UNIT -V Expert Sytems

Expert system and Applications: Introduction, phases in building expert systems, expert system architecture, expert system versus traditional systems, rule-based expert systems, application of expert systems, list of shells and tools. Uncertainty measure: probability theory: Introduction, probability theory, Bayesian belief networks, certainty factor theory, dempster- shafer theory.

Expert Systems (ES):

- Expert systems are knowledge based programs which provide expert quality solutions to the problems in specific domain of applications.
- The core components of expert system are
  - knowledge base and
  - navigational capability (inference engine)
- Generally its knowledge is extracted from human experts in the domain of application by knowledge Engineer.
  - Often based on useful thumb rules and experience rather than absolute certainties.
- A process of gathering knowledge from domain expert and codifying it according to the formalism is called knowledge engineering.

Phases in building Expert System

- There are different interdependent and overlapping phases in building an expert system as follows:
- ***Identification Phase*:**
  - Knowledge engineer finds out important features of the problem with the help of domain expert (human).
  - He tries to determine the type and scope of the problem, the kind of resources required, goal and objective of the ES.
- ***Conceptualization Phase*:**
  - In this phase, knowledge engineer and domain expert decide the concepts, relations and control mechanism needed to describe a problem solving.

- ***Formalization Phase:***
  - It involves expressing the key concepts and relations in some framework supported by ES building tools.
  - Formalized knowledge consists of data structures, inference rules, control strategies and languages for implementation.
- ***Implementation Phase:***
  - During this phase, formalized knowledge is converted to working computer program initially called prototype of the whole system.
- ***Testing Phase:***
  - It involves evaluating the performance and utility of prototype systems and revising it if need be. Domain expert evaluates the prototype system and his feedback help knowledge engineer to revise it.

Expert System Architecture

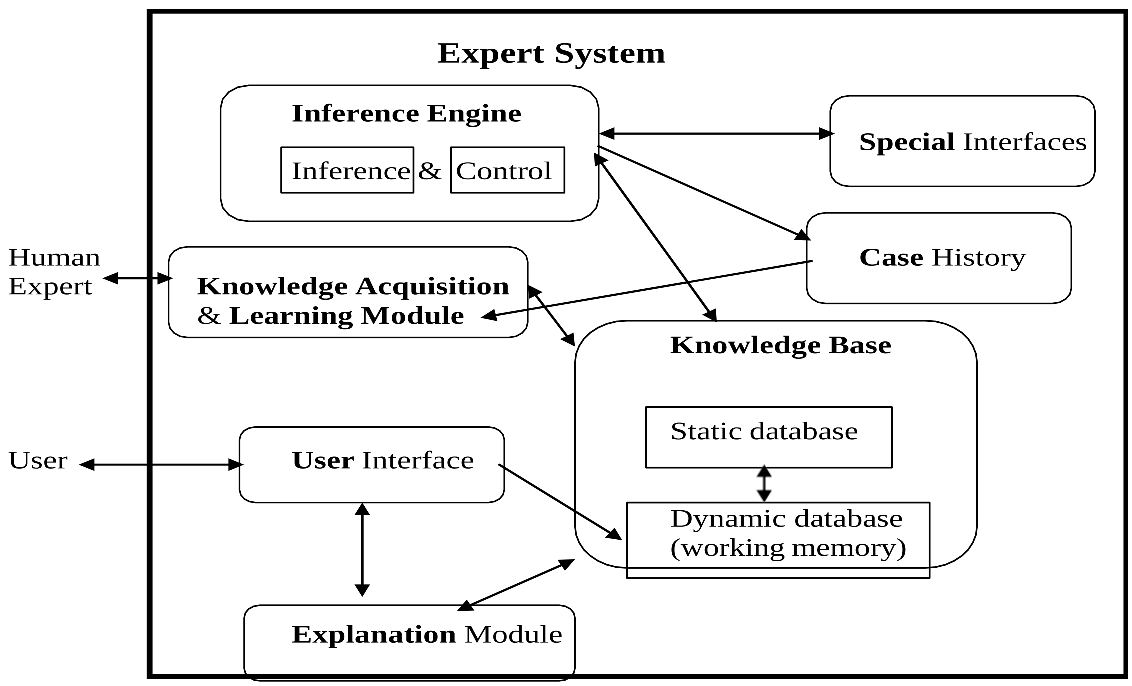


Fig: Architecture of Expert System

Knowledge Base (KB)

- KB consists of knowledge about problem domain in the form of static and dynamic databases.
- Static knowledge consists of
  - rules and facts which is complied as a part of the system and does not change during execution of the system.
- Dynamic knowledge consists of facts related to a particular consultation of the system.
  - At the beginning of the consultation, the dynamic knowledge base often called working memory is empty.
  - As a consultation progresses, dynamic knowledge base grows and is used along with static knowledge in decision making.

Working memory is deleted at the end of consultation of the system

Inference Engine

- It consists of inference mechanism and control strategy.
- Inference means search through knowledge base and derive new knowledge.
- It involve formal reasoning involving matching and unification similar to the one performed by human expert to solve problems in a specific area of knowledge.
- Inference operates by using modus ponen rule.
- Control strategy determines the order in which rules are applied.
- There are mainly two types of control mechanism viz., forward chaining and backward chaining.

Knowledge Acquisition

- Knowledge acquisition module allows system to acquire knowledge about the problem domain.
- Sources of Knowledge for ES

- text books, reports, case studies,
- empirical data and
- domain expert experience.

- Updation of Knowledge can be done using knowledge acquisition module of the system.
  - insertion,
  - deletion and
  - updation of existing knowledge

Case History

- Case History stores the file created by inference engine using the dynamic database created at the time of consultation.
- Useful for learning module to enrich its knowledge base.
- Different cases with solutions are stored in Case Base system.
- These cases are used for solving problem using Case Base Reasoning (CBR).

Explanation module

- Most expert systems have explanation facilities that allow the user to ask the system ***why*** it asked some question, and ***how*** it reached to conclusion.
- It contains 'How' and 'Why' modules attached to it.
  - The sub-module 'How' tells the user about the process through w
  - hich system has reached to a particular solution
  - 'Why' sub-module tells that why is that particular solution offered.
- It explains user about the reasoning behind any particular problem solution.
- Questions are answered by referring to the system goals, the rules being used, and any existing problem data.

