Uncertainty Management

Bayesian theory

1. Objectives

After studying this chapter you should be able to:

Define or distinguish between:

Uncertain rule	Certainty factor
Uncertain evidence	Crisp set
Bayes'Theorem (or Rule)	Fuzzy set
Prior probability	Fuzzy variable
Affirms weight	Fuzzy rule
Denies weight	Fuzzify
Shallow Bayesian inference network	Defuzzify
Deep Bayesian inference network	

Goal: Write simple rules for operation under conditions of uncertainty using Baysian updating and understand how they work.

2. Introduction

We have now studied the use of rules in considerable depth . Now, We will complicated the use of rules further still by inroducing the idea of ***uncertainty***.
Why should we bother with uncertainty? Essentially because uncertainty is all around. We recognise at least three forms of uncertainty:

a. Uncertain data b.Uncertain rules

Categories a and b can be handled using techniques based on probabily theory. The most important of these techniques is called *Bayesian updating*. The idea here is that knowledge is updated according to the changing degree of certainty as evidence. An attraction of Baysian updating is that it has a theoritical basis.

The other technique is *certainty theory(MEYCIN)*, see the section uncertainty in Prolog ex. of car diagnosis.

The third category of uncertainty is called fuzzy logic. This techniques recognises that is artificial to make decisions on the basis of sharp cut offs. Fuzzy rules allow the dividing line to be blurred. See the chapter Fuzzy logic.

3. Bayesian theory

3.1. Representing uncertainty by probability

Fuzzy: If temperature is high and not (water level is low) then pressure is high.
Mycin: If temperature is high and not (water level is low) then pressure is high cf 80

A Bayesian version of this rule is:
If the temperature is high (affirms 18.0; Denies 0.11) and water level low (affirms 0.1; Denies 1.90) then pressure is high.

O(H) = P(H)/P(~H) = P(H)/(1-P(H)) where P is the probability of the Hypothese H and ~H is NOT H.

P(H) = O(H)/(O(H)+1). Ex. P(H) = 0.2 → O(H) = 0.25.

The standard formula for updating the odds (chance) of a hypothesis H, given that the evidence E is observed is: O(H|E) = **A***O(H).
Where O(H|E) is the odds of H, given the presence of evidence E.
A is the Affirms weight of E. **A** = P(E|H)/P(E|~H)

The standard rule for updating the odds of a hypothesis H given that the evidence E is absent is: O(H|~E) = **D***O(H)
Where O(H|~E) is the odds of H, given the absence of evidence E, and D is the denies weight of E.
D = P(~E|H)/P(~E|~H) or D = 1-P(E|H)/(1-P(E|~H).

3.2. Combining Baysian rules with production rules

Rule: If release valve is stuck THEN release valve needs cleaning.
In this case the hypothesis "release valve needs cleaning" can be asserted with the same probability as the evidence: release valve is stuck.

If evidence1 and evidence 2 THEN hypothesis3.
The probability of hypothesis3 is given by:
P(hypothesis3) = P(evidence1)*P(evidence2).

IF evidence1 or evidence2 THEN Hypothesis3.
The probability of hypothesis3 is given by:
P(Hypothesis3) = P(evidence1) + P(evidence2) – (P(evidence1)*P(evidence2)).

3.3. Worked example 1: Power station boiler

H	E	P(H)	O(H)= P(H)/P(~H)	P(E|H)	P(E|~H)	A= P(E|H)/P(E|~H)	D= 1-P(E|H)/(1-P(E|~H)
release_valve needs cleaning	release_valve is stuck	-	-	-	-	-	-
1.release_valve is stuck	warning_light on	0.02	0.02	0.88	0.4	2.2	0.2
2.release_valve is stuck	pressure is high	0.02	0.02	0.85	0.01	85.0	0.15
3.pressure is high	temperature is high	0.1	0.11	0.90	0.05	18	0.11
4.pressure is high	water_level is low	0.1	0.11	0.05	0.5	0.1	1.90

Values used in worked example 1 of Baysian updating

rule 1
if the release_valve is stuck then the release_valve is need_cleaning .

rule 2
if the warning_light is on **affirms** 2.20 **denies** 0.20 then the release_valve is stuck .

rule 3
if the pressure is high **affirms** 85.00 **denies** 0.15 then the release_valve is stuck .

rule 4
if the temperature is high **affirms** 18.00 **denies** 0.11 and
the water_level is low **affirms** 0.10 **denies** 1.90 then the pressure is high .

We suppose that the rules are fired in the following order:

Rule 4 → Rule 3 → Rule 2 → Rule 1
Rule 4
The rule is:
If Temperature high and NOT(water level low) Then pressure high.

