UNIT 5

Uncertain Knowledge and Learning

Syllabus

Uncertainty: Acting Under Uncertainty, Basic Probability Notation, Inference Using Full Joint Distributions, Independence, Bayes' Rule and its Use,

Probabilistic Reasoning: Representing Knowledge in an Uncertain Domain, The Semantics of Bayesian Networks, Efficient Representation of Conditional Distributions, Approximate Inference in Bayesian Networks, Relational and First-order Probability, Other Approaches to Uncertain Reasoning; Dempster-Shafer Theory.

Learning: Forms of Learning, Supervised Learning, Learning Decision Trees. Knowledge in Learning: Logical Formulation of Learning, Knowledge in Learning, Explanation-based Learning, Learning Using Relevance Information, Inductive Logic Programming.

Learning Objectives

- ✓ Procedure of Handling Uncertain Knowledge
- ✓ Representation of Probabilities, Method of Probabilistic Inference
- ✓ Property of Independence
- ✓ Baye's Rule and its Application
- ✓ Use of Bayesian Network for Representing Knowledge in Uncertain Domains
- ✓ Construction of Bayesian Network, Exact and Approximate Inference in Bayesian Networks
- ✓ Use of Relational Probability Models
- ✓ Handling Uncertainty using Dempster-Shafer Theory
- ✓ Definition, Forms and Issues in Learning, Learning using Decision Trees
- ✓ Formulation of Hypothesis using Logical Sentences
- ✓ Explanation and Relevance-based Learning.

Introduction

Uncertainty is considered as a prominent task in diagnosis performed using first order logic in order to write the rules for diagnosis. It utilizes probabilistic concepts where probability refers to the extent to which an event is likely to occur. The computation obtained from the observed evidence of posterior probabilities for propositions of a query is known as probabilistic inference. Baye's rule is a simple equation that acts as a basis for probabilistic inference of various updated Artificial Intelligence (AI) systems. It involves two unconditional probabilities and one conditional probability for calculating conditional probability.

Probabilistic reasoning can be defined as a reasoning strategy used to manage the uncertain situations arising in the intelligent systems. It makes use of probabilistic concepts which are applied using reasoning theory. Bayesian network is a data structure represented in the form of graph which is used to describe the dependencies existing between variables. It provides a compact view of overall specifications of joint probability distributions.

If agents improvise their performance for the tasks performed in future then they are said to be in learning state. The major forms of learning are, supervised learning, unsupervised learning, reinforcement learning and semi-supervised learning. One important concept in learning is decision trees. Decision tree is a tool for decision supporting which takes the form of tree structure making possible decisions by performing the set of tests.

PART-A SHORT QUESTIONS WITH SOLUTIONS

Q1. List the three main reasons that cause failure of medical diagnosis.

Model Paper-II, Q1(i)

Answer :

The three main reasons that cause failure of medical diagnosis are as follows,

(i) Lack of Effort

It requires large amount of work to include all the probable sets of subsequents which in turn make a rule hard.

(ii) Lack of Theoretical Knowledge

There is no complete theory for the field of medical science.

(iii) Lack of Practical Knowledge

Even if the rules are identified then also all the essential tests cannot be performed on a specific patient.

Q2. Write about marginalization rule.

Model Paper-I, Q1(i)

Answer :

The process of obtaining the unconditional or marginal probability of a variable by summing up its entities is known as marginalization or summing out. The following is the general marginalization rule for any sets of variables A and B.

$$P(A)=\sum_{B} P(A,B)$$

This means, a distribution over A could be derived by adding up all other variables from any joint distribution containing A. Furthermore, another form of this rule includes conditional probabilities instead of joint probabilities, using the following product rule,

$$P(A)=\sum_{B} P(A/B)P(B)$$

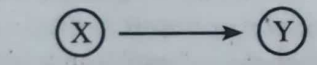
Q3. What are the properties of Bayesian networks?

Answer :

The properties of a Bayesian Network are as follows,

(i) A node in a Bayesian network represents a discrete or a continuous random variable.

(ii) A set of directed arrows/links is used for connecting the nodes within a network i.e., an arrow from node X to Y represents that X is a parent of Y.



Parent Node Child Node

(iii) It is a Directed Acyclic Graph (DAG) as it doesn't contain any directed cycle.

(iv) Each individual node has an associated conditional probability distribution $P(X_i \mid \text{Parents } (X_i))$ that determines the effect of the parents on a particular node.

The arrangement of nodes and their interconnections within the network is independent of conditional relationships. The arrows represent that one node has direct impact on the other. Therefore, these direct relationships can be easily determined in a Bayesian network. When the arrangement of lines and nodes is finalized, the conditional probability can be assigned to the child variables.

Q4. Define,

(a) Conditional Probability Table (CPT)

(b) Conditional Case.

Answer : Model Paper-III, Q1(i)

(a) Conditional Probability Table

This table is used for discrete and mutually dependent random variables to depict conditional probabilities of child nodes derived from its parents.

(b) Conditional Case

It refers to the possible combination of values for the parent nodes. Each row adds upto a sum of '1'. The reason being the entries representing all the possible set of cases for the variables.

Q5. State chain rule.

Answer : Model Paper-II, Q1(j)

The identity for entries in joint distribution can be written in terms of conditional probability using product rule as,

$P(x_1, x_2, ... x_n) = P(x_n|x_{n-1}, ... x_1)\ P(x_{n-1}, ... x_1)$

The process of rewriting is iterated while reducing each conjunctive probability to a conditional probability and a smaller conjunction results in a big product. This identity is called the chain rule.

Mathematically, it can be expressed as,

$P(x_1, ... x_n) = P(x_n \mid x_{n-1}, ..., x_1)\ P(x_{n-1} \mid x_{n-2}, ... x_1) ..... P(x_2 \mid x_1)\ P(x_1)$

$P(X_i \mid X_{i-1} ..... X_1) = P(X_i \mid \text{parents } (X_i))$

Q6. Define deterministic node.

Answer :

A deterministic node is a node whose value is determined by the values of its parents without any uncertainty. The parent-child relationship can either be logical or numerical.

An example of logical relationship is the relationship between the parent nodes India, China, Pakistan and the child node Asia where child is a disjunction of its parents.

An example of numerical relationship is, if the parent nodes represent the price of a specific brand of laptop at different laptop dealers and the child node is the final discounted price paid by the customer, then the child node is the minimum of the values that the parent nodes has.

Q7. List the problems of first order models.

Answer : Model Paper-III, Q1(j)

The problems with first-order models are as follows,

1. The first-order models are infinite.
2. The summation of the possible worlds could be infeasible i.e.,

$$P(\phi) = \sum_{w:\phi \text{is true in } w} P(w) - (\text{Infeasible})$$

3. The specification of a complete and consistent distribution over an infinite set of worlds may be very difficult.

The above problems can be overcome by using Relational Probability Models (RPMs).

Q8. What are the issues considered while designing a learning element?

Answer : Model Paper-I, Q1(j)

The different issues considered while designing a learning element are as follows,

1. Learning of components that are related to performance element.
2. Reading the obtained feedback in order to learn the components.
3. Specifying the representation for the learned components.

PART-B ESSAY QUESTIONS WITH SOLUTIONS

5.1 UNCERTAINTY

5.1.1 Acting Under Uncertainty

Q9. Define uncertainty. Explain the procedure of handling uncertain knowledge.

Answer : Model Paper-I, Q10(a)

Uncertainty

Uncertainty is considered as prominent task in diagnosis wherein first order logic is used in order to write the rules for fever diagnosis. Procedure to handle uncertain knowledge is as follows,

$$\forall P \text{ symptom}(P, \text{fever}) \Rightarrow \text{disease}(P, \text{jaundice})$$

The problem with this rule is that, not all patients with fever have jaundice. There is a possibility that they can have malaria, typhoid, viral or flu. So considering other causes for fever,

$$\forall P \text{ symptom}(P, \text{fever}) \Rightarrow \text{disease}(P, \text{jaundice}) \vee \text{disease}(P, \text{malaria}) \vee \text{disease}(P, \text{typhoid}) \ldots\ldots$$

These possible causes cannot be added to the list of probable reasons in order to make this rule true. Hence, writing the rule in the following manner.

$$\forall P \text{ disease}(P, \text{jaundice}) \Rightarrow \text{symptom}(P, \text{fever})$$

Even this rule is appropriate in same cases i.e., every patient that has jaundice may not have fever. The only method that can be used to make the rule logically exhaustive is to add more information needed for jaundice to cause a fever on the left hand side. Even after augmenting the information, there is a possibility that the patient may have fever and jaundice are not connected.

The three main reasons that cause failure of medical diagnosis are as follows,

(i) Lack of effort/Laziness

(ii) Lack of theoretical knowledge/Theoretical ignorance

(iii) Lack of practical knowledge/Practical ignorance.

(i) Lack of Effort

It requires large amount of work to include all the probable sets of subsequents which in turn make a rule hard.

(ii) Lack of Theoretical Knowledge

There is no complete theory for the field of medical science.

(iii) Lack of Practical Knowledge

Even if the rules are identified then also all the essential tests cannot be performed on a specific patient.

Q10. Discuss in brief about,

(i) Utility theory

(ii) Decision theory

(iii) Decision theoretic agent.

Answer :

i) Utility Theory

Utility theory is used for representing and reasoning with respect to preferences. Basically, preferences refer to the choice hat an agent makes between different possible outcomes of various plans. This theory says that every individual state is assigned degree of utility based on which an agent selects the state. In general, an agent prefers the state that is assigned highest utility. he utility of state is related to the agent whose preferences are to be represented by utility function.

(ii) Decision Theory

Decision theory is the combination of preferences that are expressed by utilities (i.e., utility theorem) and the probabilities present in the general theory of rational decisions. The decision theory applies principle of maximum expected utility which states that an agent is said to be a rational agent, if it selects as action which generates the highest expected utility when compared to all the possible outcomes of action. It is represented as,

Decision theory = Probability theory + Utility theory

(iii) Decision Theoretic Agent

The structure of theoretic agent is similar to the structure of logical agent at an abstract level. However, the major difference is that in former one, the knowledge of the current state is uncertain, whereas in the latter one, the knowledge of current state is certain. In the belief state, decision theoretic agent is a set of probabilities associated with every possible actual state of real world. Whenever a belief state is known, it is possible for an agent to make probabilistic predictions regarding the outcome of the action and therefore can choose the action that yields the highest expected utility.

Algorithm for Decision Theoretic Agent

Function dcsn_theory_agent(P) returns actn

Static bs, actn

Update *bs* depending on *actn* and *P* compute probabilities of action outcome.

Given description of action and current *bs*

choose *actn* with highest expected utility

given outcome probabilities and utility information return *actn*.

Here, actn = action

bs = belief_state

P = Percept.

5.1.2 Basic Probability Notation

Q11. Write short notes on,

(i) Probability

(ii) Evidence.

Answer :

(i) Probability

There are several definitions available to understand probability. Among these, the widely used definitions are as follows,

1. Mathematical Definition/Classical Definition

Consider a random experiment with N mutually exclusive and equally likely events, out of which 'S' events are favorable for a particular event M, then the probability of the occurrence of M is given by,

$$P(M) = \frac{S}{N} = \frac{\text{Number of favourable events with respect to } M}{\text{Total number of events in the experiment}}$$

This probability is also called as *probability of success of M* or *priori probability*.

The probability of failure is denoted as $P(\overline{M})$ and is given by,

$$P(\overline{M}) = \frac{N-S}{N} = 1 - \frac{S}{N} = 1 - P(M)$$

$$\therefore P(\overline{M}) = 1 - P(M)$$

Where, $(N-S)$ outcomes are not favourable for the event M.

Note that the sum of the probabilities is always equal to unity.

$$\therefore P(M)+P(\overline{M})=1$$

In general, the probability of success and probability of failure are denoted by P and Q respectively,

$$P=P(M)=\frac{S}{N} \quad [\because 0 \le P \le 1]$$

$$Q=P(\overline{M})=\frac{N-S}{N} \quad [\because 0 \le Q \le 1]$$

and $P+Q=1$

2. Statistical Definition

Suppose an experiment is repeated 'n' times under essentially identical conditions. Let an event 'A' occurs 'm' times then $\frac{m}{n}$ is defined as the relative frequency of A. The limiting value of the relative frequency of occurrence is called the probability of outcome A, i.e.,

$$P(A)=\lim_{n\to\infty}\left(\frac{m}{n}\right);\ 0 \le \frac{m}{n} \le 1$$

This probability is also called as '*aposteriori probability*', i.e., probability determined after the event.

3. Axiomatic Definition

Probability is a number that is given to every element in a group of events belonging to a random experiment and that satisfies the following properties,

If 'S' is the sample space and 'M' is any event in a random experiment, then the following axioms are satisfied,

(a) $0 \le P(M) \le 1$ for each event M in S

According to this axiom, the probability of each element is a real number during the interval 0 to 1.

(b) $P(S)=1$

According to this axiom, the sample space as a whole has the probability of 1.

(c) If E_1 and E_2 are two mutually exclusive events in S, then

$$P(E_1 \cup E_2)=P(E_1)+P(E_2)$$

According to the third axiom, the probability of the sum of two mutually exclusive events is equal to the sum of their probabilities.

(ii) Evidence

Evidence is defined as percepts that are based on probable assertions. Evidence plays a pivotal role in determining our beliefs that are dependent on the percepts of agents receiving up-to-date information.

Example

Consider an agent who wishes to draw a heart from ordinary deck of playing cards. The agent prior to picking the card assigns a probability of $1/13$ and assumes that the card being selected is an ace of heart. After selecting the card, the probability of propositions can be either true (1) or false (0). Therefore, it can be said that assigning probability to proposition is similar to entailing logical sentence from knowledge base instead of its truth values (true or false).

All probability statements specify the evidence corresponding to the probability that is being estimated. If an agent receives new percepts then its probable estimated value are updated to reflect the new evidence.

Q12. Explain in detail the prior probability distribution.

Answer :

Prior Probability

The degree of beliefs with respect to the absence of any other information of a proposition is referred to as prior or unconditional probability. It is represented as $P(x)$.

