Intelligent Systems: Reasoning and Recognition

James L. Crowley

ENSIMAG 2 / MoSIG M1 Second Semester 2015/2016

Lesson 12 25 march 2015

Uncertainty and Plausible Reasoning

Sources:

Buchanan, B.G. and Shortliffe, E.H. (eds). Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. Reading, MA: Addison-Wesley, 1984

C. M. Bishop, "Pattern Recognition and Machine Learning", Springer Verlag, 2006.

E. T. Jaynes, Probability Theory: The Logic Of Science, Cambridge University Press, 2003

MYCIN (continued)

Backward Chaining Rules

(Context, Parameters, Value, CF)

Facts: Quadruple (C, P, V, CF)

C = Context (a group of facts related to an entity

P = Parameter (facts)

V = Value (Boolean, Numeric, Symbol, List)

CF = Confidence Factor $CF : [-1,1]$

Backward chaining rules are interpreted as an "AND-OR tree"

$$(C, P_2, V_2, CF_2) \xrightarrow{CF_R} (C, P_1, V_1, CF_1)$$

$$CF_1 = CF_R \cdot CF_2$$

$$(C, P_3, V_3, CF_3) \text{ AND } (C, P_4, V_4, CF_4) \xrightarrow{CF_R} (C, P_1, V_1, CF_1)$$

$$CF_1 = CF_R \cdot \min\{CF_3, CF_4\}$$

Mycin rules can be disjunctive (use OR)

$$(C, P_2, V_2, CF_2) \text{ OR } ((C, P_3, V_3, CF_3) \text{ AND } (C, P_4, V_4, CF_4)) \xrightarrow{CF_R} (C, P_1, V_1, CF_1)$$

$$CF_1 = CF_R \cdot \max\{CF_2, \min\{CF_3, CF_4\}\}$$

Backward Rules Generate Goals

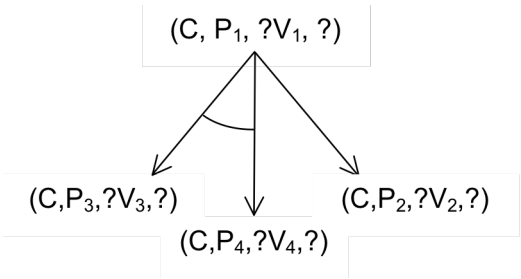
Backward Chaining with Rules generates Goals: (C, P, ?V, ?)

Our two rules are interpreted as:

IF Goal = (C, P_1, ?V_1, ?) Then Ask (C, P_2, ?V_2, ?)

IF Goal = (C, P_1, ?V_1, ?) Then Ask (C, P_3, ?V_3, ?) AND (C, P_4, ?V_4, ?)

Graphical Representation:



Goals are then expanded recursively with additional rules.

Some rules can recursively open a new context. This context must be completed before the previous context.

Independent Rules and the Combine Function.

NOTE that disjunction is different from mutually independent Rules.

With two separate Rules, R1 and R2.

$$(C, P_2, V_2, CF_2) \xrightarrow{CF_{R1}} (C, P_1, V_1, CF_1)$$
$$(C, P_3, V_3, CF_3) \text{ AND } (C, P_4, V_4, CF_4) \xrightarrow{CF_{R2}} (C, P_1, V_1, CF_1)$$

$$CF_1 = \text{Combine}(CF_{R1} \cdot CF_2, CF_{R2} \cdot \min\{CF_3, CF_4\})$$

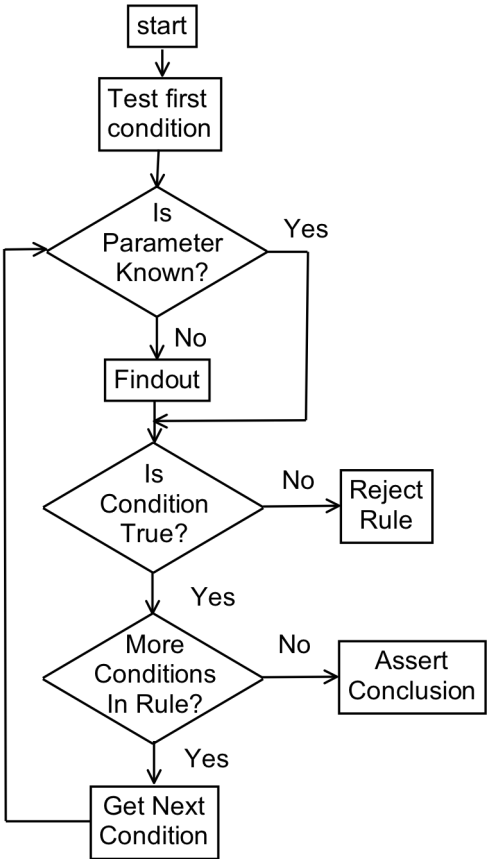
The confidence from independent rules is interpreted with the function Combine

$$Combine(CF_1, CF_2) = \begin{cases} CF_1 + CF_2 - CF_1 \cdot CF_2 & if\ CF_1 \geq 0 \text{ AND } CF_2 \geq 0 \\ -Combine(-CF_1, -CF_2) & if\ CF_1 < 0 \text{ AND } CF_2 < 0 \\ \dfrac{CF_1 + CF_2}{1 - \min(|CF_1|, |CF_2|)} & if\ CF_1 \cdot CF_2 < 0 \end{cases}$$