H = pressure is high	O(H) = 0.11
E1 = temperature is high	A1 = 18.0
E2 = water level is low	D2 = 1.9

O(H|(E1&~E2)) = O(H)*A1*D2 = 3.76 <=> update odds of pressure is high is 3.76

Rule 3

H = release valve is stuck	O(H) = 0.02
E = pressure is high	A = 85.0

Because E is not certain (O(E) = 3.76, P(E) = 0.79) the inference engine must calculate an interpolated value A' for the affirms weight of E,
A' = (2*(A-1) * P(E)) + 2 – A = 49.7
O(H|(E)) = O(H)*A' = 0.99

A' = (2*(A-1) * P(E)) + 2 – A	if P(E) >= 0.5
D' =(2*(1-D)*P(E))+D	if P(E) < 0.5

Update odds of release valve is stuck is 0.99 corresponding to a probability of approximately 0.5.

Rule 2

H = release valve is stuck O(H) = 0.99
E = warning light is on A = 2.2
O(H|(E)) = O(H) * A = 2.18
Update odds of release valve is stuck are 2.18

Rule 1

H = release valve needs cleaning
E = release valve is stuck

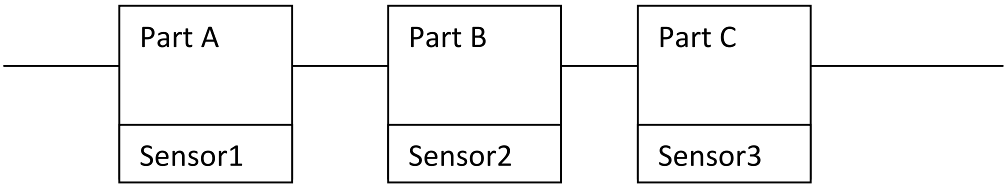
This a production rule, so the conclusion is asserted with the same probability as the evidence. O(E) = 2.18 implies O(H) = 2.18.
Update odds of release valve needs cleaning is 2.18.

O(release valve needs cleaning) = 2.18
P(release valve needs cleaning) = 0.69.

The probability that the valve needs cleaning is 0.69.

The trend is important and not the value!!!

3.4. Worked example 2: Robot



Representaion of an uncertain knowledge-base

Evidence, E	Hypothesis **H**	P(E|**H**)	P(E|**~H**)	A	D
if Sensor1 value high **then**	PartA faulty	0.66	0.12	5.50	0.39
if Sensor2 value high **then**	PartA faulty	0.89	0.2	4.45	0.14
if Sensor3 value high **then**	PartB faulty	0.84	0.24	3.50	0.21
if Sensor2 value high **then**	PartC faulty	0.75	0.12	6.25	0.28
if PartA faulty **then**	Expensive repair	0.34	0.05	6.80	0.69
if PartB faulty **then**	Expensive repair	0.04	0.96	0.04	24.00
if PartC faulty **then**	Expensive repair	0.68	0.13	5.23	0.37
if Expensive repair **then**	Slow recovery	0.92	0.07	13.14	0.09

A and D are calculate from P(E|H) and P(E|~H) according to the formulae:

A = P(E|H)/ P(E|~H)
D = P(~E|H)/ P(~E|~H) = (1 - P(E|H))/ (1- P(E|~H))

if Sensor1 value high Affirms 5.5 Denies 0.39 **then** PartA faulty.

The Bayesian propagation is based on the following rule: O(H|E1 & E2 & E3... En) = A*O(H).
Where $A = P(E_1\&E_2....E_n|H) / P(E_1 \& E_2 ...E_n| \sim H)$.

In this example we have the P(E|H) and P(E|~H), also we calculate the A and D from P(E|H) and P(E|~H) according to the formulae:
A = P(E|H)/ P(E|~H)
D = P(~E|H)/ P(~E|~H) = (1 - P(E|H))/ (1- P(E|~H)).

The A and D are directly used in the rules as Affirms and denies values.

Getting A and D the rules can be coded as follow(this a special version of Prolog):
Rule r41a

If the repair IS expensive AFFIRMS 13.14 DENIES 0.09 then THE recovery is slow.

This rule confirms with a high probability that the recovery is slow in case the repair is expensive.

3.5. *Results*

Also, we see if sensor 1 and sensor 2 and sensor 3 are high then PARTA is fault with a probability of P4 = 0.9607 and PartB is fault with a probability of P5 = 0.777, PartC can be faut with a probability of P6 = 0.862. This means that that recovery will be slow with a probability of P8 = 0.875. This is significant in the real life.

The statement „recovery is slow“ derive from repair expensive which is affirmed with a probability of P7 = 0.92.

4. **Exercice:** Download Netica: www.norsy.com