Example

If the prior probability that the finger has to be fractured is 0.1 then it can be written as,

$$P(\text{Fracture} = \text{True}) = 0.1 \text{ or } P(\text{Fracture}) = 0.1$$

Characteristics of Prior Probability

1. Prior Probability Distribution

The denotion which represents probabilities of all the possible values of a random variable is prior probability distribution. For example, probability distribution for day is,

$$P(\text{Day}) = \{\text{Sun, Mon, Tue, Wed, Thu, Fri, Sat}\}$$

2. Joint Probability Distribution

It supports joint probability distribution which describe the probability distribution of entire set of random variables of the world. For example, if fracture, handpain and day are the variables of the world then the joint probability distribution will be,

$$P(\text{fracture, handpain, day})$$

It represents a complete specification of one's uncertainty about the world since it specifies the probability of every atomic event. For continuous variables, the joint probability cannot be represented as a table as it contains infinite values. Instead, a value N is used as a parametrized function of N.

3. Probability Density Function

Prior probability also provides the probability distribution for continuous variables which is referred to as probability density function.

Q13. State and explain the conditional probability.

Answer :

Conditional probability is defined as the probability of occurrence of a particular value of one random variable when the other variable has already occurred.

Let 'A' and 'B' be any two events. The probability of occurrence of event B such that A has already occurred is denoted by $P(A|B)$ and is defined as,

$$P(A|B) = \frac{P(A \wedge B)}{P(B)}$$

Interms of product rule, it can be written as,

$$P(A \wedge B) = P(A|B).\ P(B)$$

In general, the implementation and use of prior probability is easy, but most of the people prefer conditional probability as their vehicle for probabilistic inferences.

'P' is also used to notify the conditional distributions. Generally,

$$P(M|N) = P(M = m_i|N = n_j) \text{ for every } i, j.$$

The probability 'P' can be made more compact by applying the product rule to every case where A is assert particular value of m and B is assert particular value of n. Then,

$$P(M = m_1 \wedge N = n_1) = P(M = m_1 \mid N = n_1)\ P(N = n_1)$$

$$P(M = m_1 \wedge N = n_2) = P(M = m_1 \mid N = n_2)\ P(N = n_2)$$

Then, it can be combined as,

$$P(M, N) = P(M|N)\ P(N).$$

5.1.3 Inference Using Full Joint Distributions

Q14. Discuss in brief the probabilistic inference with an example.

Answer : Model Paper-II, Q10(a)

Probabilistic Inference

The computation obtained from the observed evidence of posterior probabilities for propositions of a query is known as probabilistic inference. This inference employs the full joint distribution as the knowledge base, from which solutions to all queries are obtained.

Example

Consider a domain consisting of three boolean variables: tournament, match and win. In this case, the full joint distribution will be a 2 × 2 × 2 table as follows,

	Tournament		¬ Tournament	
	Win	¬ Win	Win	¬ Win
Match	0.109	0.014	0.082	0.008
¬ Match	0.017	0.074	0.166	0.586

Here, as needed by the axioms of probability, the probabilities in the joint distribution sum up to 1. Moreover, the probability of any proposition can be directly calculated using the following equation, $P(a) = \sum_{e_i \in e(a)} P(e_i$. Therefore, the atomic events in which the proposition is true and add up their probabilities can be easily identified. For instance, there are six atomic events in which match ∨ tournament holds, that is,

$$P(\text{Match} \vee \text{Tournament}) = 0.109 + 0.014 + 0.082 + 0.008 + 0.017 + 0.074 = 0.304$$

Furthermore, summing the entries of the first row offers the unconditional or marginal probability of match, that is,

$$P(\text{match}) = 0.109 + 0.014 + 0.082 + 0.008 = 0.213$$

Q15. Write short notes on,

(i) Marginalization rule

(ii) Conditioning rule.

Answer :

(i) Marginalization Rule

The process of obtaining the unconditional or marginal probability of a variable by summing up its entities is known as marginalization or summing out. The following is the general marginalization rule for any sets of variables A and B.

$$P(A) = \sum_B P(A, B)$$

This means, a distribution over A could be derived by adding up all other variables from any joint distribution containing A. Furthermore, another form of this rule includes conditional probabilities instead of joint probabilities, using the following product rule,

$$P(A) = \sum_B P(A/B)P(B)$$

(ii) Conditioning Rule

The rule that uses conditional probabilities instead of joint probabilities is known as conditioning rule.

It uses the product rule which is as follows,

$$P(A) = \sum_B P(A/B)P(B)$$

The conditioning rule is very useful, as in many situations there is a need to calculate the conditional probabilities of few variables. These conditional probabilities can be obtained by initially using the following equation to derive an expression in terms of unconditional probabilities,

$$P(a/b) = \frac{P(a \wedge b)}{P(b)}$$

Then, evaluating the expression from the full joint distribution. For example, consider the following three variables tournament, match and win and their full joint distribution.

	Tournament		¬ Tournament	
	Win	¬ Win	Win	¬ Win
Match	0.109	0.014	0.082	0.008
¬ Match	0.017	0.074	0.166	0.586

Here, the probability of a match can be computed, given the evidence of a tournament, which is as follows,

$$P(\text{Match} \mid \text{Tournament}) = \frac{P(\text{Match} \wedge \text{Tournament})}{P(\text{Tournament})}$$

$$= \frac{0.109 + 0.014}{0.109 + 0.014 + 0.017 + 0.074}$$

$$= \frac{0.123}{0.214}$$

$$= 0.5747$$

To verify, the probability that there is no match, given a tournament can also be computed which is as follows,

$$P(\neg \text{Match} \mid \text{Tournament}) = \frac{P(\neg \text{Match} \wedge \text{Tournament})}{P(\text{Tournament})}$$

$$= \frac{0.017 + 0.074}{0.109 + 0.014 + 0.017 + 0.074}$$

$$= 0.4252$$

Here, in the above two calculations, the term $1/P(\text{Tournament})$ will be constant, irrespective of the value of match being calculated. It can be considered as a normalization constant for the distribution $P(\text{Match} \mid \text{Tournament})$, which ensures that it sums up to 1.

5.1.4 Independence

Q16. Discuss in detail the independence property with relevant example.

Answer : Model Paper-III, Q10(a)

Independence Property

The property of a variable that makes it independent of the other variables in the domain is known as independence property. For example, consider a domain with four variables-tournament, match, win and weather. Its full joint distribution would be,

$$P(\text{Tournament, Match, Win, Weather})$$

Now, it is obvious that the relationship between these variables must be known, that is, how are P(Tournament, Match, Win, Weather = Rainy) related. Here, the product rule is utilized in order to answer all such queries, i.e., P[Tournament, Match, Win, Weather = Rainy]

$$\Rightarrow P(\text{Weather} = \text{Rainy} \mid \text{Tournament, Match, Win})$$

$$= P(\text{Tournament, Match, Win})$$

Hence, it should not be interpreted that a game can influence the weather. Therefore, the following assertion would be appropriate,

$$P(\text{Weather} = \text{Rainy} \mid \text{Tournament, Match, Win})$$

$$= P(\text{Weather} = \text{Rainy})$$

From this, it can be deduced that,

$$P(\text{Tournament, Match, Win, Weather} = \text{rainy})$$

$$= P(\text{Weather} = \text{Rainy})\, P(\text{Tournament, Match, Win})$$

Similarly, for every new variable in P, a same equation will exist. Hence, the general equation would be,

$$P(\text{Tournament, Match, Win, Weather})$$

$$= P(\text{Tournament, Match, Win})\, P(\text{Weather})$$

Therefore, the weather is independent of a game which is called variables, independence or marginal independence. The graphical representation showing this independence is as follows,

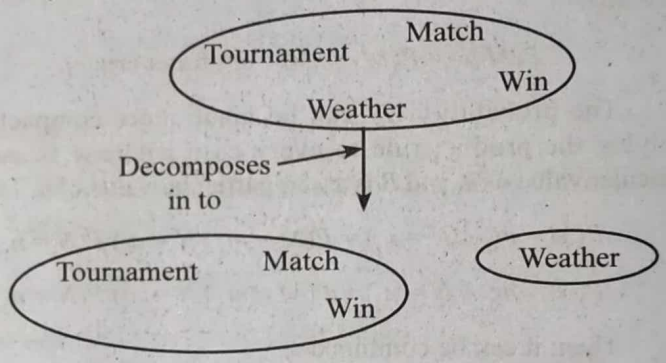


Figure: Segmenting Large Distribution into Smaller Distributions using Marginal Independence

Furthermore, independence between two propositions y and z can be defined as,

$$p(y/z) = p(y) \text{ or } p(z/y) = p(z) \text{ or } p(y \wedge z) = p(y)\, p(z)$$

5.1.5 Baye's Rule and its Use

Q17. State the Baye's rule and apply it in a simple case.

Answer : Model Paper-I, Q10(b)

Baye's Rule

Baye's rule is a simple equation that acts as a basis for probabilistic inference of various updated Artificial Intelligence (AI) systems.

Generally, it is obtained from the product rule i.e.,

$$P(x \wedge y) = P(x|y)\, P(y) \quad \ldots (1)$$

$$P(x \wedge y) = P(y|x)\, P(x) \quad \ldots (2)$$

Equations (1) and (2) can be written as,

$$P(x|y)\, P(y) = P(y|x)\, P(x)$$

$$P(y|x) = \frac{P(x \mid y)\, P(y)}{P(x)} \quad \ldots (3)$$

Typically, the P notation for multivalued variables is written as,

$$P(B|A) = \frac{P(A \mid B)\, P(B)}{P(A)}$$

Considering the background evidence 'e'. A generalized form of the equation can be generated as follows,

$$\boxed{P(B \mid A, e) = \frac{P(A \mid B, e)\, P(B \mid e)}{P(A \mid e)}} \quad \ldots (4)$$

Applying Baye's Rule in a Simple Case

Baye's rule involves two unconditional probabilities and one conditional probability for calculating conditional probability. Practically, it is very useful in the situations where the values of the above terms have a good probability estimates from which the fourth estimate can be easily generated. For example, consider medical diagnosis, where the diagnosis is performed based on the conditional probabilities existing on the causal relationships. A doctor is familiar that patients suffering from Urolithiasis (kidney stone) disease will have severe low back pain say about 40% of the time and also knows that the probability of the patient to suffer from urolithiasis is 1/2000 and the prior probability that the patient has low back pain and 'U' be the proposition that a patient has urolithiasis, then,

$P(l|u) = 0.4$

$P(u) = 1/2000$

$P(l) = 1/5$

$$P(u|l) = \frac{P(l \mid u) \cdot P(u)}{P(l)}$$

$$= \frac{0.4 \times \frac{1}{2,000}}{\frac{1}{5}}$$

$P(u|l) = 0.001$

Therefore, the expected number of patients suffering from severe low back pain to have urolithiasis is 1 in 100 patients. Although the prior probability indicates that patients have urolithiasis because of low back pain, the probability of this disease is very small as the prior probability of patients having low back pain is higher than the probability of urolithiasis.

The probability of evidence can be ignored by calculating the posterior probability of every query variable and normalizing the results. This process is applied when using Bayes' rule. Then,

$P(u|l) = \alpha(P(l|u)P(u),\ P(l|\neg u)\ P(\neg u))$

Here, $P(l|\neg u)$ is to be computed instead of $P(l)$.

The general form of Bayes rule with normalization is,

$$\boxed{P(B \mid A) = \alpha\, P(A \mid B)\, P(B)} \quad \ldots (5)$$

Where, α is normalization constant which is required to make the entries in $P(B|A)$ sum to 1.

Q18. Illustrate how Baye's rule is useful for answering probabilistic queries conditioned on one piece of evidence.

Answer :

The full joint distribution is not efficient when there are large number of variables. That is, for the three variables: tournament, match and win, the distribution is defined as P(Match | tournament $\wedge$ win) = α(value, value). But this does not scale up when more number of variables will be added. Hence, Baye's rule is used to redefine the problem, that is,

$$P(\text{Match} \mid \text{Tournament} \wedge \text{catch}) = \alpha P(\text{Tournament} \wedge \text{Win} \mid \text{Match})\ P(\text{Match}) \quad \ldots (1)$$

This redefining of the problem is applicable only when the conditional probabilities of the conjunction, tournament $\wedge$ win are discovered for every value of Match. However, this might be sufficient for just two evidence variables but will not scale up. That is, for n possible evidence variables, there are 2^n combinations of observed values for which the conditional probabilities must be known.

Hence, this approach is also not applicable in practical implementation. Therefore, another technique has been developed based on independence property of a variable. Here, it is better if the variables: tournament and win are independent, but they are not. That is, if a tournament has been won, then will definitely have been a match played. However, these variables could be independent, given the presence or the absence of a match, as each is directly caused by a match, but neither of them has a direct effect on the other. This property is represented by the following expression,

$$P(\text{Tournament} \wedge \text{Win} \mid \text{Match}) = P(\text{Tournament} \mid \text{Match})\ P(\text{Win} \mid \text{Match}) \quad \ldots (2)$$

This expression represents the conditional independence of tournament and win, provided there is a mateh. Furthermore, the probability of a match can be derived by joining this expression with the one obtained earlier using Baye's reformulation (i.e., equation (1))

i.e., $P(\text{Match} \mid \text{Tournament} \wedge \text{Win}) = \alpha P(\text{Tournament} \mid \text{Match})\ P(\text{Win} \mid \text{Match})\ P(\text{Match})$ ... (3)

This makes the information needs same as for inference using every piece of evidence separately. That is, the earlier probability P(Match) for the query variable and the conditional probability for every effect.

Generally, the conditional independence of two variables A and B can be given by the following formula when there exists another variable C.

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

For example, the conditional independence of Tournament, win, provided a game.

$$P(\text{Tournament, Win}|\text{Match}) = P(\text{Tournament}|\text{Match})\ P(\text{Win}|\text{Match})$$

The assertions are stronger than the equation (2), whose assertions are independent of some particular values i.e., tournament and Win. But, according to the absolute independence,

$P(A|B,C) = P(A|C)$ and $P(B|A, C) = P(B|C)$ is also accepted.

Generally, the process of decomposition can be adopted by absolute independent assertions with which smaller parts can be generated from full joint distribution. The same process can also be adopted in the conditional independence assertions.