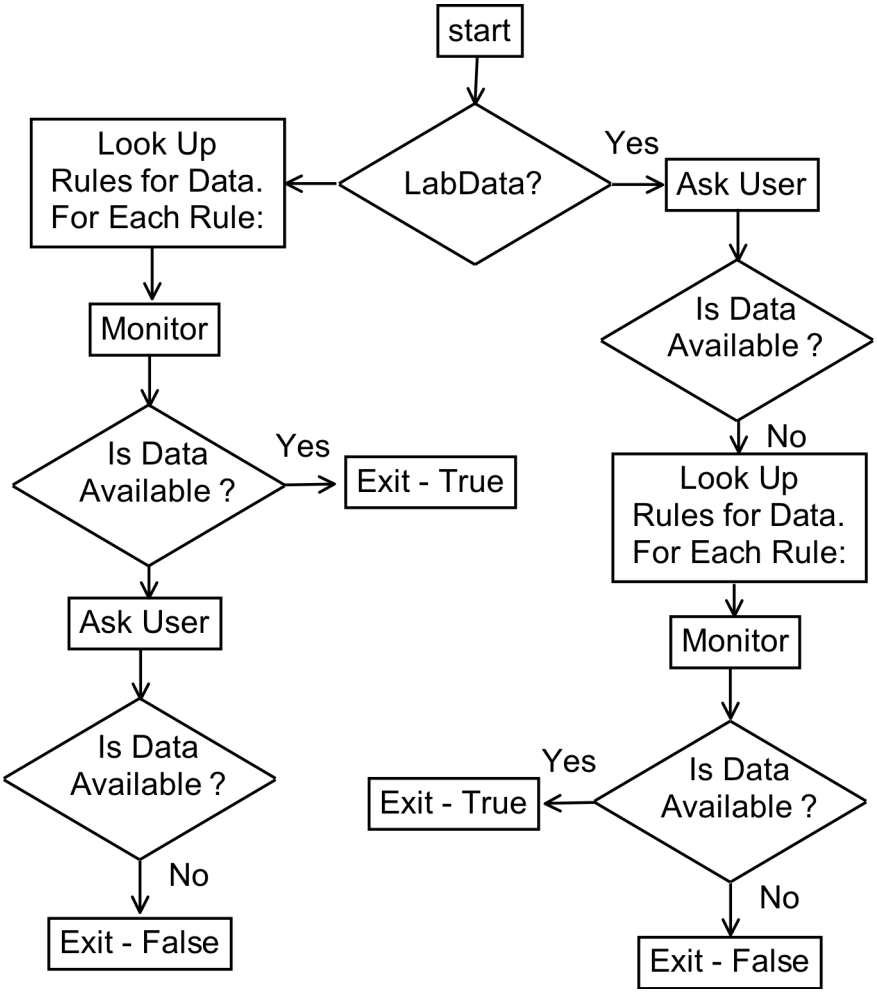
Co-routines: Findout and Monitor

Mycin rules are interpreted using two recursive Coroutines: Findout and Monitor

MONITOR :



FINDOUT:



At any instant the user may ask

WHY? The system provide an interpretation of the trace of reasoning
HOW: The system provides the source for a fact.

Coupled with the extensive use of preprogrammed sentences, this made the system appear to be intelligent. However the knowledge was shallow.

An Example: Your computer

Contexts:

1) Computer: Parameters:
   - Type: one_of{Desktop, Laptop, Tablet}
   - Power: Boolean
   - Running: Boolean

2) Screen: Parameters:
   - State: one_of{on, dark, off, login-window, locked,}
   - Brightness: one_of{dark, dim, bright}

3) Network
   - Wifi: one_of{None, Off, Broken, Connected, not-Connected, Working}
   - Ethernet: one_of{None, Broken, Connected, not-Connected, Working}
   - IP-assigned: Boolean

Parameters are described a variable : AskFirst and by a phrase to request value, and example:

Parameter (Computer, Power, ?V, ?CF)
Type: Ask-First
Question: "Is the computer turned on?")

