# Artificial Intelligence & Expert System

## UNIT III

## Handling uncertainty & Learning:

# Bayes' Theorem

One of the most significant developments in the probability field has been the development of Bayesian decision theory which has proved to be of immense help in making decisions under uncertain conditions.

The probability given under Bayes theorem is also known by the name of inverse probability, posterior probability or revised probability. T

This theorem finds the probability of an event by considering the given sample information; hence the name posterior probability. The bayes theorem is based on the formula of conditional probability.

# Bayes' Theorem

$$P(A_1/B) = \frac{P(A_1 \text{ and } B)}{P(B)}$$

Similarly probability of event $A_1$ given event $B$ is

$$P(A_2/B) = \frac{P(A_2 \text{ and } B)}{P(B)}$$

Where

$$P(B) = P(A_1 \text{ and } B) + P(A_2 \text{ and } B)$$

$$P(B) = P(A_1) \times P(B/A_1) + P(A_2) \times P(BA_2)$$

# Bayes' Theorem

$P(A_1/B)$ can be rewritten as

$$P(A_1/B) = \frac{P(A_1)\times P(B/A_1)}{P(A_1)} \times P(B/A_1) + P(A_2) \times P(BA_2)$$

Hence the general form of Bayes Theorem is

$$P(A_i/B) = \frac{P(A_i)\times P(B/A_i)}{\sum_{i=1}^{k} P(A_i)\times P(B/A_i)}$$

Where $A_1$ , $A_2$ ... $A_i$ ... $A_n$ are set of n mutually exclusive and exhaustive events.

# Bayesian Networks

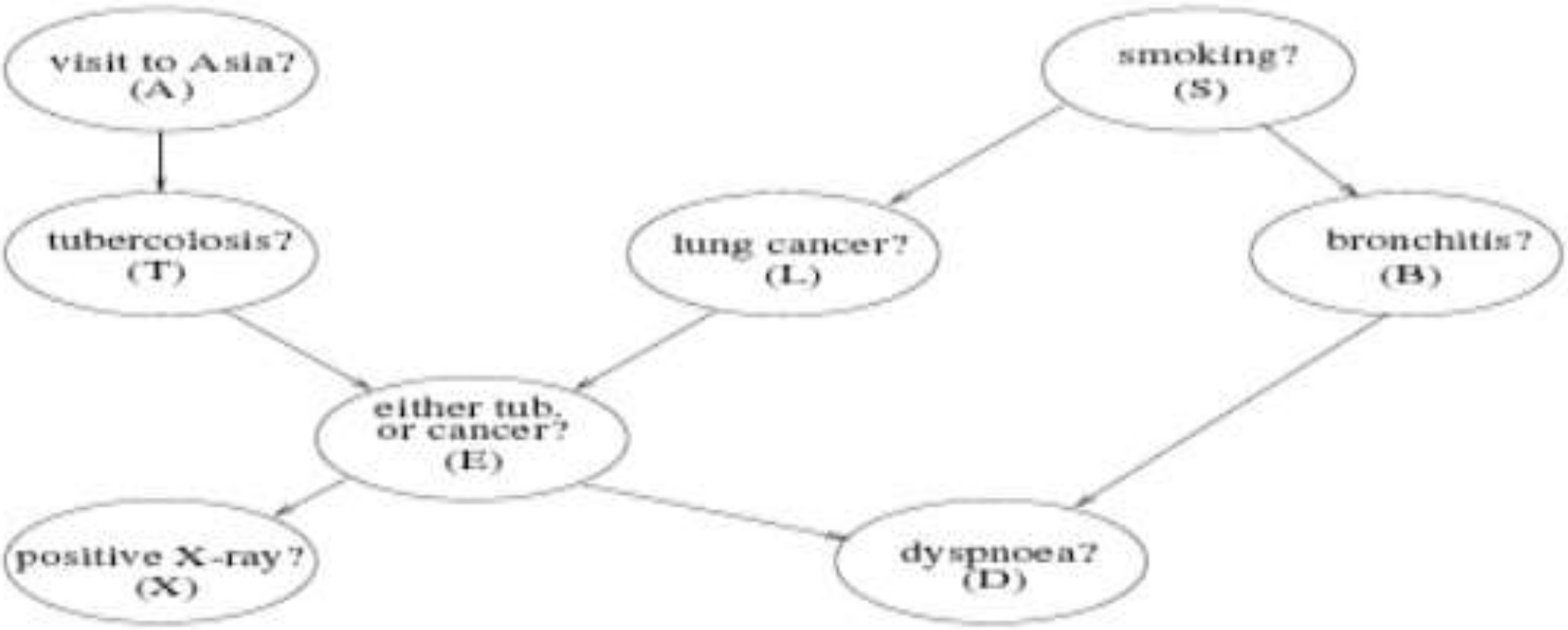
Bayes nets (BN) (also referred to as Probabilistic Graphical Models and Bayesian Belief Networks) are directed acyclic graphs (DAGs) where each node represents a random variable.