For instance, the decomposition of assertion in the equation (2) is as follows,

P(Tournament, Win, Match)

$= P$(Tournament, Win|Match). P(Match)

[$\therefore$ Product rules]

$= P$(Tournament|Match) P(Win|Match) P(Match)

[$\therefore$ From equation (2)]

The major table is decomposed into some small tables. In general, the table consists of $2^3 - 1$ i.e., seven independent numbers. Now, each of the smaller tables contain five independent numbers which depict that for n conditionally independent symptoms with the third variable Match, the representation of size increases to $O(n)$ instead of $O(2^n)$. Using conditional independence, large probabilistic domains are divided into some weakly connected subsets which are known to be the great developments in the evolution of artificial intelligence.

5.2 PROBABILISTIC REASONING

5.2.1 Representing Knowledge in an Uncertain Domain

Q19. What are the problems of full joint probability distribution? How to overcome from it and represent the knowledge in uncertain domain?

Answer : Model Paper-II, Q10(b)

Problems of Full Joint Probability Distribution

1. The increase in number of variables makes the full joint probability distribution to become intractably large.
2. Specification of probabilities for all the possible things in a sequential manner is a tiresome and a tedious job.

The above problems can be overcome by using a data structure called "Bayesian network".

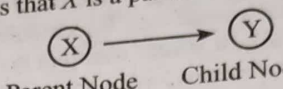
Bayesian Networks

Bayesian network also known as a *belief network, probabilistic network, causal network or a knowledge map* is a directed graph in which each node has a quantitative probability information.

Properties of a Bayesian Network

(i) A node in a Bayesian network represents a discrete or a continuous random variable.

(ii) A set of directed arrows/links is used for connecting the nodes within a network i.e., an arrow from node X to Y represents that X is a parent of Y.



(iii) It is a Directed Acyclic Graph (DAG) as it doesn't contain any directed cycle.

(iv) Each individual node has an associated conditional probability distribution $P(X_i \mid \text{Parents } (X_i))$ that determines the effect of the parents on a particular node.

The arrangement of nodes and their interconnections within the network is independent of conditional relationships. The arrows represent that one node has direct impact on the other. Therefore, these direct relationships can be easily determined in a Bayesian network. When the arrangement of lines and nodes is finalized, the conditional probability can be assigned to the child variables.

Example

Consider the following Bayesian network.

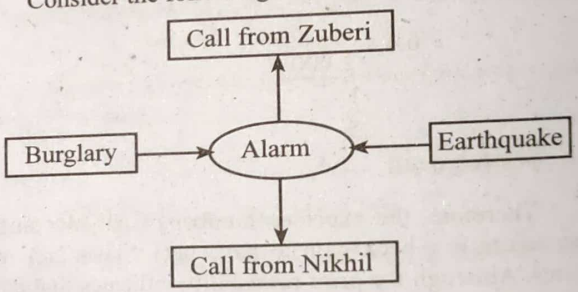


Figure: Bayesian Network

The above Bayesian network carries an alarm system which raises alarm when a burglary or earthquake is detected. The owner of house requested his two neighbours Zuberi and Nikhil to inform him when they hear any alarm. Zuberi always responds on the alarm and calls the owner but he sometimes calls even when he hear a telephone ring. Nikhil not always calls the owner because he likes to hear loud music. Apart from this, there can be other probabilities such as power failure. These conditional probabilities are not included in the network thereby leading to the uncertainties to be involved in the network. These uncertainties can be minimized by including some description in the Bayesian network. One such solution is to use Conditional Probability Tables (CPT) for each of the node in the network. Thus the modified Bayesian network can be given as follows,

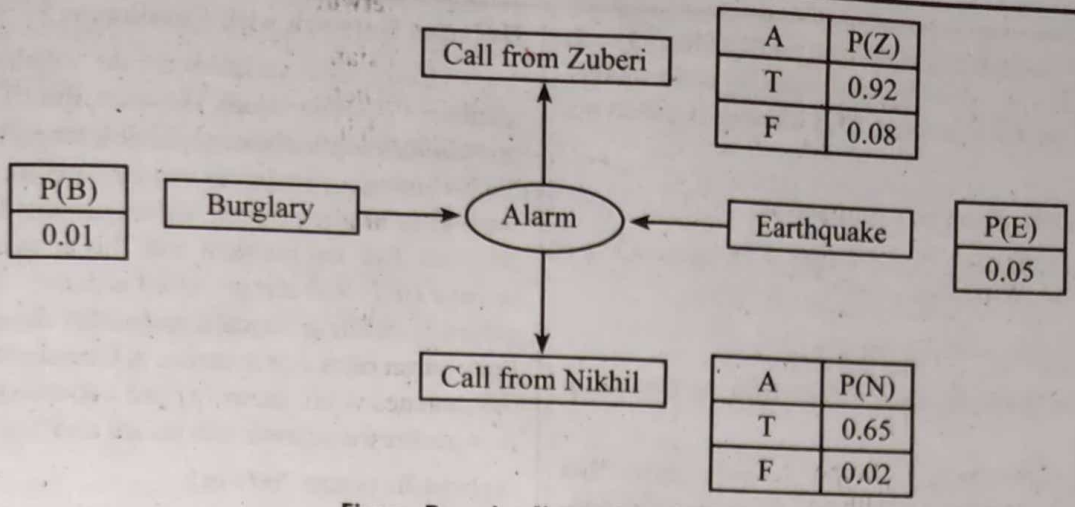


Figure: Bayesian Network with CPTs

5.2.2 The Semantics of Bayesian Networks

Q20. Explain the two ways in which one can understand the semantics of Bayesian networks.

Answer :

Model Paper-III, Q10(b)

The two ways in which the semantics of Bayesian networks can be understood are as follows,

1. Network Representation by Joint Probability Distribution

Bayesian network is a directed acyclic graph where each node is associated with some numeric parameters (α_i). Its semantics can be defined by representing joint distribution with respect to the variables.

Each entry in the joint probability distribution is actually the probability of conjunction of individual variables corresponding to each node 'X_i' i.e.,

$$P(X_i = x_1 \wedge x_2 \wedge x_3, \wedge \ldots \wedge x_n)$$

$$P(x_1, \ldots x_n) = \prod_{i=1}^{n} \alpha(x_i \mid \text{Parent}(X_i)) \quad \ldots (1)$$

Where,

Parent(X_i) = Values of Parent(X_i) that appear in $x_1, x_2, \ldots x_n$

Hence, each entry in the joint probability distribution is expressed as the product of values present in CPTs of the Bayesian network.

Equation (1) can then be rewritten as $P(x_1, x_2, x_3, \ldots x_n) = \prod_{i=1}^{n} P(x_i \mid \text{Parent}(x_i))$

Thus, in the example of burglary and earthquake, the resultant probability can be represented as follows,

$$P(Z, N, Al, \neg B, \neg E) = P(Z \mid Al)\, P(N \mid Al)\, P(Al \mid \neg B \wedge \neg E)\, P(\neg B)\, P(\neg E)$$

$$= 0.92 \times 0.65 \times 0.01 \times 0.99 \times 0.95$$

$$= 0.00562419$$

Hence, the Bayesian network when represented using the joint probability distribution can be used to answer any query just by adding up all the relevant joint entries.

The basic concepts used in Bayesian networks are as follows,

(a) Locally Structured Systems

The system in which each subcomponent directly communicates with only a bounded limited number of components irrespective of the total number of components is called locally structured systems. It is also called the sparse system. The locally structured domain even leads to a compact Bayesian network if the ordering of nodes can be chosen wisely.

(b) Chain Rule

The identity for entries in joint distribution can be written in terms of conditional probability using product rule as,

$$P(x_1, x_2, \ldots x_n) = P(x_n \mid x_{n-1}, \ldots x_1)\, P(x_{n-1}, \ldots x_1)$$

The process of rewriting is iterated while reducing each conjunctive probability to a conditional probability and a smaller conjunction results in a big product. This identity is called the chain rule.

Mathematically, it can be expressed as,

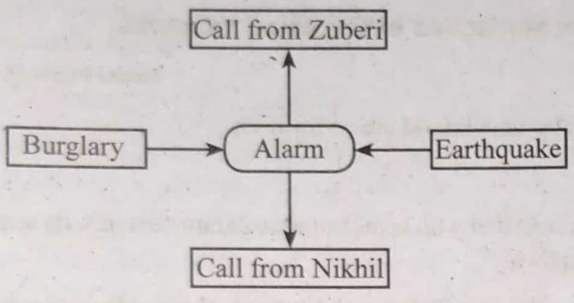
$$P(x_1, \ldots x_n) = P(x_n | x_{n-1}, \ldots, x_1) P(x_{n-1} | x_{n-2}, \ldots x_1) \ldots P(x_2 | x_1) P(x_1)$$

$$P(X_i | X_{i-1}) \ldots X_1) = P(X_i | \text{parents } (X_i))$$

2. Conditional Independence Relation in Bayesian Networks

The topological semantics of a network states that each of the variables is conditionally independent of the non-descendants associated with it provided its parents.

Example



In the above example, "Call form Nikhil" is independent of "Burglary", "Earthquake" and "Call from Zuberi" provided the value of "Alarm".

The topological semantics of a network also states that a node is conditionally independent of all the other nodes present in the network provided its parents, children and the parents of children. This property is referred to as the 'Markov Blanket'.

Referring to above figure, the act of 'Burglary' is independent of the 'Call from Zuberi' and 'Call from Nikhil' provided the variables 'Alarm' and 'Earthquake'.

5.2.3 Efficient Representation of Conditional Distributions

Q21. Define deterministic node with suitable example. Also discuss about conditional distribution in Bayesian network with continuous variables and hybrid Bayesian network.

Answer :

A deterministic node is a node whose value is determined by the values of its parents without any uncertainty. The parent-child relationship can either be logical or numerical.

An example of logical relationship is the relationship between the parent nodes India, China, Pakistan and the child node Asia where child is a disjunction of its parents.

An example of numerical relationship is, if the parent nodes represent the price of a specific brand of laptop at different laptop dealers and the child node is the final discounted price paid by the customer, then the child node is the minimum of the values that the parent nodes has.

Bayesian Network with Continuous Variables

Continuous variables are the variables that have infinite number of possible values. However, specification of conditional probabilities explicitly is not possible for each of available values. Such continuous variables can be handled using discretization method in all the possible values are divided in a fixed set of intervals. But, the problem with this approach is that, it results in large CPTs and compromised accuracy. This problem can be solved by defining standard probability density functions having finite parameters. For instance, in Gaussian distribution $N(\mu, \sigma^2)$ the parameters are mean (μ) and variance (σ^2). In some cases, non-parametric representations are also used.

Hybrid Bayesian Network

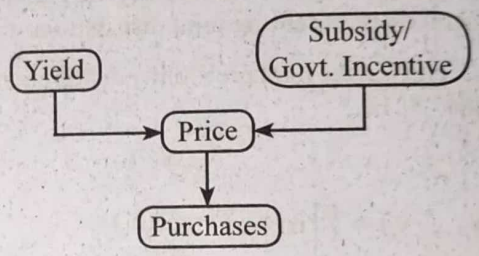
A Bayesian network consisting of both continuous and discrete variables is called a "Hybrid Bayesian Network".

The specification of a hybrid-network involves the following distributions.

(i) The conditional distribution for a discrete variable provided the continuous parents.

(ii) The conditional distribution for a continuous variable provided the continuous or discrete parents.

Example



Consider an example of a customer purchasing some vegetables. This purchase depends on the price of the vegetables which is inturn dependent on the crop yield and the government's subsidy applicable on the vegetables. Here, the variable 'price' is continuous and has both continuous and discrete parents and the variable 'purchases' is discrete with a continuous parent.

The probability P(Price/Yield, Subsidy) needs to be specified for the 'Price' variable. The enumeration method is used to handle the discrete parent i.e., P(Price|Yield, Subsidy) and P(Price|Yield, ¬ Subsidy). The variable 'Yield' can be handled by specifying the distribution of price 'p' and its dependency on the continuous value 'y' of 'Yield'. In such situation, the linear Gaussian distribution is used in which the child has a Gaussian distribution with mean 'μ' and standard deviation 'σ'. Here 'μ' varies linearly with the parent's value and 'σ' is fixed.

Hence, the following two distributions are required.

$$P(p|y, \text{Subsidy}) = N(a_t y + b_t, \sigma_t^2)(p) = \frac{1}{\sigma_t \sqrt{2\pi}} e^{\frac{-1}{2}\left(\frac{\phi - (a_t y + b_t)}{a_t}\right)^2}$$

and

$$P(p|y, \neg \text{Subsidy}) = N(a_f y + b_f, \sigma_f^2)(p) = \frac{1}{\sigma_f \sqrt{2\pi}} e^{\frac{-1}{2}\left(\frac{P - (a_f y + b_f)}{a_f}\right)^2}$$

5.2.4 Approximate Inference in Bayesian Networks

Q22. Explain the need of approximate inference in Bayesian networks. Also, discuss the direct sampling methods.

Answer : Model Paper-I, Q11(a)

Need of Approximate Inference in Bayesian Networks

Large networks with multiple connections are intractable for exact inference hence, it is necessary to consider approximate inference methods. For this, many randomized sampling algorithms (also called Monte Carlo Algorithms) have been designed. These algorithms provide approximate answers where the accuracy of such approximate answers relies on the amount of samples generated.

Direct Sampling Methods

Direct sampling can be used to sample variables in a specific order from Bayesian network which does not have any evidence. Typically, the samples are generated from a probability distribution which known. For instance, consider the event of tossing a coin in which there are two possibilities i.e., either head or tail. Therefore the prior distribution of coin P(coin) is (0.5, 0.5). Here, the probability of getting head is 0.5 and the probability of getting tail is 0.5. Hence, it is easy to perform sampling when distribution carries a single or known variables.

In direct sampling, the sampling is performed by conditioning the probability distribution with respect to the values assigned to the parents of the variables. The algorithm employed is as follows,

Function PRIOR-SAMPLE(ps) **returns** an event which is generated after performing sampling on prior of ps.