Explanation Modules Responses

User	**Why module**
System	Do you have running nose?
User	Yes.
System	I am trying to prove that you have cold. Do you have headache?
User	Yes
System	Do you have sore throat?
User	No
System	Then you do not have cold. Do you have fever?
User	Yes
System	Now I am looking at the possibility of measles. Do you have cough?
User	Yes
System	I can infer measles using rule "If symptoms are fever, cough, running_nose, then patient has measles" measles is concluded.
User	**How Module**
System	Since you have fever, running_nose and cough and there is a rule "If symptoms are fever, cough, running_nose, then patient has measles". So measles is concluded for you.

User Interfaces

- Allows user to communicate with system in interactive mode and helps system to create working knowledge for the problem to be solved.

Dialogue Module (User Interface)	
System	Do you have fever?
User	Yes
System	Do you have bad throat?
User	No
System	Do you have cough?
User	Yes
System	Are you suffering from running nose?
User	Yes
System	Are you suffering from headache?
User	No

Special interfaces

- It may be used for specialized activities such as handling uncertainty in knowledge.
- This is a major area of expert systems research that involves methods for reasoning with uncertain data and uncertain knowledge.

- Knowledge is generally incomplete and uncertain.
- To deal with uncertain knowledge, a rule may have associated with it a *confidence factor* or a weight.
- The set of methods for using uncertain knowledge in combination with uncertain data in the reasoning process is called *reasoning with uncertainty*.

Rule Based Expert Systems

- A rule based expert system is one in which knowledge base is in the form of rules and facts.
  - Knowledge in the form of rules and facts is most popular way in designing expert systems.
- It is also called *production system*.
- Example: Suppose doctor gives a rule for measles as follows:

"If symptoms are fever, cough, running_nose, rash and conjunctivitis then patient probably has measles".

- Prolog is most suitable for implementing such systems.

```
hypothesis(measles) :- symptom(fever), symptom(cough),
                       symptom(running_nose),symptom(conjunctivitis),
                       symptom(rash).
```

Simple Medical diagnostic system with dynamic databases:

- The system starts with consultation predicate, that initiates dialog with user to get information about various symptoms.
- Positive and negative symptoms are recorded in dynamic database and **'hypothesis(Disease)'** is satisfied based on stored facts about symptoms.
- If the hypothesis goal is satisfied then the disease is displayed otherwise display 'sorry not able to diagnose'.
- Finally in both the situations, symptom database for a particular user is cleared.

Query:?-consultation.

Medical Consultation System

```
consultation            :-  writeln('Welcome to MC System'),
                            writeln('Input your name),
                            readln(Name),

                            hypothesis(Dis), !,

                            writeln(Name, 'probably has', Dis),
                            clear_consult_facts.

consultation            :-  writeln('Sorry, not able to diagnose'),
                            clear_consult_facts

hypothesis(flu)         :-  symptom(fever),
                            symptom(headache),

                            symptom(body_ache),
                            symptom(sore_throat),
                            symptom(cough),
                            symptom(chills),

                            symptom(running_nose),
                            symptom(conjunctivitis).

symptom(fever)          :-  positive_ symp('Do you have
                            fever(y/n) ?', fever).

symptom(cough)          :-  positive_ symp('Do you have
                            cough (y/n) ?', cough).

symptom(chills)         :-  positive_ symp('Do you have
                            chills (y/n) ?', chills).

positive_ symp(_, X)    :-  positive(X), !.

positive_ symp(Q, X)    :-  not(negative(X)),
                            query(Q, X, R), R = 'y'.

query(Q, X, R)          :-  writeln(Q), readln(R),
                            store(X, R).

store(X, 'y')           :-  asserta(positive(X)).

store(X, 'n')           :-  asserta(negative(X)).
```

clear_consult_facts :- retractall(positive(_)).

clear_consult_facts :- retractall(negative(_)).

Forward Chaining

- Prolog uses backward chaining as a control strategy, but forward chaining can be implemented in Prolog.
- In forward chaining, the facts from static and dynamic knowledge bases are taken and are used to test the rules through the process of unification.
- The rule is said to be fired and the conclusion (head of the rule) is added to the dynamic database when a rule succeeds.
- Prolog rules are coded as facts with two arguments, first argument be left side of rule and second is the list of sub goals in the right side of the rule.
- Represent prolog rule as a fact by **rule_fact** predicate and simple facts by **fact** predicate.
- Consider the following Prolog rules and facts with their corresponding new fact representations.

a:-b⇒rule_fact(a, [b]).

c:-b, e, f.⇒rule_fact(c, [b, e, f]).

b.⇒fact(b).

e.⇒fact(e).

f.⇒fact(f).

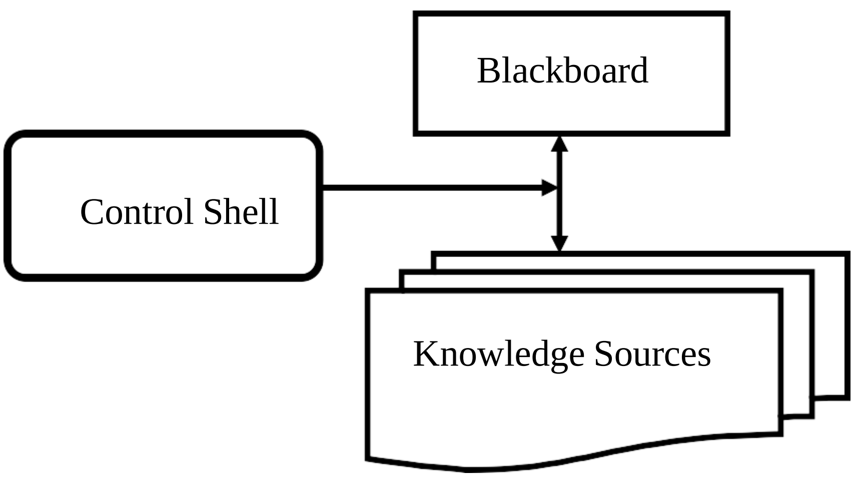
- Here a, b, c, e, f are atoms (predicates with arguments, if any).
    - Newly generated facts are stored in database file 'dfile' which is consulted in the prolog program.

