Complex problem-solving strategy based on the techniques of differential diagnosis. 
Aim: to model human way of reasoning

**Knowledge representation:**

- **Disease profiles:**
  - Set of findings associated with a disease.
  - For each finding two numbers:
    - Evoking strength - the likelihood of the disease given the finding
    - Frequency - the likelihood of the finding given the disease.

- **Disease tree:** hierarchy of disease types

- Information about what diseases tend to be associated with what other diseases.

**Problem solving:**

1. Collects initial evidence, selects plausible hypotheses
2. Builds 4 lists for each hypothesis:
   a. Observed findings consistent with the disease
   b. Observed findings not associated with the disease
   c. Findings associated with the disease but not observed
   d. Findings associated with the disease and not known if present
3. Scores the hypothesis according to the above lists. Findings in list (a) contribute positively, in lists (b) and (c) - negatively.
4. Asks for additional information for the hypothesis with highest score.

**Summary:**

- Backward chaining for small number of hypothesis
- Differential methods for large number of hypothesis
- Representing the knowledge about diseases and symptoms in a Bayesian network gives better results.