Inputs: ps which is a Bayesian network carrying joint distribution $P(X_1, X_2, ..., X_n)$

x: Event carrying 'n' number of elements

For i = 1 **to** n **do**

$x_i \leftarrow$ a sample drawn from $P(X_i|\text{parents}(X_i))$ randomly **return** x

The algorithm can be applied on the following sample diagram,

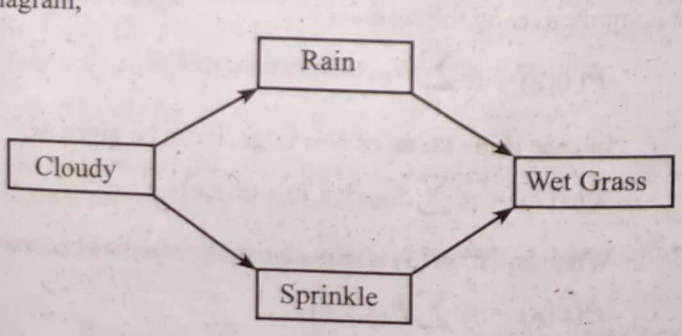


Figure: Sample Multiply Connected Network

Consider that the ordering of events in the above figure is [Cloudy, Rain, Sprinkle, Wet Grass]. The prior distribution (sampling) of these events are as follows,

- $P(\text{Cloudy}) = (0.5, 0.5)$; let the event returns true.
- $P(\text{Rain}|\text{Sprinkle} = \text{true}) = (0.1, 0.9)$; let the event returns false
- $P(\text{Sprinkle}|\text{Cloudy} = \text{true}) = (0.8, 0.2)$; let the event returns true
- $P(\text{WetGrass}|\text{Rain} = \text{false}, \text{Sprinkle} = \text{true}) = (0.9, 0.1)$; let the event returns true

The prior-sample results of the example data returns the event in the sequence [true, false, true, true].

The generation of samples using PRIOR-SAMPLE algorithm is simple. To illustrate this, consider the probability of an event $S_{ps}(x_1, x_2, ..., x_n)$ which is obtained using PRIOR-SAMPLE algorithm. The steps involved in the sampling process is dependent on their parents. Therefore, the probability can given as,

$$S_{ps}(x_1, x_2, ..., x_n) = \prod_{i=1}^{n} P(x_i|\text{parents}(X_i))$$

According to the Bayesian network representation, $S_{ps}(x_1, x_2, ..., x_n) = P(x_1, x_2, ..., x_n)$. Hence, it becomes simple to answer queries using samples.

To compute the sampling probability, typical sampling algorithm is considered where the computation of answers is performed based on the number of samples generated. The frequency of the set of events $x_1, x_2, ..., x_n$ can be constrained within the limits as,

$$\lim_{N \to \infty} \frac{N_{ps}(x_1, x_2, ..., x_n)}{N} = S_{ps}(x_1, x_2, ..., x_n) = P(x_1, x_2, ..., x_n)$$

Rejection Sampling in Bayesian Networks

Rejection sampling is used for generating samples from hand-to-sample distribution when easy-to-sample distribution is provided. This sampling method is used to evaluate the value P(X|e) which is called the conditional probabilities. Typical algorithm for rejection sampling is as follows,

Function REJECTION-SAMP (X, e, ps, N) **returns** estimated value of $P(X|e)$

inputs: query variable 'X'

evidence 'e'

Bayesian network 'ps'

count of samples to be generated 'N'

Local Variables: Vector of counts over query variable whose value is initialized to zero.

for j = 1 to N **do**

x ← PRIOR-SAMPLE(ps)

If x is consistent with e where x is a random value

then

$N[x] \leftarrow N[x] + 1$

return Normalize($N[x]$)

In this algorithm, initially the prior distribution is considered to produce the samples. In the next step, it compares the samples with the evidence and rejects the ones which are not matching. In the last step, the estimation of $\hat{P}(X = x|e)$ is computed which is nothing but the count of x occurrence in the samples.

To determine whether the output generated by this algorithm is consistent or not, consider estimated distribution which is the output of this algorithm as $\hat{P}(X|e)$. Then according to the rejected sampling,

$$\hat{P}(X|e) = \alpha N_{ps}(X,e) = \frac{N_{ps}(X,e)}{N_{ps}(e)} \quad \ldots (1)$$

According to the consistent estimate formula for probability of partially specified events,

$$P(x_1, x_2, \ldots, x_n) \approx P(x_1, x_2, \ldots, x_n)/N \quad \ldots (2)$$

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

$$\hat{P}(X|e) \approx \frac{P(X,e)}{P(e)} = P(X|e)$$

Hence, the algorithm returns consistent estimate. However, the problem associated with this algorithm is that the number of samples rejected is larger and therefore, it cannot be used for complex problems.

Likelihood Weighting Algorithm

Likelihood weighting algorithm overcomes the drawback of rejection sampling algorithm which rejects large number of samples which do not match with the evidence. In likelihood weighting, the events are considered which are consistent with the evidence. Typical algorithm for likelihood weighting is as follows,

Function LIKELIHOOD-WEIGHTING(X,e,bn,N) **returns** estimated value of $P(X|e)$

Inputs : Query variable 'x'
evidence e
Bayesian Network 'bn'
Count of samples to be generated N.

Local variables : a vector whose weighted count is over and with an initial value of zero,

for j=1 to N

do

$x,w \leftarrow$ weighted sample(bn)

$W[x] \leftarrow W[x] + w$

return NORMALIZE ($W[X]$)

Function WEIGHTED-SAMPLE(bn,e)

returns event and weight

$x \leftarrow$ event which carries 'n' number of elements.

for i=1 to n

do

if X_i carry x_i in e

then $w \leftarrow w \times P(X_i = x_i \mid \text{parents}(X_i))$

else $x[i] \leftarrow$ sample drawn from $P(X_i \mid \text{parents}(X_i))$ randomly.

return x,w

This algorithm overcomes the drawback of rejection algorithm by correcting the events with respect to the evidence. The events which cannot be corrected are rejected. Based on the likelihood of events and evidence, each event is assigned with certain weight(value). This value is computed by multiplying the probabilities (conditional) of every event variable. Minimum weight is allotted to the events in which evidence becomes unlikely.

The working likelihood weighting algorithm can be illustrated by considering the sampling distribution S_{ws} where W_s refers to the weighted sample. The variables in the evidence E are represented with the values e. Let Z be some other variables where,

$Z = \{X\} \cup \{Y\}$. When Z is provided along with its parents, the likelihood algorithm performs sampling as,

$$S_{WS}(z, e) = \prod_{i=1}^{l} P(z_i \mid \text{parents}(Z_i)) \quad \ldots (1)$$

Z_i in this algorithm focuses on both evidence variables and hidden variables. The parents of Z_i impacts on the sampled values of Z_i. However the focus of S_{WS} is more on posterior distribution $P(z|e)$ instead of the evidence.

The weight w used in this algorithm is computed as the difference between required and actual values of the distribution. However if the weight of certain sample x which is the composition of both z and e, then it can be computed as the product of variables provided with their parents.

$$w(z, e) = \prod_{i=1}^{m} P(e_i \mid \text{parents}(E_i)) \quad \ldots (2)$$

The product of equations (1) and (2) concludes that the probability of sample is in a convenient form which is,

$$S_{WS}(z, e)\, w(z, e) = \prod_{i=1}^{l} P(z_i \mid \text{parents}(z_i)) \prod_{i=1}^{m} P(e_i \mid \text{parents}(E_i))$$

$$= P(z, e) \quad \ldots (3)$$

The posterior probability using likelihood weighting can be computed using the formula

$$\hat{P}(x \mid e) = \alpha \sum_{y} N_{WS}(x,y,e)\, w(x,y,e)$$

In case if the value of N is large, it can be given as,

$$\hat{P}(x \mid e) \approx \alpha' \sum_{y} S_{WS}(x,y,e)\, w(x,y,e)$$

When equation (3) is considered, the equation becomes,

$$\hat{P}(x \mid e) = \alpha' \sum_{y} P(x,y,e)$$

$$= \alpha' P(x,e) = P(x|e)$$

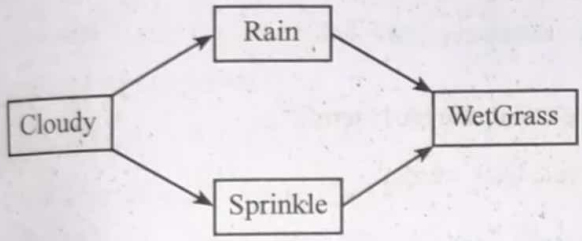
Hence, the estimations provided by a likelihood weighting algorithm are also consistent.

Q23. Briefly discuss about the Markov Chain Monte Carlo algorithm.

Answer :

MCMC Algorithm

Markov Chain Monte Carlo (MCMC) is an algorithm which produces events by randomly modifying previous events instead of producing them from scratch. Here, the network is considered as being in a current state carrying certain values for each of the variables. The successor state is produced by sampling the value of non evidence variable X_i which is drawn randomly, depends on the present value of a variable. Hence, MCMC ambles around the state space of all available assignments and flips variables one by one while considering evidence variables as constant. For example, P(Rain|sprinkle = true, WetGrass = true) is a query for the following figure,



Figure

In this, the Sprinkle and WetGrass are the evidence variables whose values are fixed to their observed values. Cloudy and rain are the hidden variables where the value or Rain is true and the value of cloudy is false. Hence, [true, true, false, true] is known as initial state. After this, the steps mentioned below are carried out iteratively.

1. **Cloudy is Sampled**

When the current values of Markov blanket variables are provided, the cloudy is sampled from the query. P(Cloudy|Sprinkle = true, Rain = false). If the output is Cloudy = false, then the current state is modified to [false, true, false, true].

2. **Rain is Sampled**

When the current values of Markov blanket variables are provided, the Rain is sampled from the query P(Rain|Cloudy = false, Sprinkle = true, WetGrass = true). If the output is Rain = true, then the current state is modified to [false, true, true, true]. Suppose, The process visited 20 states and 60 states when the Rain is true and false respectively, then the result is,

NORMAL1ZE(<20,60>) = <0.25, 0.75>.

Now, the MCMC algorithm for approximate inference in Bayesian network is as follows,

function MCMC_ASK(X, e, B, N) returns an estimate of P(X|e)

Inputs: N[X] is a vector of counts over X, primarily zero

Z is the set of nonevidence variables in B

X is the current state of the network, primarily copied from e

Assign x with random values for the variables in Z

for j=l to N do

N[x]→N[x]+1 where x is the value of X in x

for each Z_i in Z do

sample the value of Z_i in x from $P(Z_i|mb(Z_i))$ given the values of $mb(Z_i)$ in x

return NORMALIZE(N[X])

Working of MCMC

The output of MCMC algorithm is consistent with respect to the estimates for posterior probabilities. In this, the sampling process coincides with the dynamic equilibrium where the long-run fraction of time consumed in every state is directly proportional to its posterior probability. It supports the transition probability, using which a process can be moved stage to stage based on the conditional distribution included in Markov blanket of the variable. Let the process provides a transition from a state x to x' with a probability of $q(x \rightarrow x')$ which refers to the Markov chain on the state space.

Let,

t be the number of steps for which Markov chain runs.

$\pi_t(x)$ be the probability of a system in state x at time 't'

$\pi_{t+1}(x')$ be the probability of a system in state x' at time 't+1'.

Now, the $\pi_{t+1}(x')$ can be calculated by summing all the possible states the system can be at time, the probability of a transition from x to x'

$$\pi_{t+1}(x') = \sum_x \pi_t(x) q(x \rightarrow x') \quad \ldots (1)$$

From the above equation, one can conclude that the chain has arrived to its stationary distribution if $\pi_t = \pi_{t+1}$. Now, call the stationary distribution(π) which is represented as,

$$\pi(x') = \sum_x \pi(x).q(x \rightarrow x') \text{ for every } x' \quad \ldots (2)$$

There exists only one distribution π which satisfies the equation for any value of q under some specific conditions regarding the transition probability distribution.

Equation (2) can be considered as, the expected out-flow associated with every state is identical to the expected in flow of all states. This condition can be satisfied when the expected flow between, any two states is equal in both directions which is called as a property of detailed balance which can be given as,

$$\pi(x')q(x' \rightarrow x) = \pi(x)q(x \rightarrow x') \ \forall x, x' \quad \ldots (3)$$

This equation leads to stationarity simply by summing over x in the equation (3). Then

$$\sum_x \pi(x') \ (x', \rightarrow \ x) \Rightarrow \sum_x \ \pi(x)q(x \rightarrow \ x') \Rightarrow \pi(x') \sum_x$$

$$q(x' \rightarrow \ x) = \pi \ (x')$$

Here, the last step continues due to the occurrence of a transition from x'.

To prove that the transition probability $q(x \rightarrow x')$ referred by a sampling step in MCMC_ASK follows the detailed balance property with a stationary distribution equal to $P(x|e)$, two steps are required.

- ❖ Markov chain is introduced where all the variables should be dependent on the current values of other variables, and it satisfies detailed balance.
- ❖ Bayesian network is monitored to track that the sampling is conditionally equal on the remaining variables, in $\text{mb}(x_i)$.

Assume that, the variable to be sampled is x_i and its value in the current state is x_i. The hidden variable to be sampled is $\bar{x}_i$ and its value in the current state is x_i the hidden variable to be sampled is $\bar{x}_i$ and its values in the current state is $\bar{x}_i$ and its values in the current state is $\bar{x}_i$. Suppose a value x_i' is introduced for X_i conditionally on all the variables. Which includes evidence variable(e). Then,

$$q(x \rightarrow x') = q((x_i, \bar{x}_i) \rightarrow (x_i', \bar{x}_i)) = P(x_i' | \bar{x}_i, e)$$

This is called as Gibbs sampler which is a specific convenient form of MCMC. The Gibbs sampler in a detailed balance property with the true posterior can be given as,

$$\pi(x)q(x \rightarrow x') = P(x|e)P(x_i' | \bar{x}_i, e)$$

$$= P(x_i, \bar{x}_i | e)P(x_i' | \bar{x}_i, e)$$

$$= P(x_i | \bar{x}_i e)P(\bar{x}_i | e)P(x_i' | \bar{x}_i, e) \qquad [\because \text{from the chain rule on } 1^{st} \text{ term}]$$

$$= P(x_i | \bar{x}_i, e)P(x_i', \bar{x}_i | e) \qquad [\because \text{Chain rule back wards}]$$

$$\Rightarrow \pi(x')q(x' \rightarrow x)$$

The variable in a Markov blanket does not depends on the other variables. Therefore,

$$P(x_i' | \bar{x}_i, e) = P(x_i' | \text{mb}(X_i))$$

Where,

$\text{mb}(X_i)$ is the Markov blanket of X_i which includes the values of all variables in X_i

The probability of a variable in Markov blanket is directly proportional to the product of the probabilities of the variables when parents are provided along with the respective child probability. It can be given as,

$$P(X_i' | \text{mb}(X_i)) = \alpha P(x_i' | \text{parents}(X_i)) \times \prod_{y_j \in \text{children}(X_i)} P(y_j | \text{parents}(Y_j)) \qquad \ldots (4)$$

Therefore, the number of multiplications needed to every variable X_i is equal to the number of children of X_i.