Blackboard System - BS

- Blackboard systems are used to capture dynamic environment with the help of domain experts.

- BS uses a functional modularization of expertise knowledge in the form of Knowledge Sources (KS).
- KS are independent computational modules containing the expert knowledge needed to solve the problem.
  - Blackboard approach has an ability to integrate contributions dynamically for which relationships would be difficult to specify by the KS writer in advance.
- BS consists of three main components viz., Knowledge Sources, Blackboard and Control Shell.
  - BS does not allow direct interaction among modules, as all communication is done via the blackboard through control shell.

System Architecture



Knowledge Source - KS

- KS can be widely diverse in their internal representation and computational techniques and they do not interact directly with each other.
- KS is a specialist at solving certain aspects of the overall application and is separate and independent of all other KSs.
- Once it finds the information it needs on the blackboard, it can proceed without any assistance from other KSs.

- Additional KSs can be added to the blackboard system, existing KS can be upgraded or even can be removed.
- Each KS is aware of its conditions under which it can contribute toward solving the problem.

Blackboard

- The blackboard is a global data repository and shared data structure available to all KSs
- It contains raw input data, partial solutions and final solutions, control information, communication medium etc.
- The system can retain the results of problem-solved earlier, thus avoiding re-computing them later.
- Structuring of information on blackboard is important issue.
- It should enable a KS to efficiently inspect the blackboard to see if relevant information is present.

Control Shell

- The control shell directs the problem-solving process by allowing KSs to respond opportunistically to changes made to the blackboard.
- The control shell reports about the kind of events in which each KS is interested in.
- It maintains this triggering information and directly considers the KS for activation whenever that kind of event occurs.

Information Representation on Blackboard

- There are two ways for storing information on blackboard viz.,
  - specialized representation and
  - fully general representation.
- In specialized representation,
  - KSs may only operate on a few classes of blackboard objects.
  - Sharing data by only a few KSs limits the extendibility and scalability of the system.

- In fully general representation, all aspects of blackboard data are understood by all KSs.

There is a trade-off between these two representations

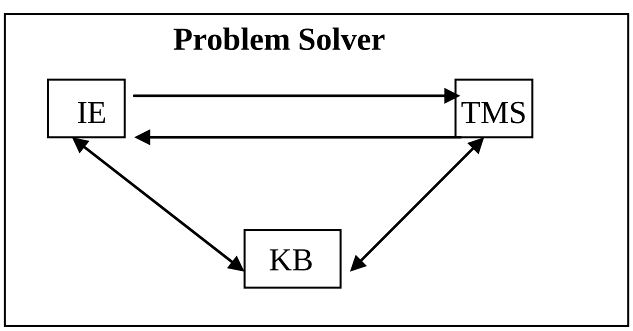
Blackboard System verses Rule Based System

- A blackboard system is different from a rule-based system especially in the size and scope of rules versus the size and complexity of KSs.
- The KSs are substantially larger and more complex than each isomorphic rule in an expert system.
- While expert systems work by firing a rule in response to stimuli, a blackboard system works by executing an entire KS in response to an event.
- A single KS in a blackboard system could be implemented as a complete rule-based system.

Truth Maintenance System (TMS)

- Truth maintenance system (TMS) works with inference engines for solving problems within large search spaces.
- The TMS and inference engine both put together can solve problems where algorithmic solutions do not exist.

TMS maintains the beliefs for general problem solving systems.



- TMS can be used to implement monotonic or non-monotonic systems.
- In monotonic system, once a fact or piece of knowledge is stored in KB, it can not change.

- In monotonic reasoning, the world of axioms continually increases in size and keeps on expending.
- Predicate logic is an example of monotonic form of reasoning. It is a deductive reasoning system where new facts are derived from the known facts.

- Non-monotonic system allows retraction of truths that are present in the system whenever contradictions arise.
  - So number of axioms can both increase and decrease and depending upon the changes in KB, it can be updated.

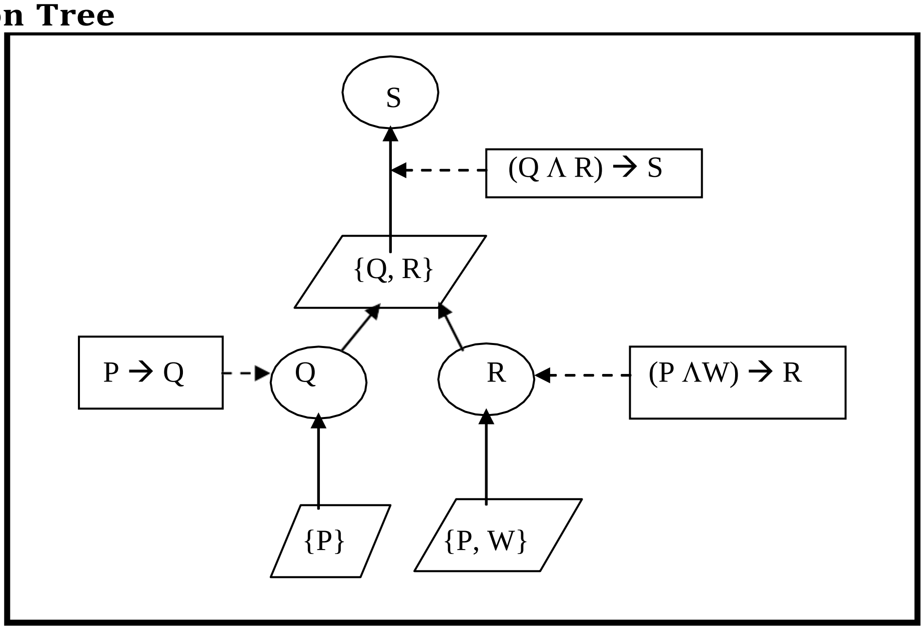
Example - Monotonic TMS

- Suppose we are given the premise set ∑ = {P, W} and the internal constraint set

$$\{P \rightarrow Q, (P \land W) \rightarrow R, (Q \land R) \rightarrow S\}.$$

- TMS are able to derive S from these constraints and the premise set ∑.
- TMS should provide the justifications of deriving S from constraints and premises.
- Therefore, for any given set of internal constraints and premise set ∑, if a formula S can be derived from these, then justification functions generate a justification tree for S.