Findout:

```
If (C, P, ?V, ?CF < Threshold) and (C, P, AskFirst=T) then
    {   Findout(C, P);
        if (C, P, ?V, ?CF < Threshold) THEN Monitor(C, P)
    }
If (C, P, ?V, ?CF < Threshold) and (C, P, AskFirst=F) then
    {   Monitor(C, P);
        if (C, P, ?V, ?CF < Threshold) THEN  Findout(C, P)
    }
```

Rules are associated to Parameters.
Example: Context: Screen, Parameter: State, Value: Dark
For example:

(Screen, State, dark, CF) $\xrightarrow{CF_{R1}}$ (Computer, Power, off, CF)
(PowerSwitch, State, off, CF) $\xrightarrow{CF_{R1}}$ (Computer, Power, off, CF)

Problems with MYCIN approach:

1) Building the domain knowledge is VERY expensive.
2) The uncertainty reasoning system is ad-hoc and unprincipled

Alternative: Bayesian Reasoning

Graphical Models

Graphical models (also known as Bayes Nets) provide a compact and powerful representation for probabilistic reasoning.

1) Then provide a simple way to visualize the structure of a probabilistic model.
2) They provide insights into properties such as conditional independence
3) The can be used to represent probabilistic causal relations in a compact, easy to understand representation.

Graphical models are based on repeated application of two simple rules:

1) The sum rule:

$$P(X) = \sum_{Y} P(X,Y)$$

2) The product rule:

$$P(X,Y) = P(X \mid Y)P(Y)$$

A graph is a structured composed of notes (also called vertices) and links (also called edges and arcs). Each note represents a random variable or a group of random variables. Each link represents a probabilistic relation between random variables. The graph captures product factors.

We will be concerned with Bayesian networks, represented by directed graphical models in which the links have a direction. Such graphs must NOT have any cyclic relations. These are sometimes called "Directed Acyclic Graphs" or DAGs.

These are convenient for representing causal relations between random variables.
That is, they can represent statements such as

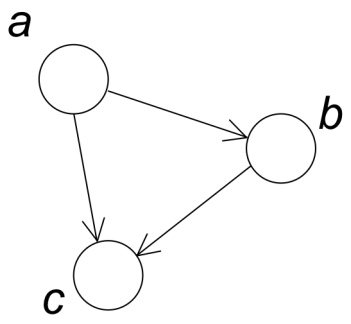
A causes B with probability P.

Consider the relation $p(a, b, c)=p(c \mid a, b)\, p(a, b)$

A second application of the product rule might give

$$p(a, b, c)=p(c \mid a, b)\, p(b \mid a)\, p(a)$$

This would be represented



This would be valid both for probability values and for probability densities.

If P(a), P(b), and P(c) represent discrete probabilities then the graph represents a table (A Conditional Probability Table or CPT).

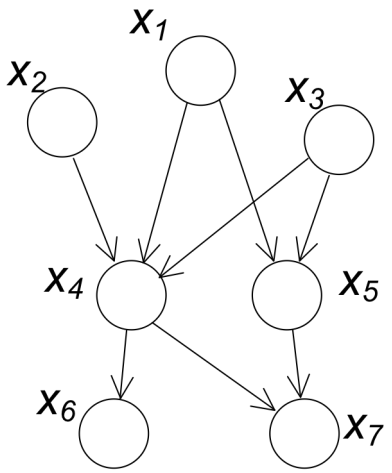
If p(a), p(b) and p(c) are density functions then the result is a Conditional Probability Density.

In general it is the absence of links that conveys interesting information.

For example, the expression |

$$p(x_1)\, p(x_2)\, p(x_3)\, p(x_4 \mid x_1, x_2, x_3)\, p(x_5 \mid x_1, x_3)\, p(x_6 \mid x_4)\, p(x_7 \mid x_4, x_5)$$

would be represented as:



The joint distribution defined by a graph is computed as:

$$P(x) = \prod_{k=1}^{K} P(x_k \mid parents_k)$$

where the $parents_k$ are the parents of x_k in the graph.

Plausible Reasoning

What is truth?
Logical truth: Logical consistency with a set of axioms and postulates.

Plausibility. A state of knowledge of the world. Plausible truth concerns the ability to predict and explain the world.

In the real world, new information is continually arriving. Plausible reasoning assimilates new information as it arrives.

"Plausibility" represents the degree to which a statement can be believed. This can be represented by likelihood or probability.

Plausible reasoning seeks to validate or invalidate hypotheses using uncertain or unreliable information. Plausible reasoning can be used to reason about the truth of single hypothesis (H or ¬H) or choose from a number of competing hypotheses $\{H_i\}$.

Bayes Rule gives us a technique for using evidence to support hypotheses.
Let H be an hypothesis and let E be evidence for the hypotheses.