The intuitive meaning of an arrow from a parent to a child is that the parent directly influences the child. These influences are quantified by conditional probabilities.

BNs are graphical representations of joint distributions.

# Bayesian Networks

The BN for the medical expert system mentioned previously represents a joint distribution over 8 binary random variables {A,T,E,L,S,B,D,X}.



# Bayesian Networks

Conditional Probability Tables:

Each node in a Bayesian net has an associated conditional probability table or CPT. (Assume all random variables have only a finite number of possible values).

This gives the probability values for the random variable at the node conditional on values for its parents. Here is a part of one of the CPTs from the medical expert system network.

$$P(D = t|E = t, B = t) = 0.9 \qquad P(D = t|E = t, B = f) = 0.7$$
$$P(D = t|E = f, B = t) = 0.8 \qquad P(D = t|E = f, B = f) = 0.1$$

# Bayesian Networks

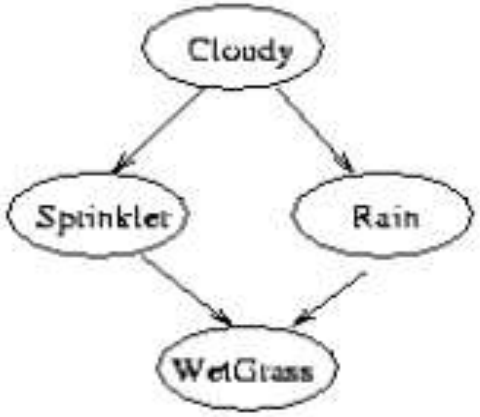
If a node has no parents, then the CPT reduces to a table giving the marginal distribution on that random variable.

$$P(A = t) = 0.1$$
$$P(A = f) = 0.9$$

# Bayesian Networks

Consider another example, in which all nodes are binary, i.e., have two possible values, which we will denote by T (true) and F (false).

| P(C=F) | P(C=T) |
|---|---|
| 0.5 | 0.5 |




| C | P(S=F) | P(S=T) |
|---|---|---|
| F | 0.5 | 0.5 |
| T | 0.9 | 0.1 |

| C | P(R=F) | P(R=T) |
|---|---|---|
| F | 0.8 | 0.2 |
| T | 0.2 | 0.8 |

| S | R | P(W=F) | P(W=T) |
|---|---|---|---|
| F | F | 1.0 | 0.0 |
| T | F | 0.1 | 0.9 |
| F | T | 0.1 | 0.9 |
| T | T | 0.01 | 0.99 |

# Bayesian Networks

We see that the event "grass is wet" (W=true) has two possible causes:

- either the water sprinker is on (S=true)
- or it is raining (R=true).

The strength of this relationship is shown in the table.

For example, we see that Pr(W=true | S=true, R=false) = 0.9 (second row), and hence, Pr(W=false | S=true, R=false) = 1 - 0.9 = 0.1

# Bayesian Networks

since each row must sum to one. Since the C node has no parents, its CPT specifies the prior probability that it is cloudy (in this case, 0.5).

Think of C as representing the season: if it is a cloudy season, it is less likely that the sprinkler is on and more likely that the rain is on.

# Bayesian Networks

**Semantics of Bayesian Networks:**

The simplest conditional independence relationship encoded in a Bayesian network can be stated as follows:

A node is independent of its ancestors given its parents, where the ancestor/parent relationship is with respect to some fixed topological ordering of the nodes.