Justification Tree



Non-Monotonic TMS:

- TMS basically operates with two kinds of objects
  - ‘Propositions’ declaring different beliefs and
  - ‘Justifications’ related to individual propositions for backing up the belief or disbelief expressed by the proposition.
- For every TMS, there are two kinds of justifications required namely ‘Support list’ and ‘Conditional proof’.

Support list (SL):

- It is defined as “SL(IN-node)(OUT-node)”, where IN-node is a list of all IN-nodes (propositions) that support the considered node as true.
  - Here IN means that the belief is true.
  - OUT-node is a list of all OUT nodes for the considered node to be true. OUT means that belief is not true.

Node number	**Facts/assertions**	**Justification (justified belief)**
1	It is sunny	SL(3) (2,4)
2	It rains	SL() ()
3	It is warm	SL(1) (2)
4	It is night time	SL() (1)

Uncertainty

- Most intelligent systems have some degree of uncertainty associated with them.
- Uncertainty may occur in KBS because of the problems with the data.
  - Data might be missing or unavailable.
  - Data might be present but unreliable or ambiguous due to measurement errors, multiple conflicting measurements etc.
  - The representation of the data may be imprecise or inconsistent.
  - Data may just be expert's best guess.
  - Data may be based on defaults and the defaults may have exceptions.
  - Given numerous sources of errors, the most KBS requires the incorporation of some form of uncertainty management.
  - For any form of uncertainty scheme, we must be concerned with three issues.
  - How to represent uncertain data?
  - How to combine two or more pieces of uncertain data?
  - How to draw inference using uncertain data?
  - Probability is the oldest theory with strong mathematical basis.
  - Other methods for handling uncertainty are Bayesian belief network, Certainty factor theory etc.

Probability Theory

- Probability is a way of turning opinion or expectation into numbers.
- It lies between 0 to 1 that reflects the likelihood of an event.
- The chance that a particular event will occur = the number of ways the event can occur divided by the total number of all possible events.

Example: The probability of throwing two successive heads with a fair coin is 0.25

- Total of four possible outcomes are :

  HH, HT, TH & TT

- Since there is only one way of getting HH,

  probability = ¼ = 0.25

Event: Every non-empty subset A (of sample space S) is called an event.

- null set Φ is an impossible event.
- S is a sure event

- P(A) is notation for the probability of an event A.
- P(Φ) = 0 and P(S) = 1
- The probabilities of all events S = {A1, A2, ..., An} must sum up to certainty i.e. P(A1) + … + P(An) = 1
- Since the events are the set, it is clear that all set operations can be performed on the events.
- If A and B are events, then
  - $A \cap B$; $A \cup B$ and A' are also events.
  - A - B is an event "A but not B
  - Events A and B are mutually exclusive, if $A \cap B = \Phi$

Axioms of Probability

- Let S be a sample space, A and B are events.

- $P(A) \geq 0$
- $P(S) = 1$
- $P(A') = 1 - P(A)$
- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$
- If events A and B are mutually exclusive, then

  $P(A \cup B) = P(A) + P(B)$,

- In general, for mutually exclusive events A1,…,An in S

  $P(A1 \cup A2 \cup \ldots \cup An) = P(A1) + P(A2) + \ldots + P(An)$

Joint Probability

- Joint Probability of the occurrence of two independent events is written as P (A and B) and is defined by

  $P(A \text{ and } B) = P(A \cap B) = P(A) * P(B)$

Example: We toss two fair coins separately.

Let $P(A) = 0.5$, Probability of getting Head of first coin

$P(B) = 0.5$, Probability of getting Head of second coin

- Probability (Joint probability) of getting Heads on both the coins is

  $= P(A \text{ and } B)$

  $= P(A) * P(B) = 0.5 \text{ X } 0.5 = 0.25$

- The probability of getting Heads on one or on both of the coins i.e. the union of the probabilities P(A) and P(B) is expressed as

  $P(A \text{ or } B) = P(A \cup B) = P(A) + P(B) - P(A) * P(B)$

  $= 0.5 \text{ X } 0.5 - 0.25$

  $= 0.75$

Conditional Probability

- It relates the probability of one event to the occurrence of another i.e. probability of the occurrence of an event H given that an event E is known to have occurred.
- Probability of an event H (Hypothesis), given the occurrence of an event E (evidence) is denoted by P(H | E) and is defined as follows:

$$P(H \mid E) = \frac{\text{Number of events favorable to H which are also favorable to E}}{\text{No. of events favorable to E}}$$

$$= \frac{P(H \text{ and } E)}{P(E)}$$

- What is the probability of a person to be male if person chosen at random is 80 years old?
- The following probabilities are given
  - Any person chosen at random being male is about 0.50
  - probability of a given person be 80 years old chosen at random is equal to 0.005
  - probability that a given person chosen at random is both male and 80 years old may be =0.002
- The probability that an 80 years old person chosen at random is male is calculated as follows:

  P(X is male | Age of X is 80)

  = [P(X is male and the age of X is 80)] / [P(Age of X is 80)]

  = 0.002 / 0.005 = 0.4

Conditional Probability with Multiple Evidences

- If there are n evidences and one hypothesis, then conditional probability is defined as follows:

$$P(H \mid E1 \text{ and } \ldots \text{ and } En) = \frac{P(H \text{ and } E1 \ldots \text{ and } En)}{P(E1 \text{ and } \ldots \text{ and } En)}$$

Bayes' Theorem

- Bayes theorem provides a mathematical model for this type of reasoning where prior beliefs are combined with evidence to get estimates of uncertainty.
- This approach relies on the concept that one should incorporate the prior probability of an event into the interpretation of a situation.
- It relates the conditional probabilities of events.
- It allows us to express the probability P(H | E) in terms of the probabilities of P(E | H), P(H) and P(E).