5.2.5 Relational and First-Order Probability

Q24. What are the problems associated with first-order models? How to overcome from it using relational probability models.

Answer :

Problems with First-Order Models

1. The first-order models are infinite.
2. The summation of the possible worlds could be infeasible i.e.,

$$P(\phi) = \sum_{w:\phi \text{ is true in } w} P(w) - (\text{Infeasible})$$

3. The specification of a complete and consistent distribution over an infinite set of worlds may be very difficult.

The above problems can be overcome by using Relational Probability Models (RPMs).

Relational Probability Models (RPMs)

RPMs are the models that specify probabilities on relations irrespective of individual objects. They do not make any closed-world assumptions.

They also have a constant, function and predicate symbols like the first-order logic models. Also, a type signature containing the type of each argument and the value of the function.

Consider the book recommendation domain, let

Customer, 'cust' and book, 'book' be the types. So, the type signatures for the functions and predicates can be described as follows,

Hon: Cust → {T, F}

Kind: Cust → {1, 2, 3, 4, 5}

Qual: Book → {1, 2, 3, 4, 5}

Recom: Cust × Book → {1, 2, 3, 4, 5}

Here, the names of the customers and the title of the book are the constant symbols that exist in the retailer's data set as $\{C_1, C_2, ...\}$ and $\{B_1, B_2, ...\}$ respectively. Thus, the random variables of the RPM can be obtained by instantiating each function with all the combinations of objects that are possible i.e.,

Hon(C_1), Qual(B_2), Recom(C_1, B_2) etc.

In order to complete the RPM, the dependencies that exists among these random variables are identified as follows,

Hon(c) ~ (0.01, 0.99)

Kind(c) ~ (0.3, 0.3, 0.2, 0.1, 0.1)

Qual(b) ~ (0.4, 0.3, 0.1, 0.16, 0.04)

Recom (c, b) ~ Rec CPT (Hon(c), Kind(c), Qual(b))

Where,

Rec CPT is a conditional distribution with 50 rows (2 × 5 × 5 rows).

This model can be refined by a context-specific independence model which defines a variable irrespective of some of its parents provided certain values of others.

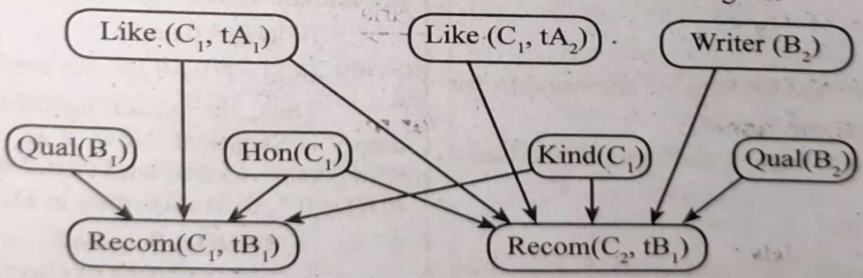
Hence,

Recom(c, b) ~ if Hon(c) then

Hon Rec CPT (Kind(c), Qual(b))

else (0.3, 0.1, 0.1, 0.2, 0.3)

A more realistic approach can also be used to elaborate this model as shown in figure.



In the above figure 'like' represents an honest customer who likes the author/writer and always prefer his book and gives a high rating to the book irrespective of its quality i.e.,

Recom(c, b) ~ if Hon(c) then

if Like(c, Writer(b)) then Exactly(5)

else Hon Rec CRT (Kind(c), Qual(b))

else (0.3, 0.1, 0.1, 0.2, 0.3)

Again, Like(c, Writer(b)) i.e., the conditional test is unknown but if a customer of a book gives '5' to a specific writer's book and is not kind then the posterior probability that the customer likes the author is very high.

Q25. Write short notes on Open Universe Probability Models (OUPMs.)

Answer :

Open Universe Probability Models (OUPMs)

Open Universe Probability Models are the models that are based on the standard semantics of the first-order logic models. The language used for OUPMs helps in providing a unique and consistent probability distribution over the infinite space of possible worlds.

These models add objects to the world depending on the number and type of the already existed objects in that world. For the book-recommendation domain, in order to differentiate between customers and their IDs prior log-normal distribution is used as follows,

Cust ~ Log Normal [7.2, 3.2^2] ( ) (for 100-10,000 customers)

Here,

An honest customer is assumed to have a single customer ID whereas a dishonest customer can have multiple customer IDs between 10 and 100 IDs. This can be expressed as,

#CustID (Owner = c) ~ if Hon(c) then Exactly (1) lese LogNormal [7.2, 3.2^2] ( )

The function 'Owner' is called an 'origin function' because it specifies the source of the generated object.

Advantages

1. These models define a unique distribution over the possible worlds.
2. Inference algorithms are available here which returns the answers within the available limits.

5.2.6 Other Approaches to Uncertain Reasoning: Dempster-Shafer Theory

Q26. "Logical systems in general and logical rule-based systems in particular have three properties".

Answer :

Illustrate how truth functionality is not suitable for uncertain reasoning in rule-based systems.

The three properties of logical and logical rule-based systems are as follows,

(i) Locality

In the logical systems, for a rule $A \Rightarrow B$, it can be concluded as B, provided the evidence 'A' without taking into consideration any other rules.

On a contrary, in probabilistic systems, all the evidences need to be considered.

(ii) Detachment

After getting logical proof for a proposition 'B', it can be used anywhere irrespective of how it was derived. In short, it can be detached from its justification.

On a contrary, in probabilistic systems, the source of the evidence is required for further reasoning.

(iii) Truth-functionality

In logical systems, the truth value of the components is used to compute the truth value of the complex sentences. On a contrary, probabilistic systems do not work in this way, except some strong global independence assumptions are made.

Inappropriateness of Truth Functionality for Uncertain Reasoning in Rule-bases Systems

Let E_1 be the event of flipping a fair coin with 'heads' up.

Let E_2 be the event of flipping a fair coin with 'tails' up.

and Let E_3 be the event of flipping a fair coin for the second time with 'heads' up.

The probability for all the events to occur is 0.5. So, the same belief is assigned by the truth functional system to the disjunction of any of the above two events. Whereas, the probability of disjunction is dependent on the events but not on their probabilities alone.

P(X)	P(Y)	P(X ∨ Y)
P(E$_1$ = 0.5)	P(E$_1$ = 0.5)	P(E$_1$ ∨ E$_1$) = 0.50
	P(E$_2$ = 0.5)	P(E$_1$ ∨ E$_2$) = 1.00
	P(E$_3$ = 0.5)	P(E$_1$ ∨ E$_3$) = 0.75

It leads to a worsen situation when the evidences are chained together.

Q27. Explain in detail about the Dempster-Shafer theory.

Answer : **Model Paper-II, Q11(a)**

The Dempster-Shafer theory helps in distinguishing between uncertainty and ignorance. Here, instead of calculating the proposition's probability, the probability that the evidence supports the proposition is calculated. This is called a belief function and is usually represented as Bel(X).

Consider an example of flipping a coin. If it is drawn from a magician's pocket, it may or may not be fair. Also, the belief to be assigned to the event of getting a head up is also not known. Hence, according to Dempster-Shafer theory,

Bel(H) = 0 and

Bel ($\neg H$) = 0 (As no evidence is known)

Thus, the reasoning done by using Dempster-Shafer theory is Skeptical. Now, if an expert is three at the disposal who gives evidence with 90% certainty that the coin is fair i.e., $P(H) = 0.5$, then according to Dempster-Shafer theory,

Bel(H) = 0.9 × 0.5

= 0.45 (At 90% certainty)

and (Bel)($\neg H$) = 0.45

On the whole, there is still a 10% gap that is not considered by the evidence.

Mathematically, this theory assigns masses to the events which sums up to '1'. In other words, Bel(X) is the sum of the masses of all the events that are subsets of X, inclusive of X. Hence, Bel(X) + Bel($\neg X$) ≤ 1 (atmost 1) and the interval between Bel(X) and 1-Bel($\neg X$) is interpreted as bounding the probability of X.

Due to the existence of a gap in the beliefs, the Dempster-Shafer system is not able to make any decision.

5.3 LEARNING

5.3.1 Forms of Learning, Supervised Learning

Q28. Explain in detail the different issues considered while designing a learning element.

Answer :

The different issues considered while designing a learning element are as follows,

1. Learning of components that are related to performance element.
2. Reading the obtained feedback in order to learn the components.
3. Specifying the representation for the learned components.

1. Learning of Components that are Related to Performance Element

The components which are to be learnt are as follows,

(a) The conditions lying on the current state are directly mapped to the particular actions

(b) The method of deriving the appropriate properties for the world from the percept set.

(c) The information regarding the methods that are derived by the world and also the outcomes of applicable actions that are undertaken by the agent.

(d) The desirability of world states is specified using the utility information.

(e) The desirability of actions is specified using the action value information.

(f) Classes of states that are characterized by the goals in which the utility of the agents is incremented after the completion of the task.

These components can be learnt from the relevant feedback.

For example, consider an agent who is giving training to drive a car. This training includes the following components.

(a) When the instructor instructs for a "Brake" then the agent learns the timing when to use brake.

(b) The number of camera images are viewed such that the information regarding the presence of buses is determined to the agent.

(c) The results are noticed after attempting the actions.

For example, on using a hand brake on wet road, the effects that are caused due to the actions are learnt.

(d) When any tip is not received from the passengers who have perfectly completed their trip, then the agent can learn that it is a beneficial component for its entire utility function.

2. Reading the Obtained Feedback inorder to Learn the Components

For answer refer Unit-V, Q29.

3. Specifying the Representation for the Learned Components

In order to determine the working of the learned algorithm it is necessary structure learned information. Each component of an agent can be represented with different forms. For this purpose, a wide variety of algorithms are learnt.

Q29. Define learning. Explain various forms of learning.

Answer : Model Paper-III, Q11(a)

Learning

Learning can be defined as the process of gaining knowledge about any specific subject and draw new methods that can handle any situation.

Forms of Learning

Various forms of learning are categorized based on the type of feedback obtained.

1. Supervised Learning

In supervised learning, the machine is trained by using the data that is labelled. It means that the data is already available with the correct answer. Later on, the machine is provided with a set of examples for supervised learning algorithm to analyze the training data. Then the correct output is generated from labelled data. The supervised learning algorithm will learn from the labelled training data. It helps the users to predict the outcomes for unforeseen data.

Time and highly skilled data scientists are required to successfully build, scale and deploy the accurate supervised machine learning data science model. The data scientist needs to rebuild the models to assure the insights are true until data changes.

Supervised learning is classified into two types of algorithms. They are as follows,

(a) Classification

The classification problem is useful when output variable is category like "Red" or "Blue" or "Disease" or "no disease".

(b) Regression

The regression problem is useful when output variable is a real value like dollars or weight.

2. Unsupervised Learning

Unsupervised learning is the process of training the machine through data that is not classified and labeled through algorithm to act on data without any guidance. The machine will group all the unordered data according to similarities, patterns and differences without prior training of data. The machine is not provided any type of training. So it cannot find the hidden structure in unlabeled data by itself. The unsupervised learning algorithms allows to perform more complex processing tasks. It can be more unpredicted compared to other natural learning, deep learning and reinforcement learning methods. Unsupervised learning is classified into two categories of algorithms. They are as follows,

(a) Clustering

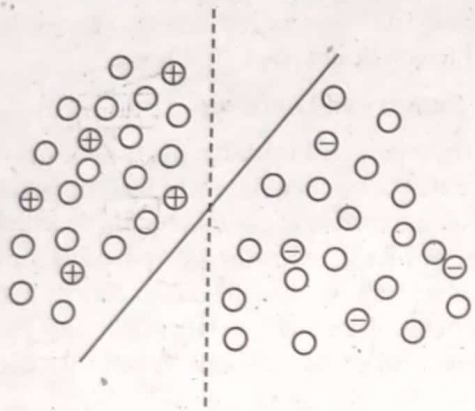
The clustering is useful to discover the inherent groupings in the data like grouping of customers by purchasing behavior.

(b) Association

An association rule learning problem is useful to discover the rules that describe large part of data like people who purchase X and Y.

3. Semi-supervised Learning

In this process, the input examples are both labeled and unlabeled. Here, the unlabeled examples are used to clarify the different boundaries in between the classes and the labeled examples are used to learn the class models. For example in case of a two-class problem, one class is chosen as positive class example and other class is chosen as negative class example.



⊖ Negative example

⊕ Positive example

○ Unlabeled example

— Decision boundary with unlabeled examples

--- Decision boundary without unlabeled examples

In the above figure, the dashed line separates the positive class examples from the negative examples. The decision boundary to the solid line can be clarified by using the unlabeled examples. Note that, the unlabeled examples are not taken into consideration. At the top right one can notice two positive labeled examples like noise or outliers.

4. Reinforcement Learning

Reinforcement learning refers to the process of learning from a set of reinforcements i.e., rewards or penalties. Here, the agent is solely responsible for selecting appropriate action prior to the reinforcement. An example of reinforcement learning is a waiter who learns that he did something wrong or fail to satisfy customer when he does not get a tip.

Q30. What is the task of supervised learning? Explain the four cases in which a function is fitted containing a unique variable to certain points?

Answer :

Task of supervised learning is to discover a function h which is used to obtain is true function f from a training set of n input-output pairs $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$. Here the value of y is obtained from a function of x i.e., $y = f(x)$.

The task of pure inductive inference is as follows,

Consider the set of examples of 'f' which results a function 'h' said to be as a hypothesis. It is used to determine that whether this 'h' value is a better appropriate value for 'f'. If the hypothesis is good then it will generate a better approximate value. This task is the most important problem of induction.