# Bayesian Networks

**Semantics of Bayesian Networks:**

In the sprinkler example above, by the chain rule of probability, the joint probability of all the nodes in the graph previously is:

$P(C, S, R, W) = P(C) * P(S|C) * P(R|C,S) * P(W|C,S,R)$ By

using conditional independence relationships, we can rewrite this as:

$P(C, S, R, W) = P(C) * P(S|C) * P(R|C) * P(W|S,R)$

# Bayesian Networks

**Semantics of Bayesian Networks:**

where we were allowed to simplify the third term because R is independent of S given its parent C, and the last term because W is independent of C given its parents S and R.

We can see that the conditional independence relationships allow us to represent the joint more compactly.

If we had n binary nodes, the full joint would require O(2^n) space to represent, but the factored form would require O(n 2^k) space to represent, where k is the maximum fan-in of a node. And fewer parameters makes learning easier.

# Bayesian Networks

**Semantics of Bayesian Networks:**

The intuitive meaning of an arrow from a parent to a child is that the parent directly influences the child. The direction of this influence is often taken to represent casual influence.

The conditional probabilities give the strength of causal influence. A 0 or 1 in a CPT represents a deterministic influence.

$$P(E = t|T = t, C = t) = 1 \qquad P(E = t|T = t, L = f) = 1$$
$$P(E = t|T = f, L = t) = 1 \qquad P(E = t|T = f, L = f) = 0$$

# Bayesian Networks

**Decomposing Joint Distributions:**

A joint distribution can always be broken down into a product of conditional probabilities using repeated applications of the product rule.

$$P(A,T,E,L,S,B,D,X) = P(X|A,T,E,L,S,B,D)P(D|A,T,E,L,S,B)$$
$$P(B|A,T,E,L,S)P(S|A,T,E,L)P(L|A,T,E)P(E|A,T)P(T|A)P(A)$$

We can order the variables however we like:

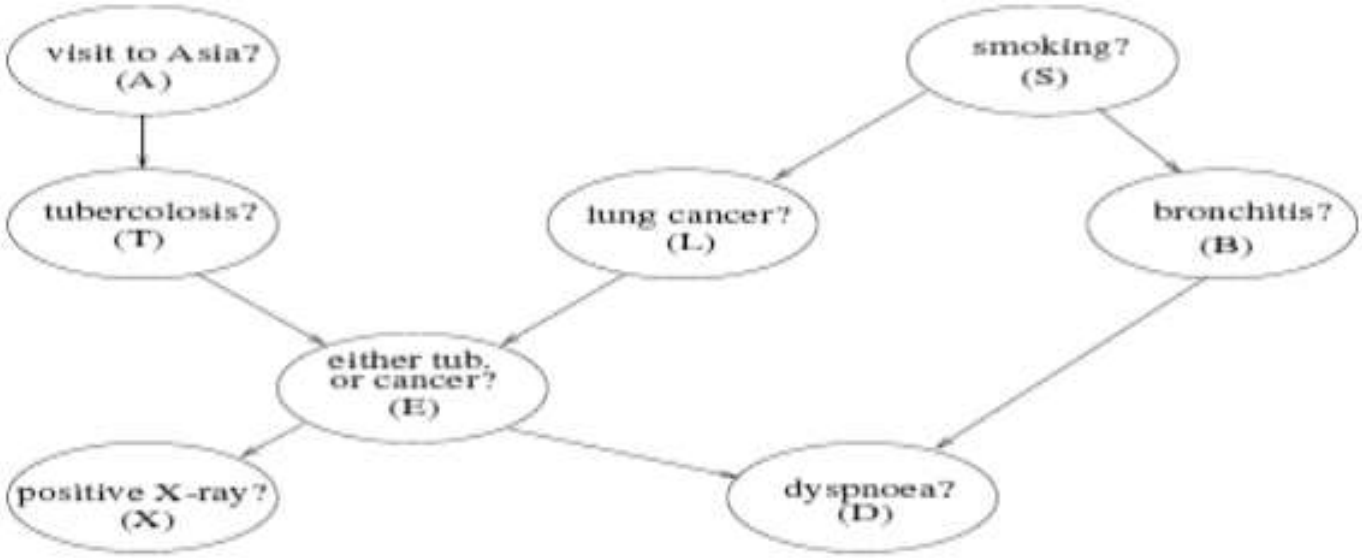
$$P(A,T,E,L,S,B,D,X) = P(X|A,T,E,L,S,B,D)P(D|A,T,E,L,S,B)$$
$$P(E|A,T,L,S,B)P(B|A,T,L,S)P(L|A,T,S)P(S|A,T)P(T|A)P(A)$$

# Bayesian Networks

**Conditional Independence in Bayes Net:**

A Bayes net represents the assumption that each node is con its non-descendants given its parents.

$$P(E|A,T,L,S,B) = P(E|T,L)$$




Note that, a node is NOT independent of its descendants given its parents. Generally,

$$P(E|A,T,L,S,B,X) \neq P(E|T,L)$$

# Bayesian Networks

**Variable ordering in Bayes Net:**

The conditional independence assumptions expressed by a Bayes net allow a compact representation of the joint distribution.