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

Proof of Bayes' Theorem

- Bayes' theorem is derived from conditional probability.

Proof: Using conditional probability

$$P(H|E) = P(H \text{ and } E) \,/\, P(E)$$

$$\Rightarrow \quad P(H|E) * P(E) = P(H \text{ and } E) \qquad (1)$$

Also $P(E|H) = P(E \text{ and } H) \,/\, P(H)$

$$\Rightarrow \quad P(E|H) * P(H) = P(E \text{ and } H) \qquad (2)$$

From Eqs (1) and (2), we get

$$P(H|E) * P(E) = P(E|H) * P(H)$$

Hence, we obtain

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

Extension of Bayes' Theorem

- Consider one hypothesis H and two evidences E1 and E2.
- The probability of H if both E1 and E2 are true is calculated by using the following formula:

$$P(H|E1 \text{ and } E2) = \frac{P(E1|H) * P(E2|H) * P(H)}{P(E1 \text{ and } E2)}$$

- Consider one hypothesis H and Multiple evidences E1,...., En.
- The probability of H if E1,..., En are true is calculated by using the following formula:

$$P(H|E1 \text{ and } \ldots \text{ and } En) = \frac{P(E1|H) * \ldots * P(En|H) * P(H)}{P(E1 \text{ and } \ldots \text{ and } En)}$$

- Find whether Bob has a cold (hypotheses) given that he sneezes (the evidence) i.e., calculate P(H | E).
- Suppose that we know / given the following.

P(H) = P (Bob has a cold) = 0.2

P(E | H)= P(Bob was observed sneezing

| Bob has a cold) = 0.75

P(E | ~H)= P(Bob was observed sneezing

| Bob does not have a cold) = 0.2

Now

P(H | E) = P(Bob has a cold | Bob was observed sneezing)

= [P(E | H) * P(H)] / P(E)

- We can compute P(E) as follows:

  P(E) = P(E and H) + P(E and ~H)

  = P(E | H) * P(H) + P(E | ~H) * P(~H)

  = (0.75)(0.2) + (0.2) (0.8) = 0.31

  – Hence P(H | E) = [(0.75 * 0.2)] / 0.31 = 0.48387

  – We can conclude that "Bob's probability of having a cold given that he sneezes" is about 0.5

- Further it can also determine what is his probability of having a cold if he was not sneezing?

  P(H | ~E) = [P(~E | H) * P(H)] / P(~E)

  = [(1 – 0.75) * 0.2] / (1 – 0.31)

  = 0.05 / 0.69 = 0.072

  – Hence "Bob's probability of having a cold if he was not sneezing" is 0.072

Advantages and Disadvantages of Bayesian Approach

Advantages:

- They have sound theoretical foundation in probability theory and thus are currently the most mature of all certainty reasoning methods.
- Also they have well-defined semantics for decision making.

Disadvantages:

- They require a significant amount of probability data to construct a KB.

- For example, a diagnostic system having 50 detectable conclusions (R) and 300 relevant and observable characteristics (S) requires a minimum of 15,050 (R*S + R) probability values assuming that all of the conclusions are mutually exclusive.

- If conditional probabilities are based on
  - statistical data, the sample sizes must be sufficient so that the probabilities obtained are accurate.
  - human experts, then question of values being consistent & comprehensive arise.

- The reduction of the associations between the hypothesis and evidence to numbers also eliminates the knowledge embedded within.
  - The ability to explain its reasoning and to browse through the hierarchy of evidences to hypothesis to a user are lost and

Probabilities in Facts and Rules of Production System

- Some Expert Systems use Bayesian theory to derive further concepts.
  - We know that KB = facts + Rules

- We normally assume that the facts are always completely true but facts might also be probably true.

- Probability can be put as the last argument of a predicate representing fact.

Example:

- a fact "battery in a randomly picked car is 4% of the time dead" in Prolog is expressed as

  battery_dead (0.04).

  - This fact indicates that 'battery is dead' is sure with probability 0.04.

Probability in Rules

- If_then rule in rule-based Systems can incorporate probability as follows:
  - if X is true then Y can be concluded with probability P

Examples:

- Consider the following probable rules and their corresponding Prolog representation.
  - "if 30% of the time when car does not start, it is true that the battery is dead "

    battery_dead (0.3) :- ignition_not_start(1.0).

Here 30% is **rule probability**. If right hand side of the rule is certain, then we can even write above rule as:

battery_dead(0.3) :- ignition_not_start.

  - "the battery is dead with same probability that the voltmeter is outside the normal range"

    battery_dead(P) :-voltmeter_measurment_abnormal(P).

Cumulative Probabilities

- Combining probabilities from the facts and successful rules to get a cumulative probability of the battery being dead is an important issue.
  - We should gather all relevant rules and facts about the battery is dead.
- The probability of a rule to succeed depends on probabilities of sub goals on the right side of a rule.
  - The cumulative probability of conclusion can be calculated by using and-combination.
- In this case, probabilities of sub goals in the right side of rule are multiplied, assuming all the events are independent of each other using the formula

Prob(A and B and C and .....) = Prob(A) * Prob(B) * Prob(C) * ...

- The rules with same conclusion can be uncertain for different reasons.
- If there are more than one rules with the same predicate name having different probabilities, then in cumulative likelihood of the above predicate can be computed by using **or-combination.**

- To get overall probability of predicate, the following formula is used to get 'or' probability if events are mutually independent.

  Prob(A or B or C or ...)

  = 1 - [(1 - Prob(A)) (1 - Prob(B)) (1 - Prob(C)). ....]

Examples

1. "half of the time when a computer does not work, then the battery is dead"

   battery_dead(P):-computer_dead(P1), P is P1*0.5.

   – Here 0.5 is a rule probability.

2. "95% of the time when a computer has electricalproblem and battery is old, then the battery is dead"

   battery_dead(P) :- electrical_prob(P1),
   battery_old(P2), P is P1 * P2 * 0.95.