Then Bayes rule tells us

$$P(H, E) = P(H \mid E)\, P(E) = P(E \mid H)\, P(H)$$

so that $$P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)}$$

We would like to apply this technique recursively, as new data arrives and use this to reason over multiple hypotheses:

Assume that we have K hypotheses, H_k and we seek to accumulate evidence to select the most likely hypothesis. For multiple evidence we could write:

$$P(H_k \mid E_1,...,E_i) = \frac{P(E_1,...,E_i \mid H_k)P(H_k)}{P(E_1,...,E_i)}$$

The problem is how to incrementally estimate $P(E_1, ..., E_i)$ and $P(E_1, ..., E_i \mid H_k)$

To simplify the notation, let us define $S_i=\{E_1,..., E_i\}$ as a body of previous composed of i observations, and E_{i+1} as a new observation.
Let us define:

H = Some hypothesis to be tested
S = Prior (background) Information
E = Evidence, a newly provided observation.

For each new observation, E The problem is how to estimate $P(H_k \mid E,S)$ using the previous evidence S. This can be written:

$$P(H_k \mid E,S) = \frac{P(E,S \mid H_k)P(H_k)}{P(E,S)}$$

The accumulated evidence S is then updated:

$$S \leftarrow S \cup E ;$$

Prior information is NOT the same as "a-priori" information. Prior information informs the plausibility of H before (prior to) including the new data. S can be used to represent "experience" with similar problems. For example, when sampling balls from an urn, S can represent the results of all previous samples.

With this approach we may reason about the truth of single hypothesis (H or ¬H) or choose from a number of competing hypotheses $\{H_k\}$.

The product rule that says

P(A, B | C) = P(A | B, C) P(B | C)

Applying this rule to evidence reasoning we see that

$$P(E,H \mid S) = P(H \mid E,S)P(E \mid S) = P(E \mid H,S)P(H \mid S)$$

From this we can derive

$$P(H \mid E,S) = \frac{P(E \mid H,S)}{P(E \mid S)} P(H \mid S)$$

the term $L(E) = \frac{P(E \mid H,S)}{P(E \mid S)}$ is the "likelihood" of E given H and S.

Applied to recursively to multiple hypothesis this gives:

$$P(H_k \mid E,S) = \frac{P(E \mid H_k,S)}{P(E \mid S)} P(H_k \mid S)$$

This can require renormalizing after each new evidence is assimilated.

Single Hypothesis - the binary case.

The binary case is a realistic and valuable model for many practical problems.
In this case we seek to choose between H or ¬H given E and S.

The product rule tells us that $P(H \mid E,S) = P(H \mid S)\frac{P(E \mid H,S)}{P(E \mid S)}$

but also $P(\neg H \mid E,S) = P(\neg H \mid S)\frac{P(E \mid \neg H,S)}{P(E \mid S)}$

If we take the ratio we obtain:

$$\frac{P(H \mid E,S)}{P(\neg H \mid E,S)} = \frac{P(H \mid S)}{P(\neg H \mid S)} \frac{P(E \mid H,S)}{P(E \mid \neg H,S)}$$

This is known as the "odds" of the hypothesis (côte en Français) :

$$O(H \mid E,S) = \frac{P(H \mid E,S)}{P(\neg H \mid E,S)}$$

Odds are widely used in gambling to express the estimated frequency of occurrence of an outcome compared to an alternative.

From the product rule we can derive:

$$O(H \mid E,S) = O(H \mid S)\frac{P(E \mid H,S)}{P(E \mid \neg H,S)}$$

That is, the posteriori odds are equal to the prior odds multiplied by the likelihood ratio.

Multiple Hypotheses.

In the case of multiple hypotheses, we have K mutually independent hypotheses, H_k. We assume that one of these is correct, and all others are false. Thus

$$\sum_k P(H_k \mid S) = 1$$

Given a new observation, E,

$$P(H'_k \mid E,S) \leftarrow \frac{P(E \mid H_k,S)}{P(E \mid S)} P(H_k \mid S)$$

We note that for this to work. $P(E \mid S) = \sum_k P(E \mid H'_k,S)$
Depending on how we compute $P(E \mid H'_k,S)$ and $P(E \mid S)$ this may not be true.

Alternatively, we can note that because the term *P(E|S)* is common to all hypotheses, it can be ignored. We can then define the relative Likelihood for each hypothesis as

$$L(H'_k \mid E,S) = P(E \mid H_k,S)P(H_k \mid S)$$

The difference is that likelihoods do not sum to 1.
The relative likelihoods can be normalized to provide probabilities by dividing by the sum of likelihoods.

$$P(H'_k \mid E,S) = \frac{L(H'_k \mid E,S)}{\sum_k L(H'_k \mid E,S)}$$

This is technique is used in found in many software tools for Bayesian reasoning.