Consider the following figure which represents the four cases in which a function is fitted containing a unique variable to certain data points. Suppose, $(x, f(x))$ is a set of pairs where $x, f(x)$ represents the real numbers. Given the hypothesis as a set of polynomials having degree maximum of k. The set of polynomial are $5x^3 - 7$, $x^{15} - 3x^2$, $8x^6 + 3x^4$ and so on.

Case (i)

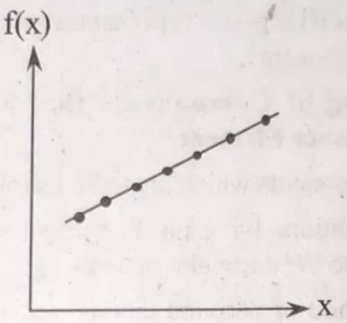


Figure: (x, f(x)) Pair with Consistent Linear Hypothesis

The above figure represents the polynomial of degree 1 which is fitted along a straight line. As the line satisfies for all the data then it is said to be consistent hypothesis.

Case (ii)

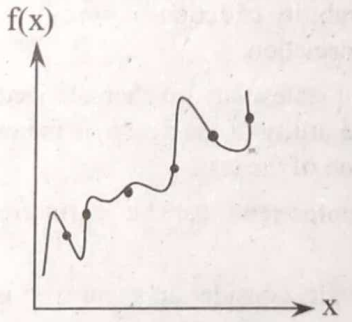


Figure: Consistent Line by Fitting Polynomial of Degree 7 for the Same Data Set

The above figure represents that a polynomial of higher degree is fitted. The line also represents as consistent for the same data.

Case (iii)

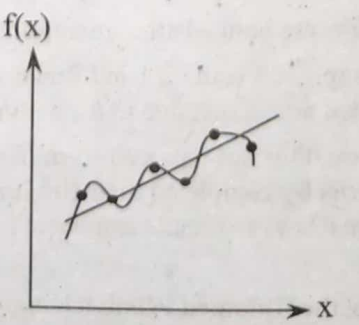


Figure: Line Fit with a Polynomial of Degree 6 for the Different Data Set

The above figures represent the second data set. The line is not consistent and it is formed with polynomial of degree 6 so as to fit exactly. Since there are seven data points, the polynomial contains the same parameters. Hence, any pattern cannot be found from this data and generalization cannot be done perfectly.

Case (iv)

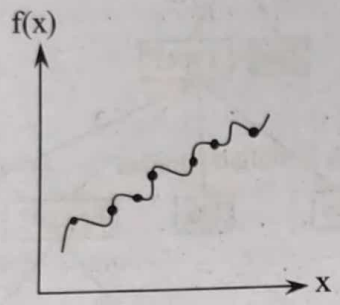


Figure: Fitting with Sinusoidal Function of Same Data Set

The above figure represent that the line is fit for the same data set as in case (iii) using the simple function specified as $ax + b + c\sin x$. This case gives more importance regarding the selection of hypothesis space. These space includes the polynomials with finite degree which are unable to specify the sinusoidal functions exactly. So the learner is not capable of learning from sinusoidal data using this hypothesis space. If the function is included in the hypothesis space then the learning problem is said to realizable or else it is unrealizable.

5.3.2 Learning Decision Trees

Q31. Discuss in brief about decision tree with example.

Answer : Model Paper-I, Q11(b)

Decision Tree

Decision tree is a tool for decision supporting which takes the form of tree structure making possible decisions by performing the set of tests. This tree usually takes a set of attributes in the form of input and generates the output value based on the input values. Each internal node represents the test that is performed on one of the properties. Furthermore, branches are assigned with certain values of the test and leaf nodes represent the values that must be returned after reaching that particular leaf.

Consider the following example that describes the problem regrading the waiting for a table in a restaurant. In this domain, the definition for the goal predicate must be learned. In order to learn this problem, the following attributes must be considered.

1. **Alternate:** Check for the availability of alternative restaurant to the nearby place.
2. **Bar:** Check if the restaurant contains a bar for waiting.
3. **Fri/sat:** The restaurant is opened for Friday or Saturday.
4. **Hungry:** Check if the people are hungry
5. **Patrons:** Check the number of people in the restaurant. It can represented as either none or some or full.
6. **Price:** The price value of the restaurant.
7. **Raining:** Check if it is raining
8. **Reservation:** Check if any reservation has been done.
9. **Type:** The restaurant can be of any type such as Indian, French, Italian etc.
10. **Wait Estimate:** The estimation of waiting by each host can be 0 – 5, 5 – 15, 15 – 60 and > 60 minutes.

The decision tree for the above domain is as follows,

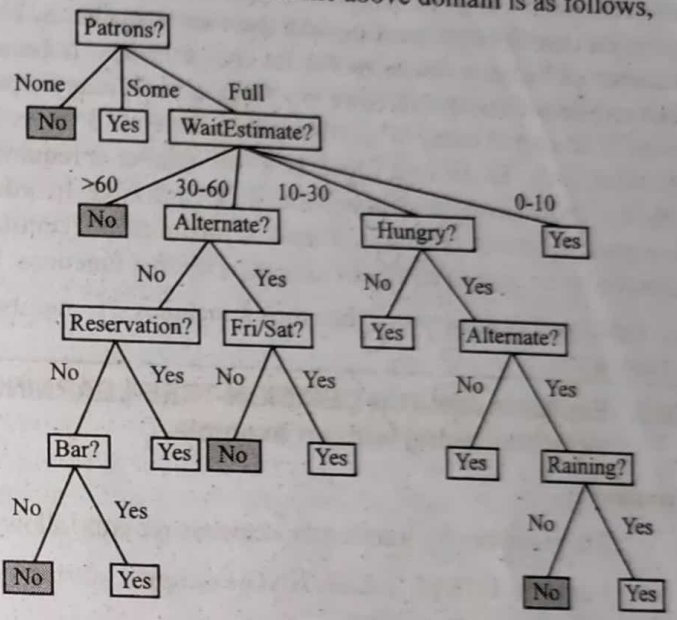


Figure: Decision Tree for Making Decision of Whether to Wait for a Table

In the above tree, the attributes such as price and type are not used since they are considered as irrelevant. The example is taken with Patrons = Full and WaitEstimate = 0 – 5 min that the people can wait for a table.

Q32. Explain briefly the expressiveness of decision trees.

Answer :

The statement used for representing the hypothesis of decision tree with the willwait goal predicate is as follows,

$$\forall m \text{ WillWait}(m) \Leftrightarrow (P_1(m) \vee P_2(m) \vee P_3(m) \ldots P_n(m))$$

In the above statement, the condition $P_i(m)$ represents conjuction of tests that are performed from the top-level (root) to bottom-level (leaf) of the decision tree resulting a positive outcome. This statement seems to be as a first order statement since it consists of only a single variable and each predicate is appeared to be unary. Furthermore, the relation between the goal predicate WillWait and the set of attributes that are taken as input values is described in the decision tree. But the decision trees are not represented for the tests that are performed with various objects. Decision trees can be expressed completely using the collection of propositional languages such that any boolean function can be used to represent in the form of a decision tree.

This can be accomplished by taking all the rows from the truth table for the function that correlates with the path of a tree. Due to this, the decision tree is represented in a large extent since there are many number of rows in the truth table. Also, they can represent multiple functions with a very small tree structure. The problem arises if the decision tree is represented for certain type of functions. For example, one of these functions include the parity function which is defined as the output value 1 if the number of inputs that are even and is equal to 1. For this purpose, a large decision tree must be required. Another function includes majority function which is defined as the output value 1 if the inputs that are greater than half is equal to 1. The representation of decision tree is good for certain type of functions and not good for certain functions. Consider there are n-attributes. The number of boolean functions for these n-attributes is based on the value of number of rows in a truth table. If every input entry is described using 'n' attributes then there are 2^n rows in the truth table. To define a function, 2^n bit number is required which is shown in the answer column of the truth table. In order to represent certain functions, atleast 2^n bits must be required irrespective of type of representation used for that functions. If the function is defined with 2^n bits then it includes 2^{2^n} number of functions.

Q33. Explain in detail the DECISION-TREE-LEARNING algorithm along with an example.

Answer :

The algorithm for learning the decision tree is as follows,

Function DTREE_LEARNING(examples, attributes, default) returns a DTree

Inputs: examples, collection of examples

attributes, collection of attributes

default, default value is taken for the goal predicate

If 'there is no example' then

return 'default' value

elseif 'the categorization is same for all the example'

then return 'the categorization'

else if 'there is no attribute' then

return MAX_VAL (examples)

else

best←SELECT_ATTRIBUTE(attributes, examples)

tree←a new DTree with root test best

m←MAX_VAL $examples_n$

for each value v_n of best do

$examples_n$ ←{elements of examples.with best=v_n}

subtree← DTREE_LEARNING($examples_n$, attributes_ best, m)

include a tree with a branch of label v_n and subtree 'subtree'

return tree

The algorithm which produces the final tree is induced with 12-examples data set which is shown as follows,

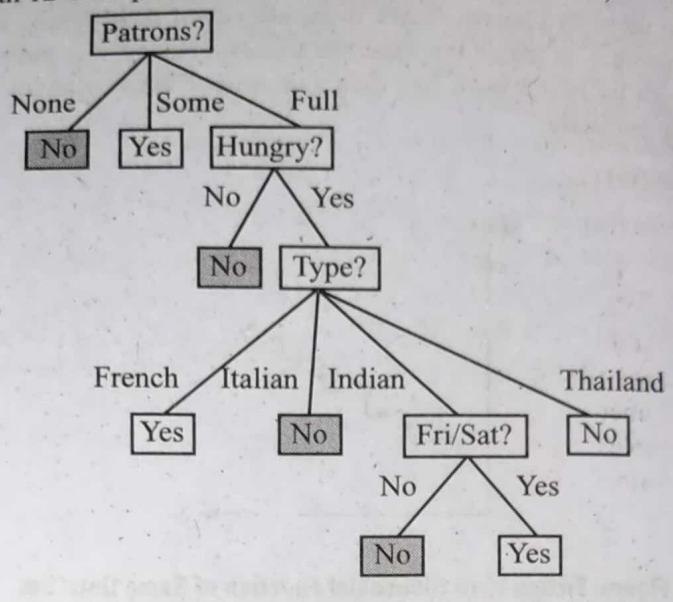


Figure: Decision Tree Induced with 12-example Data Set

The above tree is generalized from the original tree. The learning algorithm observes only the examples but not the exact function. However the hypothesis i.e., above tree satisfies all the examples.

In the above tree, the algorithm does not use the tests for 'Raining' and 'Reservation'. This is because, the algorithm can generalize all the examples without using both of them. Also, it is observed from the tree that during the Fridays and Saturdays, the first author is waiting for the 'Indian food'.

If the tree is induced with more examples then it will be similar to the original tree. However, in the above tree, even though there is a waiting of 0 – 10 minutes, the restaurant seems to be full.

Q34. Explain briefly the procedure of selecting attribute tests.

Answer :

In the procedure of selecting attributes, the formal measure is needed in which its value should be high. When the attribute is always good and its value is less, then the attribute is useless at a particular point of time.

The best example for this measure includes the expected amount of information that is provided by the attribute. For example, in order to decide if a coin returns a head, some knowledge should be required so as to answer that question whether it is a head. If the knowledge is less then more amount of information should be provided.

In case of information theory, the information can be measured in terms of bits. To decide if the answer is yes or no, only one bit of information is sufficient for certain questions.

For example, if v_n represents the number of possible answers with probabilities $P(v_n)$ then the amount of information (I) required for answering the correct answer is,

$$I(P(v_1)), P(v_2), ..., P(v_p) = \sum_{n=1}^{P} -P(v_n)\log_2 P(v_n)$$

The above equation is checked while tossing a coin, then we get,

$$I\left(\frac{1}{2},\frac{1}{2}\right)=\frac{-1}{2}\log_2\frac{1}{2}-\frac{1}{2}\log_2\frac{1}{2}=1\text{-bit}$$

If the probability of getting the head is 99% then $I\left(\frac{1}{100},\frac{99}{100}\right)=0.08$ bits of information is required and if the probability is 1 then no information is required (i.e., 0).

For example, consider the case of decision tree learning. Here, the question is what is the correct classification. The probability of getting the possible answers before the testing of any attribute is equal to the positive and negative examples contained in the trained set. Consider there are 'm' positive examples and 'n' negative examples in the training sets. Then the amount of information required to give a correct answer is,

$$I\left(\frac{m}{m+n},\frac{n}{m+n}\right)=\frac{-m}{m+n}\log_2\frac{m}{m+n}-\frac{n}{m+n}\log_2\frac{n}{m+n}$$

If $m=n=6$ then only 1 bit of information is required to give a correct answer when a test is performed on one particular attribute say A. This test will not decide how much information is required but it will provide some quantity of information. The amount of information required after the testing can be measured. If the training set S is divided into subsets S_1, S_2, S_3 ... S_n using the attribute A based upon its distinct n values then every subset S_i contains 'm_i' number of positive examples and 'n_i' number of negative examples. Hence, while going along that branch, extra information $I\left(\frac{m_i}{m_i+n_i},\frac{n_i}{m_i+n_i}\right)$ bits of information is required in order to answer the question.

Suppose i^{th} value of attribute is selected from the training set with probability $\frac{m_i+n_i}{m+n}$ then after the testing of attribute A, the amount of information required for classification will be,

$$\text{Remainder }(A)=\sum_{i=1}^{n}\frac{m_i+n_i}{m+n}I\left(\frac{m_i}{m_i+n_i},\frac{n_i}{m_i+n_i}\right)$$

The value of information gain can be calculated by subtracting the new amount of information from the original information as follows,

$$\text{Gain }(A)=I\left(\frac{m}{m+n},\frac{n}{m+n}\right)-\text{Remainder (A)}$$

Hence, the attributes which are containing the largest gain can be selected.

Q35. Define overfitting problem. Discuss in brief the technique employed for handling this problem.

Answer :

Overfitting problem generally occurs when there are more number of possible hypothesis. It also occurs if the target function is not in the random order. This problem not only affects every learning algorithm but also affects each decision tree.

The following are the two techniques that are employed for handling this problem.

1. Decision Tree Pruning

In the working of decision tree pruning, the recursive division of attributes are prevented provided that the attributes are irrelevant. This happens even if the classification is not done uniformly for the data available at each node of a tree.