   – Here 0.95 is a rule probability.

- The rule probability can be thought of hidden and is combined along with associated probabilities in the rule.

Bayesian Belief Network

- Joint probability distribution of two variables A and B are given in the following Table

Joint Probabilities	A	A'
B	0.20	0.12
B'	0.65	0.03

- Joint probability distribution for n variables require 2^n entries with all possible combinations.
- The time and storage requirements for such computations become impractical as n grows.
- Inferring with such large numbers of probabilities does not seem to model human process of reasoning.
- Human tends to single out few propositions which are known to be causally linked when reasoning with uncertain beliefs.

- This leads to the concept of forming belief network called a ***Bayesian belief network***.
- It is a probabilistic graphical model that encodes probabilistic relationships among set of variables with their probabilistic dependencies.
- This belief network is an efficient structure for storing joint probability distribution.

Definition of Bayesian Belief Network:
It is a acyclic (with no cycles) directed graph where the nodes of the graph represent evidence or hypotheses and arc connecting two nodes represents dependence between them.

- If there is an arc from node X to another node Y (i.e., $X \rightarrow Y$), then X is called a *parent* of Y, and Y is a *child* of X.
- The set of parent nodes of a node X_i is represented by parent_nodes(X_i).

Joint Probability of n variables

- Joint probability for 'n' variables (dependent or independent) is computed as follows.
- For the sake of simplicity we write $P(X_1, \ldots, X_n)$ instead of $P(X_1$ and ... and $X_n)$.

$$P(X_1, \ldots, X_n) = P(X_n \mid X_1, \ldots, X_{n-1}) * P(X_1, \ldots, X_{n-1})$$

Or

$$P(X_1, \ldots, X_n) = P(X_n \mid X_1, \ldots, X_{n-1}) * P(X_{n-1} \mid X_1, \ldots, X_{n-2}) * \ldots * P(X_2 \mid X_1) * P(X_1)$$

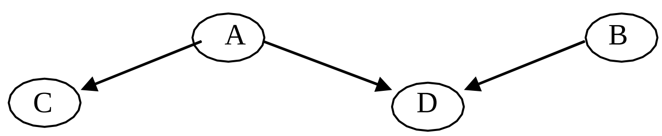
Joint Probability of 'n' Variables using B-Network

- In Bayesian Network, the joint probability distribution can be written as the product of the local distributions of each node and its parents such as:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid \text{parent\_nodes}(X_i))$$

- This expression is reduction of joint probability formula of 'n' variables as some of the terms corresponding to independent variables will not be required.

- If node Xi has no parents, its probability distribution is said to be unconditional and it is written as P(Xi) instead of P(Xi | parent_nodes(Xi)).
- Nodes having parents are called conditional.
- If the value of a node is observed, then the node is said to be an evidence node.
- Nodes with no children are termed as hypotheses node and nodes with no parents are called independent nodes.
- The following graph is a Bayesian belief network.
    - Here there are four nodes with {A, B} representing evidences and {C, D} representing hypotheses.
    - A and B are unconditional nodes and C and D are conditional nodes.



Bayesian Belief Network

To describe above Bayesian network, we should specify the following probabilities.

P(A)	=	0.3
P(B)	=	0.6
P(C|A)	=	0.4
P(C|~A)	=	0.2
P(D|A, B)	=	0.7
P(D|A, ~B)	=	0.4
P(D|~A, B)	=	0.2
P(D|~A, ~B)	=	0.01

- They can also be expressed as conditional probability tables as follows:

Conditional Probability Tables

P(A)	P(B)
0.3	0.6

A	P(C)
T	0.4
F	0.2

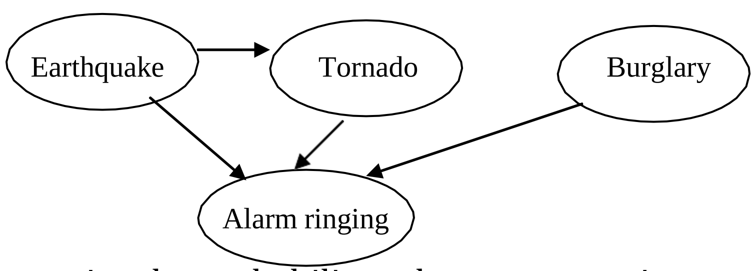
A	B	P(D)
T	T	0.7
T	F	0.4
F	T	0.2
F	F	0.01

- Using Bayesian belief network on previous slide, only 8 probability values in contrast to 16 values are required in general for 4 variables {A, B, C, D} in joint distribution probability.
- Joint probability using Bayesian Belief Network is computed as follows:

$$P(A, B, C, D) = P(D|A, B) * P(C|A) * P(B) * P(A)$$
$$= 0.7 * 0.4 * 0.6 * 0.3 = 0.0504$$

Example of Simple B-Network:

- Suppose that there are three events namely earthquake, burglary or tornado which could cause ringing of alarm in a house.
- This situation can be modeled with Bayesian network as follows.
- All four variables have two possible values T (for true) and F (for false).
  - Here the names of the variables have been abbreviated to *A = Alarm*, *E = Earthquake*, and *B = Burglary* and *T = Tornado*.



- Table contains the probability values representing complete Bayesian belief network. Prior probability of 'earthquake' is 0.4 and if it is earthquake then probability of 'tornado' is 0.8. and if not then the probability of 'earthquake' is 0.5.

Conditional Probability Tables

P(E)	P(B)
0.4	0.7

E	P(Tor)
T	0.8
F	0.5

E	B	Tor	P(A)
T	T	T	1.0
T	T	F	0.9
T	F	T	0.95
T	F	F	0.85
F	T	T	0.89
F	T	F	0.7
F	F	T	0.87
F	F	F	0.3

- The joint probability is computed as follows:

  P(E, B, T, A) = P(A| E, B, T) * P(T|E) * P(E) * P(B)

  = 1.0 * 0.8 * 0.4 * 0.7 = 0.214

- Using this model one can answer questions using the conditional probability formula as follows:
  - "What is the probability that it is earthquake, given the alarm is ringing?" P(E|A)
  - "What is the probability of burglary, given the alarm is ringing?" P(B|A)
  - "What is the probability of ringing alarm if both earthquake and burglary happens?" P(A|E, B)

Advantages of Bayesian Belief Network:

- It can easily handle situations where some data entries are missing as this model encodes dependencies among all variables.
- It is intuitively easier for a human to understand direct dependencies than complete joint distribution.
- It can be used to learn causal relationships.
- It is an ideal representation for combining prior knowledge (which often comes in causal form) and data because the model has both causal and probabilistic semantics.