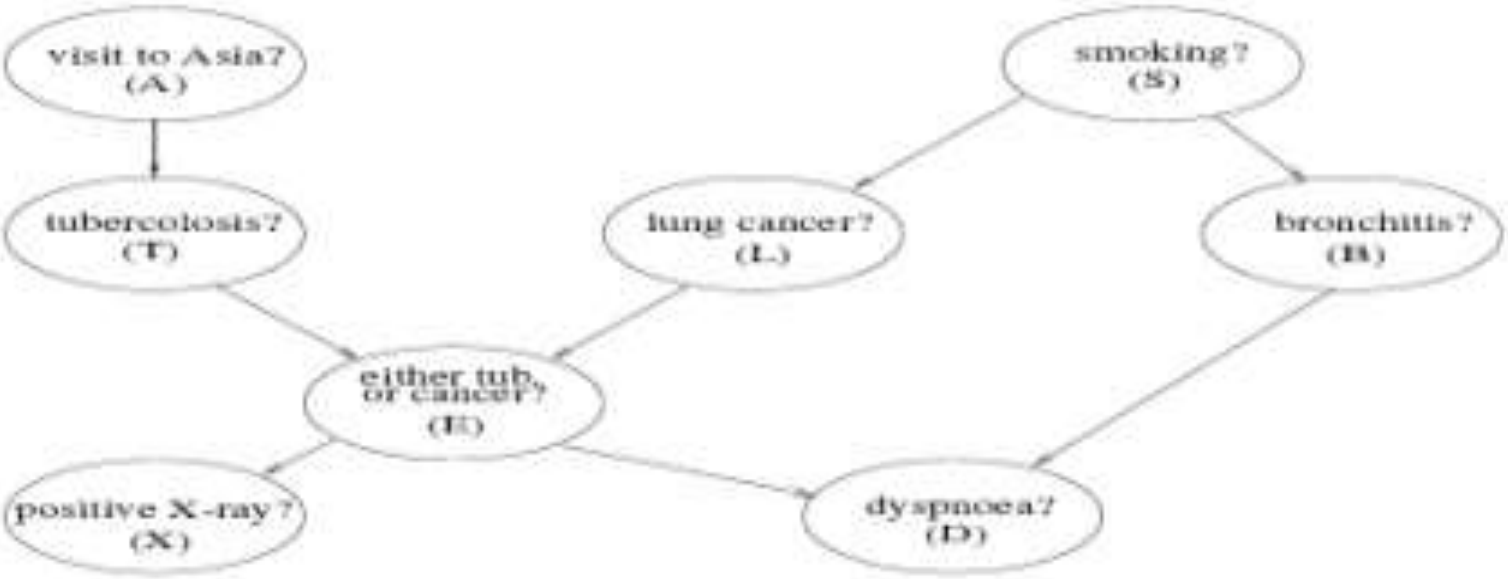
First note that the Bayes net imposes a partial order on nodes: X <= Y iff X is a descendant of Y.

We can always break down the joint so that the conditional probability factor for a node only has non-descendants in the condition.

# Bayesian Networks

**Variable ordering in Bayes Net:**

$$P(A,T,E,L,S,B,D,X) = P(X|A,T,E,L,S,B,D)P(D|A,T,E,L,S,B)$$
$$P(E|A,T,L,S,B)P(B|A,T,L,S)P(L|A,T,S)P(S|A,T)P(T|A)P(A)$$



# Limitations of Naive Bayesian system

The main limitation of Naive Bayes is the assumption of independent predictor features.

Naive Bayes implicitly assumes that all the attributes are mutually independent.

In real life, it’s almost impossible that we get a set of predictors that are completely independent or one another.