Consider a set of examples that are divided by using irrelevant attribute. Due to such division, each subset of example is considered to be contained with equal proportions when compared with the original example set. During this situation, the value of information gain will be approximately zero. Hence, the attributes are said to be irrelevant as there is an information gain.

Statistical significance test is performed to determine the amount of information gain that is required for dividing particular attribute. In the initial stage of test, it is considered that underlying pattern does not exist i.e., null hypothesis. Further the actual data is analyzed for computing the extension in which it is deviated from unavailable pattern. If the extension of deviation is very less, then it acts as better evidence for determining the availability of significant pattern from the data. Here, the probabilities are competed by taking the standard distributions of certain extent of deviation.

At this situation, the null hypothesis is considered to be irrelevant attribute where the information gain for a large sample is absolutely zero. The probability is calculated under the null hypothesis where the sample of sizes are taken which results in observed deviation from positive and negative examples that are distributed expectedly. The deviation is measured by comparing the actual $\hat{m}_i$ number of positive examples and $\hat{n}_i$ number of negative examples present in each subset m_i and n_i by considering true irrelevance.

$$\hat{m}_i=m\times\frac{m_i+n_i}{m+n},\ \hat{n}_i=n\times\frac{m_i+n_i}{m+n}$$

The measure that is calculated for the whole deviation is,

$$D_w=\sum_{i=1}^{S}\frac{(m_i-\hat{m}_i)^2}{\hat{m}_i}+\frac{(n_i-\hat{n}_i)}{\hat{n}_i}$$

The distribution of D_w value is done under the null hypothesis based on the distribution of χ^2 with $s-1$ degrees of freedom. The computation is done for finding the probability that the attribute is irrelevant. This can be done using the standard χ^2 tables or using statistical software.

2. Cross-validation

It is another technique used for handling the problem of overfitting. This technique can be applied not only for every learning algorithm but also for decision tree learning. In this technique, the determination is done for the unavailable data which is predicted by each hypothesis. This can be accomplished by keeping some quantity of known data at other place and it is used for testing the prediction performance of a hypothesis that can be obtained from the remaining data. K-fold cross validation is referred as running the k experiments by keeping $\frac{1}{k}$ amount of data aside for the purpose of testing. Later, the obtained results are taken for calculating the average value. When $k=n$ then it is said to be as leave-one-out cross validation.

It is used along with any tree construction method for the purpose of choosing a tree that has a better prediction performance. This performance is again measured using new test set in order to prevent peeking.

Q36. Discuss in brief the issues that are to be considered for extending the applicability of decision trees.

Answer :

The issues that are to be considered for extending the applicability of decision trees are as follows,

1. Missing of Attribute Values

Some of the attribute values are missed for each example available in multiple domains. The reason for such a miss is due to the reason that they may not be recorded or they are of high cost. Due to this situation, the following two problems arise.

(a) The object cannot be classified as one of the test attribute is not available in the complete decision tree.

(b) The information gain formula cannot be modified as some of the attributes are missing with their values in certain examples.

2. Attributes with Multiple Values

The measure of information gain provides improper signal towards the usefulness of the attributes as they contain multiple values.

3. *Input Attributes with Continuous and Integer Values*

These type of attributes may include 'height' and 'weight' which contain unlimited number of values. In decision tree learning algorithms, it is better to determine the split point instead of producing innumerable branches to the nodes. This situation leads in high information gain.

4. Output Attributes with Continuous Values

If the price value of particular task is declared in advance instead of discrete classification then there must be a requirement of regression tree. In this type of tree, every leaf contains a linear function that includes subset of numerical attributes instead of containing a single value. The learning algorithm should declare regarding the ending of splitting and starting of the application of linear regression by taking the advantage of other attributes.

5.3.3 Knowledge in Learning

5.3.3.1 Logical Formulation of Learning

Q37. Explain about logical formulation of hypothesis.

Answer :

Model Paper-II, Q11(b)

Hypothesis can be formulated in terms of logical sentences. Moreover, descriptions and classifications can also represented in terms of logical sentences. Such an approach can be followed for incremental construction of hypotheses where a single sentence is considered at a time. The knowledge gained during past classifications can be used in new examples. Use of logical formulation avoids several learning-based issues.

A set of examples satisfying the definition of candidates is obtained by each hypothesis predication which is referred as predicate extension. If two hypothesis have distinct extensions, they are considered as inconsistent with each other. The reason for this is that they disagree with respect to the predictions on one or more examples. However, they are considered as equivalent if they posses same extension.

An inconsistent hypothesis can result from the following possibilities,

- A false negative example where an example is positive but the hypothesis considers it as negative.
- A false positive example where an example is negative but the hypothesis considers it as positive.

In both the situations, the hypothesis is considered as inconsistent. To make the learning process effective, the inconsistent hypotheses are removed.

Current-best Hypothesis

For answer refer Unit-V, Q38.

Q38. Explain in detail the current-best hypothesis search along with its algorithm.

Answer :

Current-best Hypothesis

Current-best hypothesis is used to maintain exactly one hypothesis. It is also used to accommodate hypothesis when new examples are made so that the examples are consistent with already existing ones.

Example

Consider a hypothesis Hyp and a new example. If the example is consistent then no action is taken. However, if the new example is a false negative then actions are taken. The scenarios of different examples are shown in figure (a).

Figure (a) shows hypothesis as a region in which examples are depicted as positive (+) and negative signs.

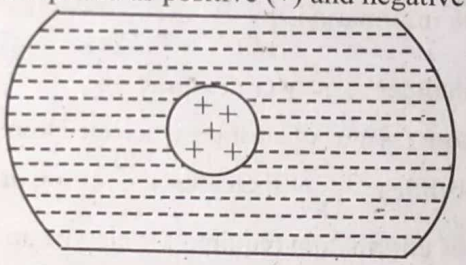


Figure 1(a): A Hypothesis that is Consistent

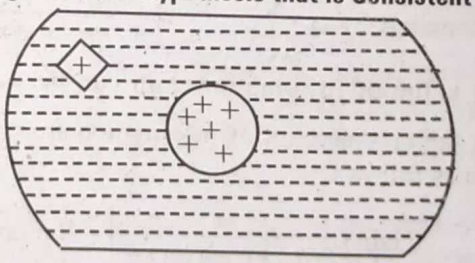


Figure 1(b): A Negative False Hypothesis

Figure 1(b) shows a new example  which is a false negative. According to the hypothesis, it should be negative but it is a positive. Therefore hypothesis must be extended to include the new example. This is called generalization.

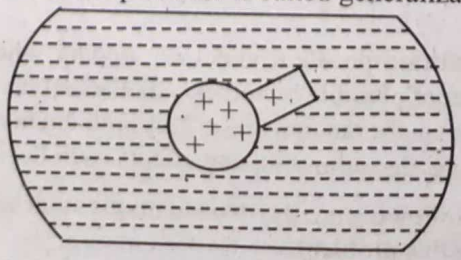


Figure 1(c): A Generalized Hypothesis

Figure 1(c) shows one form of generalization of false negatives.

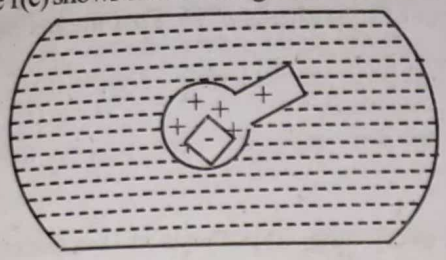


Figure 1(d): A Positive False Hypothesis

Figure 1(d) shows a false positive  . According to the hypothesis, the new example  must be positive but it is negative. Therefore the hypothesis must be reduced in order to remove it. This process is called specialization.

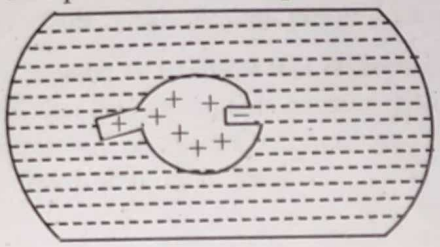


Figure 1(e): A Specialized Hypothesis

Figure 1(e) shows specialized form of the hypothesis .

The above figures show different scenarios of new example. Moreover, relation between hypothesis such as more specific than and 'more general than' offer hypothesis space which make search process more efficient. After generalization and specialization process, following algorithm can be applied.

Current-best Learning Algorithm

Function Cur-best-lrg.(examples, Hyp) return a hypothesis or fail.

Hyp ← a hypothesis that is consistent with first example

If examples is empty then return Hyp

If a is false positive for hypothesis

then

Hyp ← select a specialization of Hyp that is consistent with examples.

else if e_1 is false negative for hypothesis

then

Hyp ← select a generalization of Hyp that is consistent with examples.

If consistent generalization or specialization are not found

then fail

return Hyp

Figure (2): Algorithm of Current Best Learning

Generalization and specialization are logically related in hypothesis. If hypothesis Hyp_1 with definition def_1 is a generalized form of hypothesis Hyp_2 with definition def_2 then

$$\forall K\ def_2(k) \Rightarrow def_1(k)$$

Furthermore, if Hyp_2 needs to be generalized then def_1 that is logically implied by def_2 needs to be found. This can be done as follows,

Suppose,

If def_2 (k) is alternate (k) ∧ Dancers (k, some)

then it will be generalized as

def_1 (k) = Dancers (k some)

This process is called dropping conditions.

In general, this algorithm generates a weak definition which results in allowing many positive examples. Moreover, a hypothesis can be specialized if disfunction for a disjunctive definition is removed or if extra conditions are added to its candidate definition.

Furthermore, in this algorithm many possible generalizations and specializations can be applied. However, when a choice is made, it may not result in simple hypothesis or it may result in a non recoverable condition where it becomes inconsistent with data. If such situation arises then the program must revert to old state.

Moreover, this algorithm is applied in several machine learning systems with some modifications which may result in complexities such as,

1. If a change is to be made then its previous instances must be checked making it a very expensive choice.
2. The search need back tracking multiple times.

Q39. Explain in detail about version spaces and candidate elimination algorithm.

Answer :

The purpose of CANDIDATE-ELIMINATION algorithm is to generate a description for set of hypothesis compatible with training examples. It evaluates description of this set without computing its members. This is established through a more general than partial ordering to with old compact compatible hypothesis representation and to refine the representation of training example occurrences. Let the hypothesis be compatible with training examples if examples are correctly classified. It is said to be compatible only when $h(x) = c(x)$ for every example $<x, c(x)>$ in D.

Consistent(h, D) = ($\forall <x, c(x)> \in D$) $h(x) = c(x)$

The CANDIDATE-ELIMINATION algorithm represents set of all hypothesis compatible with observed training examples. Such type subset of hypothesis is known as version space in terms of hypothesis space H and training examples D since it holds all the possible versions of target concept.

$$VS_{H,0} \equiv \{h \in H \mid \text{Compatible (h, D)}\}$$

Version space can be represented by listing all of its members. This gives rise to a simple algorithm called LIST-THEN-ELIMINATE algorithm. It begins the version space to hold the hypothesis in H and then remove the hypothesis found incompatible with training example. It may decrease with an increase in examples until one hypothesis compatible with all examples is found.

1. Version space ← List of hypothesis in H.
2. For every training example, <x, c(x)> eliminate any hypothesis h for which h(x) ≠ c(x) from version space.
3. Generate list of hypothesis in version space.

The CANDIDATE-ELIMINATION algorithm evaluates version space holding hypothesis from H compatible with observed sequence of training examples. It starts by initializing version space to set of hypothesis in H. This means by initializing G bounding set to hold most general hypothesis in H.

$G_0 \leftarrow \{<?, ?, ?, ?, ?, ?>\}$

and initializing S boundary set to hold most specific hypothesis

$S_0 \leftarrow \{<\phi, \phi, \phi, \phi, \phi, \phi>\}$

The above boundary sets determine the complete hypothesis space since every other hypothesis in H is general than S_0 as well as specific than G_0. The S and G boundary sets can be generalized and specialized respectively for removing from version space, the incompatible hypothesis.

The computed version space only holds the hypothesis which are compatible.

Assign set of maximally general hypothesis in H to G

Assign set of maximally specific hypothesis in H to S

For each training example x do,

- If x is positive example
- Eliminate incompatible hypothesis with x from G
- For each hypothesis s in S which is incompatible with x
- Eliminate s from S
- Add minimal generalizations h of s to S such that
- h is compatible with x and some member of G is more general than h
- Eliminate hypothesis that is more general than other hypothesis in s from S
- If x is a negative example
- Eliminate hypothesis incompatible with x from S
- For each hypothesis g in G which is incompatible with x
- Eliminate g from G
- Add minimal specializations h of g to G such that
- h is compatible with x and some other member of S more specific than h
- Eliminate hypothesis that is less general than other hypothesis in G

The CANDIDATE-ELIMINATION algorithm is specified in terms of operations like evaluation of minimal generalizations nd specializations of hypothesis and there by identifying nominal and non-maximal hypothesis. The algorithm is applicable to ny concept learning task and hypothesis space with well defined operations.

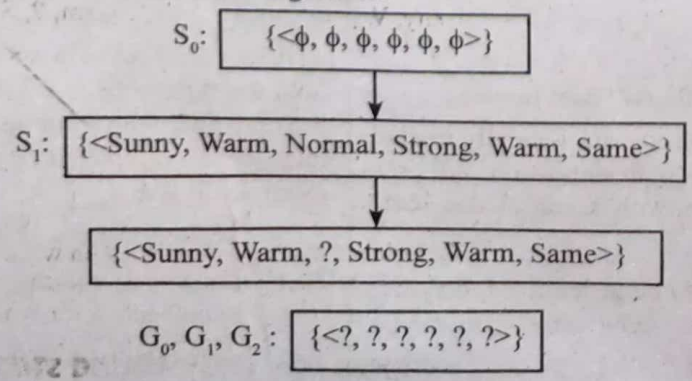
Q40. Derive an example to explain the working of candidate elimination algorithm.