Disadvantages of Bayesian Belief Network:

- The probabilities are described as a single numeric point value. This can be a distortion of the precision that is actually available for supporting evidence.
- There is no way to differentiate between ignorance and uncertainty. These are distinct two different concepts and be treated as such.
- The quality and extent of the prior beliefs used in Bayesian inference processing are major shortcomings.

- Reliability of Bayesian network depends on the reliability of prior knowledge.

Selecting the proper distribution model to describe the data has a notable effect on the quality of the resulting network. Therefore, selection of the statistical distribution for modeling the data is very important.

Certainty Factor Theory

- Certainty factor theory provides another way of measuring uncertainty by describing a practical way of compromising on pure Bayesian system.
- Certainty factor is based on a number of observations.
- In traditional probability theory, the sum of confidence for a relationship and against a relationship must add up to 1.
- In practical situation, an expert might have some confidence about some relationship being true and have no idea about the relationship being untrue.
- Confidence measures correspond to the informal evaluations that human experts attach to their conclusions, such as 'it is probably or likely true‘.
- The certainty factor is based on 'confidence for' and 'confidence against'
- The MB[H, E] is a **measure of belief** in the range [0, 1] in hypothesis H given the evidence E.
    - If evidence supports it fully then MB[H, E] = 1 and it is zero if the evidence fails to support the hypothesis.
- Similarly, MD[H, E] is a **measure of disbelief** in the range [0, 1] in hypothesis H given the evidence E.
    - It measures the extent to which the evidence E supports the negation of the hypothesis H.
- It is to be noted that MD is not compliment of MB.

Measure of belief

- The measure of belief calculates the relative decrement of disbelief in a given hypothesis H due to some evidence E.
- It may be intuitively defined as follows:

$$MB[H, E] = \frac{(1 - P(H)) - (1 - P(H|E))}{(1 - P(H))}$$

$$= \frac{P(H|E) - P(H)}{(1 - P(H))}$$

In order to avoid getting a negative value of belief, we can modify the above definition to obtain positive value of measure as follows:

$$MB[H, E] = \begin{cases} 1, & \text{if } P(H) = 1 \\ \dfrac{\text{Max}(P(H|E), P(H)) - P(H)}{(1 - P(H))}, & \text{otherwise} \end{cases}$$

Measure of disbelief

- The measure of disbelief (MD) is similarly defined as the relative decrement of belief in a given hypothesis H due to some evidence E. It may be represented as follows:
- It may be intuitively defined as follows:

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

Alternatively,

$$MD[H, E] = \begin{cases} 1, & \text{if } P(H) = 0 \\ \dfrac{P(H) - \text{Min}\{P(H|E), P(H)\}}{P(H)}, & \text{otherwise} \end{cases}$$

Certainty Factor

- Certainty factor is defined as difference of MB and MD.
- Positive certainty factor indicates evidence for the validity of the hypothesis, where evidence implies anything that is used to determine the truth of hypothesis.

– If CF = 1, then the hypothesis is said to be true, while if CF = –1, the hypothesis is considered to be false.

- Moreover, if CF = 0, then there is no evidence regarding whether the hypothesis is true or false.

CF[H, E] = MB[H, E] – MD[H, E], where, $-1 \le CF[H, E] \le 1$.

- For computing CF in general, we need to determine the mechanism for handling the following three cases:
  - Certainty factor when there are two evidences supporting hypothesis H. It is called *incrementally acquired evidence*.
  - Certainty factor for combination of two hypotheses based on the same evidence.
  - Certainty factor for chained rule.

Two Evidences supporting hypothesis

- Case1: Incrementally acquired evidence
- Compute CF(H, E1 and E2).

$$MB[H, E_1 \text{ and } E_2] = \begin{cases} 0, & \text{if } MD[H, E_1 \text{ and } E_2] = 1 \\ MB[H, E_1] + MB[H, E_2] * (1 - MB[H, E_1]), & \text{otherwise} \end{cases}$$

Let us first compute MB(H, E1 and E2) and MD(H, E1 and E2)

- Similarly MD is defined
- Suppose we make an initial observation E1 that confirms our belief in H with MB[H, E1) = 0.4 and MD(H, E1) = 0. Consider second observation E2 that also confirms H with MB[H, E2) = 0.3. Then CF(H, E1) = 0.4

MB(H, E1 and E2) = MB(H, E1) + MB(H, E2) * (1 – MB(H, E1))

= 0.4 + 0.3 * (1-MB(H, E1))

= 0.4 +0.18 = 0.58

and

MD(H, E1 and E2) = 0.0

Therefore,

CF(H, E1 and E2) = 0.58

- Here we notice that slight confirmatory evidence can larger certainty factor.
- *For other two cases refer to textbook.*
    - *Case 2:* There are two hypotheses H1 and H2 based on the same evidence E. Find CF for conjunction and disjunction of hypotheses.

    - *Case 3:* In chained rule, the rules are chained together with the result that the outcome of one rule is input of another rule. For example, if the outcome of an experiment is treated as an evidence for some hypothesis i.e., E1 → E2 → H

Dempster–Shafer Theory

- It is a mathematical theory of evidence.
- It allows one to combine evidence from different sources and arrive at a degree of belief.
- Belief function is basically a generalization of the Bayesian theory of probability.
- Belief functions allow us to base degrees of belief or confidence for one event on probabilities of related events, whereas Bayesian theory requires probabilities for each event.
- These degrees of belief may or may not have the mathematical properties of probabilities.
- The difference between them will depend on how closely the two events are related.
- It also uses numbers in the range [0, 1] to indicate amount of belief in a hypothesis for a given piece of evidence.
- Degree of belief in a statement depends upon the number of answers to the related questions containing the statement and the probability of each answer.