# Limitations of Naive Bayesian system

If a categorical variable has a category in the test dataset, which was not observed in training dataset, then the model will assign a 0 (zero) probability and will be unable to make a prediction.

This is often known as Zero Frequency. To solve this, we can use a smoothing technique.

# Inference with BBN (Bayesian Belief Network)

The basic inference problem in BNs is described as follows:

Given

1. A Bayesian network BN
2. Evidence e - an instantiation of some of the variables in BN (e can be empty)
3. A query variable Q

Compute P(Q|e) - the (marginal) conditional distribution over Q

# Inference with BBN (Bayesian Belief Network)

Given what we do know, compute distribution over what we do not. Four categories of inferencing tasks are usually encountered.

1. Diagnostic Inferences (from effects to causes) Given that John calls, what is the probability of burglary? i.e. Find P(B|J)
2. Causal Inferences (from causes to effects) Given Burglary, what is the probability that John calls, i.e. P(J|B) Mary calls, i.e. P(M|B)

# Inference with BBN (Bayesian Belief Network)

3. Intercausal Inferences (between causes of a common event) Given alarm, what is the probability of burglary? i.e. P(B|A) Now given Earthquake, what is the probability of burglary? i.e. P(B|AE)

4. Mixed Inferences (some causes and some effects known) Given John calls and no Earth quake, what is the probability of Alarm

# Inference with BBN (Bayesian Belief Network)

For inferencing procedure for BNs. As an example consider the following linear BN without any apriori evidence:

$$A \rightarrow B \rightarrow C \rightarrow D$$

Consider computing all the marginals (with no evidence). P(A) is given, and

$$P(B) = \sum_A P(B|A)P(A)$$

We don't need any conditional independence assumption for this. For example, suppose A, B are binary then we have

# Inference with BBN (Bayesian Belief Network)

For example, suppose A, B are binary then we have

$$P(B = t) = P(B = t|A = t)P(A = t) + P(B = t|A = f)P(A = f)$$

Now,

$$P(C) = \sum_B P(C|B)P(B)$$

P(B) (the marginal distribution over B) was not given originally. . . but we just computed it in the last step, so we're OK (assuming we remembered to store P(B) somewhere).

# Inference with BBN (Bayesian Belief Network)

If C were not independent of A given B, we would have a CPT for P(C|A,B) not P(C|B).

Note that we had to wait for P(B) before P(C) was calculable.

If each node has k values, and the chain has n nodes this algorithm has complexity $O(nk^2)$.

Summing over the joint has complexity $O(k^n)$.

# Inference with BBN (Bayesian Belief Network)

Complexity can be reduced by more efficient summation by "pushing sums into products".

$$
\begin{aligned}
P(D) &= \sum_{A,B,C} P(A,B,C,D) \\
&= \sum_{A,B,C} P(A)P(B|A)P(C|B)P(D|C) && \text{Conditional independence} \\
&= \sum_{C}\sum_{B}\sum_{A} P(A)P(B|A)P(C|B)P(D|C) && \text{Commutativity of addition} \\
&= \sum_{C} P(D|C) \sum_{B} P(C|B) \sum_{A} P(A)P(B|A) && xy + xz = x(y+z)
\end{aligned}
$$

# Inference with BBN (Bayesian Belief Network)

**Dynamic programming** may also be used for the problem of exact inferencing in the above Bayes Net. The steps are as follows:

1. We first compute

$$f_1(B) = \sum_A P(A)P(B|A)$$

2. $f_1(B)$ is a function representable by a table of numbers, one for each possible value of B.

# Inference with BBN (Bayesian Belief Network)

3. Here,

$$f_1(B) = P(B)$$

4. We then use $f_1(B)$ to calculate $f_2(C)$ by summation over B

This method of solving a problem (ie finding P(D)) by solving sub problems and storing the results is characteristic of dynamic programming.

# Inference with BBN (Bayesian Belief Network)

The above methodology may be generalized. We eliminated variables starting from the root, but we dont have to. We might have also done the following computation.

$$
\begin{aligned}
& P(A,E) \\
&= \sum_B \sum_C \sum_D P(A,B,C,D,E) \\
&= \sum_B \sum_C \sum_D P(A)P(B|A)P(C|B)P(D|C)P(E|D) \\
&= P(A) \sum_B P(B|A) \sum_C P(C|B) \sum_D P(D|C)P(E|D) \\
&= P(A) \sum_B P(B|A) \sum_C P(C|B) f_1(C,E) \\
&= P(A) \sum_B P(B|A) f_2(B,E) \\
&= P(A) f_3(A,E)
\end{aligned}
$$

# Inference with BBN (Bayesian Belief Network)

The algorithm computes intermediate results which are not individual probabilities, but entire tables such as $f_1(C,E)$.