Answer :

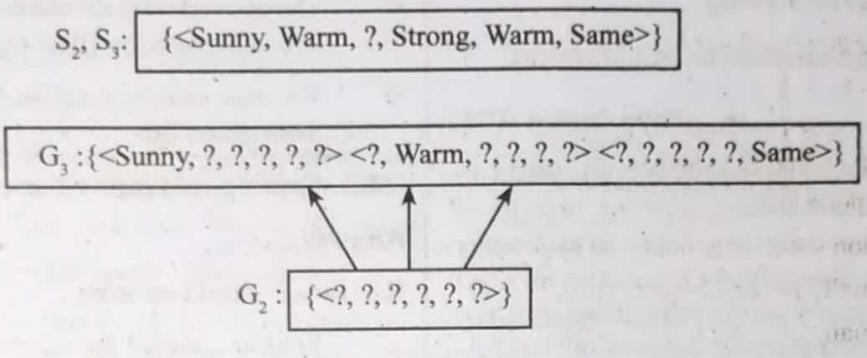
Consider the below example,

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1.	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2.	Sunny	Warm	High	Strong	Warm	Same	Yes
3.	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

The below figure traces the candidate elimination algorithm.

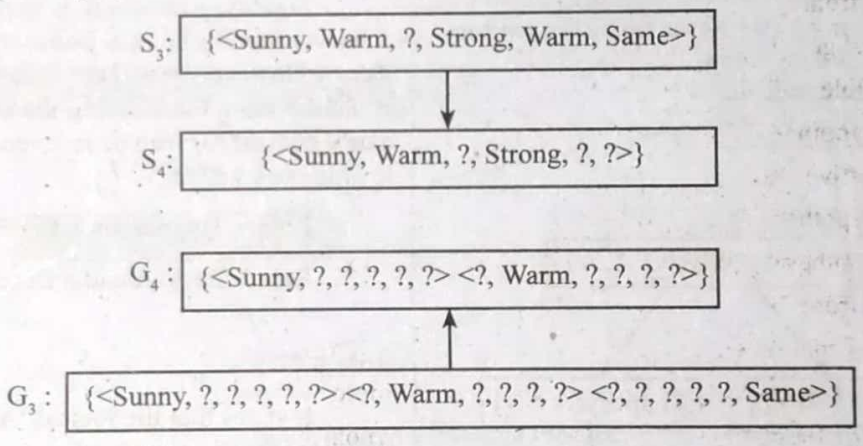


The boundary sets are initialized to G_0 and S_0. For the training example <Sunny, Warm, Normal, Strong, Warm, Same>, EnjoySport = Yes, the candidate elimination algorithm checks S boundary and then finds that it is specific and cannot cover the positive example. The boundary is updated by moving it to least general hypothesis which covers new example. The second training example <Sunny, Warm, High, Strong, Warm, Same>, EnjoySport = Yes is similar to effect of generalizing S followed by S_2 without updating G. Consider the third training example,



<Rainy, Cold, High, Strong, Warm, Change>, EnjoySport = No

This is a negative example that represents that G boundary of version space is general. The hypothesis in G will not predict accurately that new example is positive. The hypothesis in G boundary need to be specialized until it classifies new negative example accurately. The alternative minimally more specific hypothesis in above figure are members of new G_3 boundary set. Consider the fourth training example,

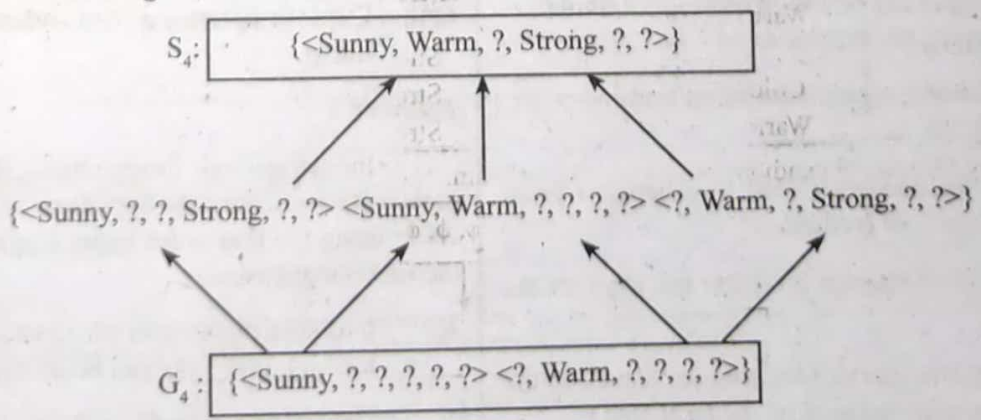


Training Example

4. <Sunny, Warm, High, Strong, Cool, Change>, EnjoySport = Yes

This fourth example will generalize the S boundary of version space. It removes a member of G boundary since it cannot cover new positive example. The hypothesis needs to be dropped from G boundary by deleting the branch of partial ordering from version space of hypothesis. The boundary sets S_4 and G_4 after processing the four examples will delimit the version space of hypothesis that are consistent with set of incrementally observed training examples.

This is depicted in below figure,



The learned version space in above figure is not dependent on sequence based on which training examples are represented. The S and G boundaries are moved closer by delimiting a smaller version space of candidate hypothesis.

5.3.3.2 Knowledge in Learning, Explanation-based Learning, Learning Using Relevance Information, Inductive Logic Programming

Q41. Explain about explanation-based learning.

Answer : Model Paper-III, Q11(b)

The process of extracting general rules from distinct observations is called as explanation-based learning. To do this, it initially utilizes existing knowledge to generate an explanation regarding an observation. It then defines a list of cases for which the generated explanation can be used. Such a definition acts as a foundation for the rule that covers all the cases of class.

An explanation generated can be a logical proof or a process offering solution to a specific problem. Here, the steps involved in the process are defined in a well organized form. The reason for such a detailed definition is to apply the same process in some other cases. For instance, consider an example of simplifying the expression $1 \times (0 + A)$ the simplification parse tree for this expression is shown below,

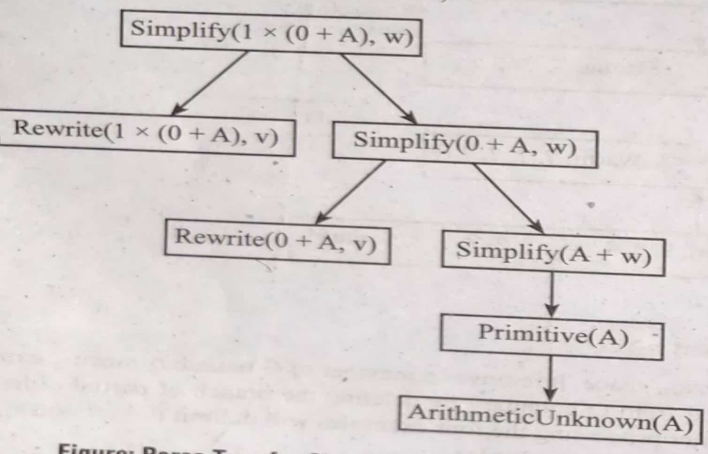


Figure: Parse Tree for Simplification of 1 × (0 + A)

A similar approach can be followed to solve similar problems like simplifying the expression $(x \times (y + z))$.

The basic process of explanation based learning is as follows,

The available knowledge is used to generate a proof applicable for the goal predicate.

A proof tree is constructed covering the steps of the original proof.

A new rule is then generated keeping the leaves on the left side and variabilized goal on the right side.

The true conditions are removed from the left-hand side irrespective of the values of variables.

The efficiency of explanation-based learning can be analyzed based on the following factors,

- The reasoning process gets effected with respect to its speed when large number of rules are added.
- The derived rules should be improved in terms of speed to compensate the effect on reasoning process.
- The derived rules must be declared as general to utilize them generally.

Q42. Explain relevance-based learning.

Answer :

Relevance-based Learning

Relevance-based learning scheme makes use of prior knowledge and identifies relevant attributes. It also reduces generated hypothesis space. It does not derive the new information from the beginning hence it is a deductive form of learning.

Example

Suppose a person Bob visits Kenya and meets a person John, who speaks French. Bob assumes that French is spoken in Kenya. However, he will not assume that every person in kenya is named John. Considering the example and its observation, a new general rule can be inferred. This rule describes the following observation,

Rule 1: Hypothesis $\wedge$ Description $\Rightarrow$ Classification

Rule 2: Background $\wedge$ Description $\wedge$ Classification $\Rightarrow$ Hypothesis

Rule 1

It states that the logical 'AND' of Hypothesis and description implies classification i.e., Hypothesis $\wedge$ Description $\Rightarrow$ Classification.

Rule 2

It describes that "Logically ANDing" of background, description and classification implies hypothesis.

i.e., Background $\wedge$ Description $\wedge$ Classification $\Rightarrow$ Hypothesis

Q43. Explain in brief about inductive logic programming.

Answer :

Inductive Logic Programming (ILP) is an integrated approach of representing the hypothesis in terms of logic programs while using the first-order logic. It is widely used because of the following reasons,

1. It offers a platform in which knowledge-based inductive learning problems can be managed rigorously.
2. General first order theories can be induced from examples which is a tedious task for attribute-based algorithms.
3. It generates easily understandable hypotheses.

ILP algorithms can also generate new predicates to assist in expressing the explanatory hypotheses. These algorithms are referred as constructive induction algorithms. Such algorithms play a vital role in cumulative learning which is considered as the most complex problem of machine learning.

There are two approaches to ILP. They are,

1. Top-down Inductive Learning
2. Inductive learning with inverse deduction.

1. Top-down Inductive Learning

Top-down ILP approach considers a general rule and modifies it accordingly according to the data. This approach is similar to decision tree learning in which the tree is grown gradually until it offers consistency with respect to the observations. In this approach, first-order literals are used as attributes and hypothesis is used as decision tree.

2. Inductive Learning with Inverse Deduction

Inductive learning with inverse deduction approach follows the concept of inverting deductive proof. Here, inverse resolution concept is used where a fact is proven by resolution. It involves a search process where each of the step involved in inverse resolution is non-deterministic.

An example of inverse resolution process is shown below,

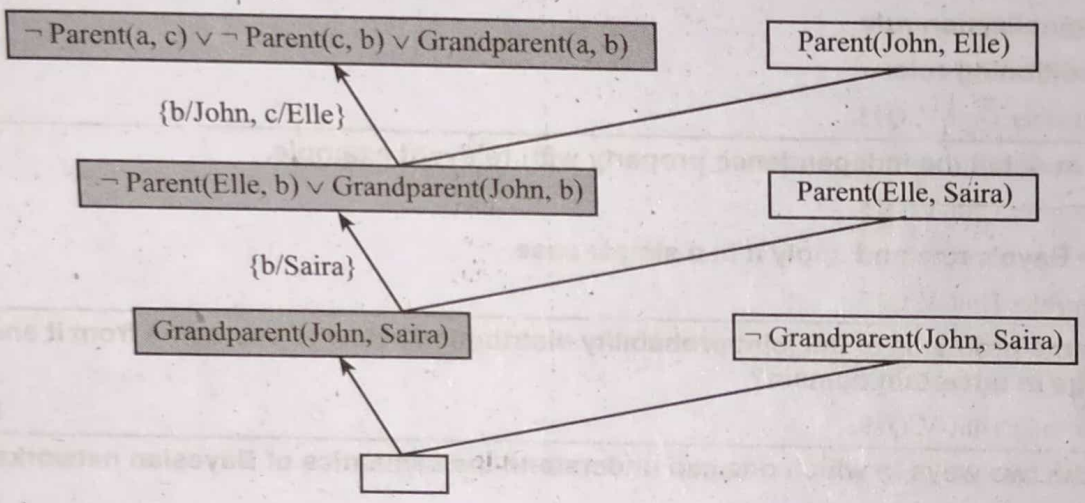


Figure: Inverse Resolution Process

IMPORTANT QUESTIONS

SHORT QUESTIONS

Q1. Write about marginalization rule.

Ans: For answer refer Unit-V, Q2. Important Question

Q2. What are the properties of Bayesian networks?

Ans: For answer refer Unit-V, Q3. Important Question

Q3. Define,

(a) Conditional Probability Table (CPT)

(b) Conditional Case.

Ans: For answer refer Unit-V, Q4. Important Question

Q4. Define deterministic node.

Ans: For answer refer Unit-V, Q6. Important Question

Q5. What are the issues considered while designing a learning element?

Ans: For answer refer Unit-V, Q8. Important Question

ESSAY QUESTIONS

Q6. Discuss in brief about,

(i) Utility theory

(ii) Decision theory

(iii) Decision theoretic agent.

Ans: For answer refer Unit-V, Q10. Important Question

Q7. Write short notes on,

(i) Probability

(ii) Evidence.

Ans: For answer refer Unit-V, Q11. Important Question

Q8. State and explain the conditional probability. Important Question

Ans: For answer refer Unit-V, Q13.

Q9. Write short notes on, Important Question

(i) Marginalization rule

(ii) Conditioning rule.

Ans: For answer refer Unit-V, Q15.

Q10. Discuss in detail the independence property with relevant example. Important Question

Ans: For answer refer Unit-V, Q16.

Q11. State the Baye's rule and apply it in a simple case. Important Question

Ans: For answer refer Unit-V, Q17.

Q12. What are the problems of full joint probability distribution? How to overcome from it and represent the knowledge in uncertain domain? Important Question

Ans: For answer refer Unit-V, Q19.

Q13. Explain the two ways in which one can understand the semantics of Bayesian networks. Important Question

Ans: For answer refer Unit-V, Q20.

Q14. Define deterministic node with suitable example. Also discuss about conditional distribution in Bayesian network with continuous variables and hybrid Bayesian network. Important Question

Ans: For answer refer Unit-V, Q21.

Q15. Explain the need of approximate inference in Bayesian networks. Also, discuss the direct sampling methods. Important Question

Ans: For answer refer Unit-V, Q22.

Q16. What are the problems associated with first-order models? How to overcome from it using relational probability models. Important Question

Ans: For answer refer Unit-V, Q24.

Q17. Explain in detail about the Dempster-Shafer theory. Important Question

Ans: For answer refer Unit-V, Q27.

Q18. Explain in detail the different issues considered while designing a learning element. Important Question

Ans: For answer refer Unit-V, Q28.

Q19. Discuss in brief about decision tree with example. Important Question

Ans: For answer refer Unit-V, Q31.

Q20. Explain about logical formulation of hypothesis. Important Question

Ans: For answer refer Unit-V, Q37.

Q21. Explain about explanation-based learning. Important Question

Ans: For answer refer Unit-V, Q41. Important Question