- In this formalism, a degree of belief (also referred to as a mass) is represented as a belief function rather than a Bayesian probability distribution

  Example

- Mary and John are friends.
  - Suppose Mary tells John that his car is stolen. Then John's belief on the truth of this statement will depend on the reliability of Mary. But it does not mean that the statement is false if Mary is not reliable.
  - Assume that probability of John's opinion about the reliability of Mary is given as 0.85. Then the probability of Mary to be unreliable for John is 0.15.
  - So her statement justifies a 0.85 degree of belief that a John's car is stolen and John has no reason to believe that his car is not stolen so it is zero degree of belief that John's car is not stolen.
  - This zero does not mean that John is sure that his car is not stolen as in the case of probability, 0 would mean that John is sure that his car is not stolen. The values 0.85 and the 0 together constitute a belief function.

Dempster Theory Formalism

- Let *U* be the *universal set* of all hypotheses, propositions, or statements under consideration.
  - The power set P(U), is the set of all possible subsets of *U*, including the empty set represented by ϕ.
  - The theory of evidence assigns a belief mass to each subset of the power set.
- A function m: P(U) $\rightarrow$ [0,1] is called a *basic belief assignment* (BBA) function. It satisfies the following axioms:
  - $m(\phi) = 0$; $\sum m(A) = 1, \forall A \in P(U)$
- The value of m(A) is called *mass assigned to A* on the unit interval.

- It makes no additional claims about any subsets of A, each of which has, by definition, its own mass.

Dempster's Rule of Combination

- The original combination rule, known as Dempster's rule of combination, is a generalization of Bayes' rule.
- Assume that *m1* and *m2* are two belief functions used for representing multiple sources of evidences for two different hypotheses.
- Let A, B $\subseteq$ U, such that m1(A) ≠ 0, and m2(B) ≠ 0.
- The Dempster's rule for combining two belief functions to generate an *m3* function may be defined as:

$$m3(\phi) = 0$$

$$m3(C) = \frac{\sum_{A \cap B = C} (m1(A) * m2(B))}{1 - \sum_{A \cap B = \phi} (m1(A) * m2(B))}$$

- This belief function gives new value when applied on the set C = A $\cap$ B.
- The combination of two belief functions is called the *joint mass*.
  - Here m3 can also be written as (m1 o m2).
- The expression [$\sum_{A \cap B = \phi}$ (m1(A) * m2(B))] is called normalization factor.
  - It is a measure of the amount of conflict between the two mass sets.
- The normalization factor has the effect of completely ignoring conflict and attributing any mass associated with conflict to the null set.

Example : Diagnostic System

- Suppose we have mutually exclusive hypotheses represented by a set U = {flu, measles, cold, cough}.
- The goal is to assign or attach some measure of belief to the elements of U based on evidences.

- It is not necessary that particular evidence is supporting some individual element of U but rather it may support subset of U.

- For example, an evidence of 'fever' might support {flu, measles}.

- So a belief function 'm' is defined for all subsets of U.
- The degree of belief to a set will keep on changing if we get more evidences supporting it or not.
- Initially assume that we have no information about how to choose hypothesis from the given set U.
- So assign m for U as 1.0 i.e., m(U) = 1.0
  - This means we are sure that answer is somewhere in the whole set U.
- Suppose we acquire evidence (say fever) that supports the correct diagnosis in the set {flu, measles} with its corresponding 'm' value as 0.8.

Then we get m({flu, measles}) = 0.8 and m(U) = 0.2

- Let us define two belief functions m1 and m2 based on evidence of fever and on evidence of headache respectively as follows:

m1({flu, measles}) = 0.8

m1(U) = 0.2

m2({flu, cold}) = 0.6

m2(U) = 0.4

- We can compute their combination m3 using these values.

Combination of m1 and m2	m2({flu, cold}) = 0.6	m2(U) = 0.4
m1({flu, measles}) = 0.8	m3({flu}) = 0.48	m3({flu, measles}) = 0.32
m1(U) = 0.2	m3({flu, cold}) = 0.12	m3(U) = 0.08

- Now previous belief functions are modified to m3 with the following belief values and are different from earlier beliefs.

m3({flu}) = 0.48

m3({flu, cold}) = 0.12

m3({flu, measles}) = 0.32

m3(U) = 0.08

- Further, if we have another evidence function m4 of sneezing with the belief values as:

  m4({cold, cough}) = 0.7

  m4(U) = 0.3

- Then the combination of m3 and m4 gives another belief function as follows:

Combination of m3 and m4	**m4({cold, cough}) = 0.7**	**m4(U) = 0.3**
m3({flu}) = 0.48	m5(ϕ) = 0.336	m5({flu}) = 0.114
m3({flu, cold)) = 0.12	m5({cold}) = 0.084	m5({flu, cold)) = 0.036
m3({flu, measles})= 0.32	m5(ϕ) = 0.224	m5({flu, measles})= 0.096
m3(U) = 0.08	m5({cold, cough}) = 0.056	m5(U) = 0.024

- If we get empty set (ϕ) by intersection operation, then we have to redistribute any belief that is assigned to ϕ sets proportionately across non empty sets using the value $(1 - \sum A \cap B = \phi \ (m1(A) * m2(B)))$ in the denominator of belief values for non empty sets.

- From the table we get multiple belief values for empty set (ϕ) and its total belief value is 0.56.

- So according to formula, we have to scale down the remaining values of non empty sets by dividing by a factor (1 - 0.56 =0.44).

  m5({flu}) = (0.144/0.44) = 0.327

  m5({cold}) = (0.084/0.44) = 0.191

  m5({flu, cold}) = (0.036/0.44) = 0.082

  m5({flu, measles})= (0.096/0.44) = 0.218

  m5({cold, cough}) = (0.056/0.44) = 0.127

m5(X) = (0.024/0.44) = 0.055

- While computing new belief we may get same subset generated from different intersection process. The 'm' value for such set is computed by summing all such values.

  - given set to functions to suit the application.