It so happens that $f_1(C,E) = P(E|C)$ but there are also cases where the intermediate tables do not represent probability distributions.

# Dempster-Shafer Theory

- Designed to deal with the distinction between uncertainty and ignorance.
- We use a belief function Bel(X) – probability that the evidence supports the proposition.
- When we do not have any evidence about X, we assign we assign Bel(X) = 0 as well as Bel(¬X) = 0

# Dempster-Shafer Theory

For example, if we do not know whether a coin is fair, then:

Bel( Heads ) = Bel( ¬Heads ) = 0

If we are given that the coin is fair with 90% certainty, then:

Bel( Heads ) = 0.9 X 0.5 = 0.45

Bel(¬Heads ) = 0.9 X 0.5 = 0.45

Note that we still have a gap of 0.1 that is not accounted for by the evidence.

# Dempster-Shafer Theory

Rather than computing the probability of a proposition it computes the probability that evidence supports the proposition.

*Applicability of Dempster-Shafer:*

- Assume lack of sufficient data to accurately estimate the prior and conditional probabilities to use Bayes rule.

- Incomplete model -> Rather than estimating probabilities it uses belief intervals to estimate how close the evidence is to determining the truth of a hypothesis.

# Dempster-Shafer Theory

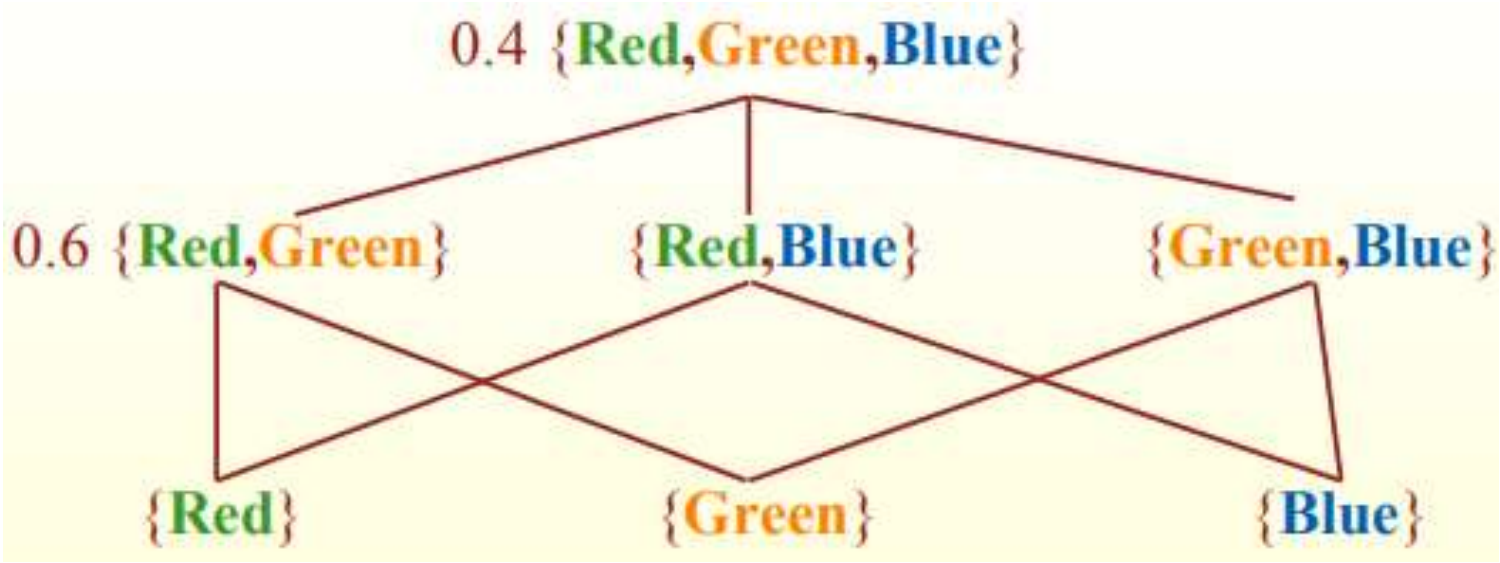
For example, suppose we are informed that one of three terrorist groups, A, B or C has planted a bomb in a building.

- We may have some evidence the group C is guilty, P ( C ) =0.8
- We would not want to say the probability of the other two groups being guilty is .1
- In traditional theory, forced to regard belief and disbelief as functional opposites p(a) + p(not a) = 1 and to distribute an equal amount of the remaining probability to each group.

# Dempster-Shafer Theory

- Allows representation of ignorance about support provided by evidence.
- Allows reasoning system to be skeptical.
- Dempster-Shafer allows you to leave relative beliefs unspecified.

# Dempster-Shafer Theory




Suppose that the evidence supports {red,green} to the degree 0.6.

The remaining support will be assigned to {red,green,blue} while a Bayesian model assumes that the remaining support is assigned to the negation of the hypothesis (or its complement) {blue}.

# Dempster-Shafer Theory

- Given a population F =(blue,red, green) of mutually exclusive elements, exactly one of which (f) is true.

- A basic probability assignment (m) assigns a number in [0,1] to every subset of F such that the sum of the numbers is 1.

- This is Mass as a representation of evidence support.

# Dempster-Shafer Theory

There are $2^{|F|}$ propositions, corresponding to

"the true value of f is in subset A". They are:

(blue), (red), (green),

(blue,red), (blue,green), (red,green),

(red,blue,green), (empty set)

A belief in a subset entails belief in subsets containing that subset.

Belief in (red) entails Belief in:

(red,green), (red,blue), (red,blue,green)

# Overview of Fuzzy Logic

The notion central to fuzzy systems is that truth values (in fuzzy logic) or membership values (in fuzzy sets) are indicated by a value on the range [0.0, 1.0].

0.0 represents absolute Falseness and 1.0 representing absolute Truth.

For example, let us take the statement:

"Jane is old."

If Jane's age was 75, we might assign the statement the truth value of 0.80.

# Overview of Fuzzy Logic

The statement could be translated into set terminology as follows:

"Jane is a member of the set of old people."

This statement would be rendered symbolically with fuzzy sets as:

mOLD(Jane) = 0.80

Where m is the membership function, operating in this case on the fuzzy set of old people, which returns a value between 0.0 and 1.0.

# Overview of Fuzzy Logic

Example:

Probabilistic approach:

"*There is an 80% chance that Jane is old*“

Fuzzy terminology:

*“Jane's degree of membership within the set of old people is 0.80*“

The semantic difference is significant:

# Overview of Fuzzy Logic

Distinction between fuzzy systems and probability:

Both operate over the same numeric range, and at first glance both have similar values:

0.0 representing False (or non- membership),

and 1.0 representing True (or membership).

However, the probabilistic approach yields the natural-language statement while the fuzzy terminology corresponds to the membership within a set.

# Overview of Fuzzy Logic

The next step in establishing a complete system of fuzzy logic is to define the operations of:

EMPTY,

EQUAL,

COMPLEMENT (NOT),

CONTAINMENT,

UNION (OR),

and INTERSECTION (AND).

The formal definitions for these operations are as follows:

# Overview of Fuzzy Logic

Definition 1:

Let X be some set of objects, with elements noted as x. Thus, X = {x}.

Definition 2:

A fuzzy set A in X is characterized by a membership function mA(x) which maps each point in X onto the real interval [0.0, 1.0].

As mA(x) approaches 1.0, the "grade of membership" of x in A increases.

# Overview of Fuzzy Logic

Definition 3:

A is EMPTY iff for all x, mA(x) = 0.0.

Definition 4:

A = B iff for all x: mA(x) = mB(x) [or, mA = mB].

Definition 5:

mA' = 1 - mA.

Definition 6:

A is CONTAINED in B iff mA <= mB.

# Overview of Fuzzy Logic

Definition 7:

C = A UNION B, where: mC(x) = MAX(mA(x), mB(x)).

Definition 8:

C = A INTERSECTION B where: mC(x) = MIN(mA(x), mB(x)).

It is important to note the last two operations, UNION (OR) and INTERSECTION (AND), which represent the clearest point of departure from a probabilistic theory for sets to fuzzy sets.

Operationally, the differences are as follows:

# Overview of Fuzzy Logic

For independent events, the probabilistic operation for AND is multiplication, which is counterintuitive for fuzzy systems.

For example: let us presume that x = Bob, S is the fuzzy set of smart people, and T is the fuzzy set of tall people.

Then, if mS(x) = 0.90 and uT(x) = 0.90,

the probabilistic result would be:

mS(x) * mT(x) = 0.81

whereas the fuzzy result would be:

MIN(uS(x), uT(x)) = 0.90

# Overview of Fuzzy Logic

The probabilistic calculation yields a result that is lower than either of the two initial values, which when viewed as "the chance of knowing" makes good sense.

However, in fuzzy terms the two membership functions would read something like

"Bob is very smart" and "Bob is very tall.“

If we presume for the sake of argument that "very" is a stronger term than "quite," and that we would correlate "quite" with the value 0.81, then the semantic difference becomes obvious.

# Overview of Fuzzy Logic

The probabilistic calculation would yield the statement:

If Bob is very smart, and Bob is very tall, then Bob is a quite tall, smart person.

The fuzzy calculation, however, would yield:

If Bob is very smart, and Bob is very tall, then Bob is a very tall, smart person.

# Non monotonic reasoning

In Non-monotonic reasoning, some conclusions may be invalidated if we add some more information to our knowledge base.

Logic will be said as non-monotonic if some conclusions can be invalidated by adding more knowledge into our knowledge base.

Non-monotonic reasoning deals with incomplete and uncertain models.

"Human perceptions for various things in daily life, "is a general example of non-monotonic reasoning.

# Non monotonic reasoning

Example:

Let suppose the knowledge base contains the following knowledge:

Birds can fly

Penguins cannot fly

Pitty is a bird

So from the above sentences, we can conclude that Pitty can fly.

However, if we add one another sentence into knowledge base "Pitty is a penguin", which concludes "Pitty cannot fly", so it invalidates the above conclusion.

# Non monotonic reasoning

Advantages of Non-monotonic reasoning:

- For real-world systems such as Robot navigation, we can use non-monotonic reasoning.
- In Non-monotonic reasoning, we can choose probabilistic facts or can make assumptions.

Disadvantages of Non-monotonic Reasoning:

- In non-monotonic reasoning, the old facts may be invalidated by adding new sentences.
- It cannot be used for theorem proving.

# Truth maintenance systems (TMS)

A truth maintenance system, or TMS, is a knowledge representation method for representing both beliefs and their dependencies.

An algorithm called the "truth maintenance algorithm" that manipulates and maintains the dependencies.

The name truth maintenance is due to the ability of these systems to restore consistency.

Truth Maintenance Systems (TMS), also called Reason Maintenance Systems.

# TMS Characteristics

*1. Justification-Based Truth Maintenance System (JTMS):*

- It is a simple TMS where one can examine the consequences of the current set of assumptions.
- The meaning of sentences is not known.

*2. Assumption-Based Truth Maintenance System (ATMS):*

- It allows to maintain and reason with a number of simultaneous, possibly incompatible, current sets of assumption.
- Otherwise it is similar to JTMS, i.e. it does not recognize the meaning of sentences.

# TMS Characteristics (continued)

*3. Logical-Based Truth Maintenance System (LTMS):*

- Like JTMS in that it reasons with only one set of current assumptions at a time.
- More powerful than JTMS in that it recognizes the propositional semantics of sentences.
- LTMS understands the relations between p and ~p, p and q and p&q, and so on.

# Dependency driven backtracking

The justification of a sentence, provides the natural indication of what assumptions need to be changed if we want to invalidate that sentence.

Our belief [by "our belief" we mean the "inference-engine's belief"] about a sentence can be any one of the following:

- *False:* the sentence is believed to be unconditionally false; this is also called a contradiction.

- *True:* the sentence is believed unconditionally true; this is also called a premise.

- *Assumed-true:* the sentence is assumed true [we may change our belief later].

# Dependency driven backtracking

- *Assumed-false:* The sentence is assumed false [we may change our belief later]; this is also called a retracted assumption

- *Assumed:* The sentence is believed by inference from other sentences

- *don't-care:* We say a sentence in in if true or assumed-true; it is out if false, assumed-false, or don't-care.

A TMS maintains a Dependency Network.