CBCS SCHEME

USN | 2 | V | D | 1 | 8 | C | S | 0 | S | 1 |

18CS71

Seventh Semester B.E. Degree Examination, Feb./Mar. 2022
Artificial Intelligence and Machine Learning

Time: 3 hrs. Max. Marks: 100

Note: ***Answer any FIVE full questions, choosing ONE full question from each module.***

Module-1

1 a. List all task domains of Artificial Intelligence. **(06 Marks)**

b. Explain Minimax procedure of tic – tac – toe. **(07 Marks)**

c. List all production rules for the water jug problem. **(07 Marks)**

OR

2 a. Illustrate Slot – and – filler structure method in Question and Answering system. **(06 Marks)**

b. Explain Hill climbing issues which terminates algorithm without finding a goal state or getting to a state from which no better state can be generated. **(04 Marks)**

c. Apply AO* algorithm for the following graph and find final path. **(10 Marks)**

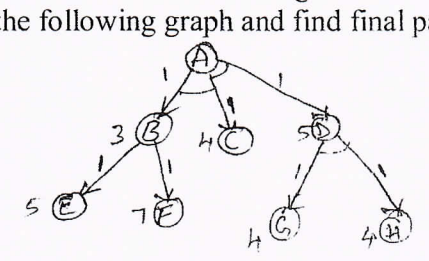


Fig. Q2(c)

Module-2

3 a. Convert the following statement into its Equivalent Predicate Logic from

i) Marcus was a man ii) Marcus was a Pompeian

iii) All Pompeians were Romans iv) Caesar was a Ruler

v) All Romans were either loyal to Caesar of hated him.

vi) Everyone is loyal to someone

vii) People only try to assassinate rulers they are not loyal to.

viii) Marcus tried to assassinate Caesar. **(08 Marks)**

b. List the issues on Knowledge representation. **(05 Marks)**

c. Construct maximally specific hypothesis for the following training examples. **(07 Marks)**

Example	Sky	Air Temp	Humidity	Wind	Water	Forecast	Enjoy Sport
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

OR

4 a. Apply Candidate Elimination algorithm for the dataset given above (Question 3(c)). How do you classify following new instance from the set of hypothesis obtained by Candidate Elimination algorithm? **(12 Marks)**

Instance	Sky	Air Temp	Humidity	Wind	Water	Forecast	Enjoy Sport
A	Sunny	Warm	Normal	Strong	Cool	Change	?
B	Rainy	Cold	Normal	High	Warm	Same	?

b. What are Horn Clauses? Write a declarative and a procedural representation. List syntactic difference between Logic and PROLOG. **(08 Marks)**

Module-3

5 a. Construct decision tree using ID3 algorithm for the following data : **(12 Marks)**

Day	Outlook	Temp	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	Yes
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	No
5	Rain	Cool	Normal	Weak	Yes

b. Derive Gradient descent rule. **(08 Marks)**

OR

6 a. Give decision tree to represent the following Boolean functions :

i) $A \wedge \neg B$ ii) $A \vee [B \wedge C]$ iii) A XOR B iv) $[A \wedge B] \vee [C \wedge D]$. **(08 Marks)**

b. Explain Perceptron with appropriate diagram Represent AND Boolean function using Perceptron. **(04 Marks)**

c. Write Back propagation algorithm. **(08 Marks)**

Module-4

7 a. A patient takes a lab test and the result comes back positive. The test returns a correct positive result in only 98% of the cases in which the disease is actually present and a correct negative result in only 97% of the cases in which the disease is not present. Further, 0.008 of the entire population have the Cancer. Does a patient have Cancer or not? **(10 Marks)**

b. Derive Brute force MAP learning and also mention assumption made in this process. **(10 Marks)**

OR

8 a. Explain Minimum Description Length Principle (MDL). **(06 Marks)**

b. Explain Naïve Bayes classifier and Bayesian belief Networks. **(08 Marks)**

c. Write EM algorithm. **(06 Marks)**

Module-5

9 a. Explain K – NN algorithm. **(06 Marks)**

b. Explain steps of Locally Weighted Linear regression. **(07 Marks)**

c. Describe Radial basis function with appropriate diagram. **(07 Marks)**

OR

10 a. Illustrate the basic concept of Q – learning using Simple Deterministic World. **(10 Marks)**

b. Explain Q – Learning algorithm. **(10 Marks)**

* * * * *

0 — Artificial Intelligence & Machine Learning :-

Feb/March-2022 Solution

Module-1

Q1 a) List all task domains of Artificial Intelligence.

soln

Mundane Tasks

→ Perception
- vision
- speech

→ Natural language
- understanding
- Generation
- Translation

→ Common Sense reasoning

→ Robot control

Formal Tasks

→ Games
- chess
- Backgammon
- Checkers-Go

→ Mathematics
- Geometry
- Logic
- Integral Calculus
- Proving properties of programs.

→ Expert Tasks

→ Engineering
- Design
- Fault Finding
- Manufacturing planning

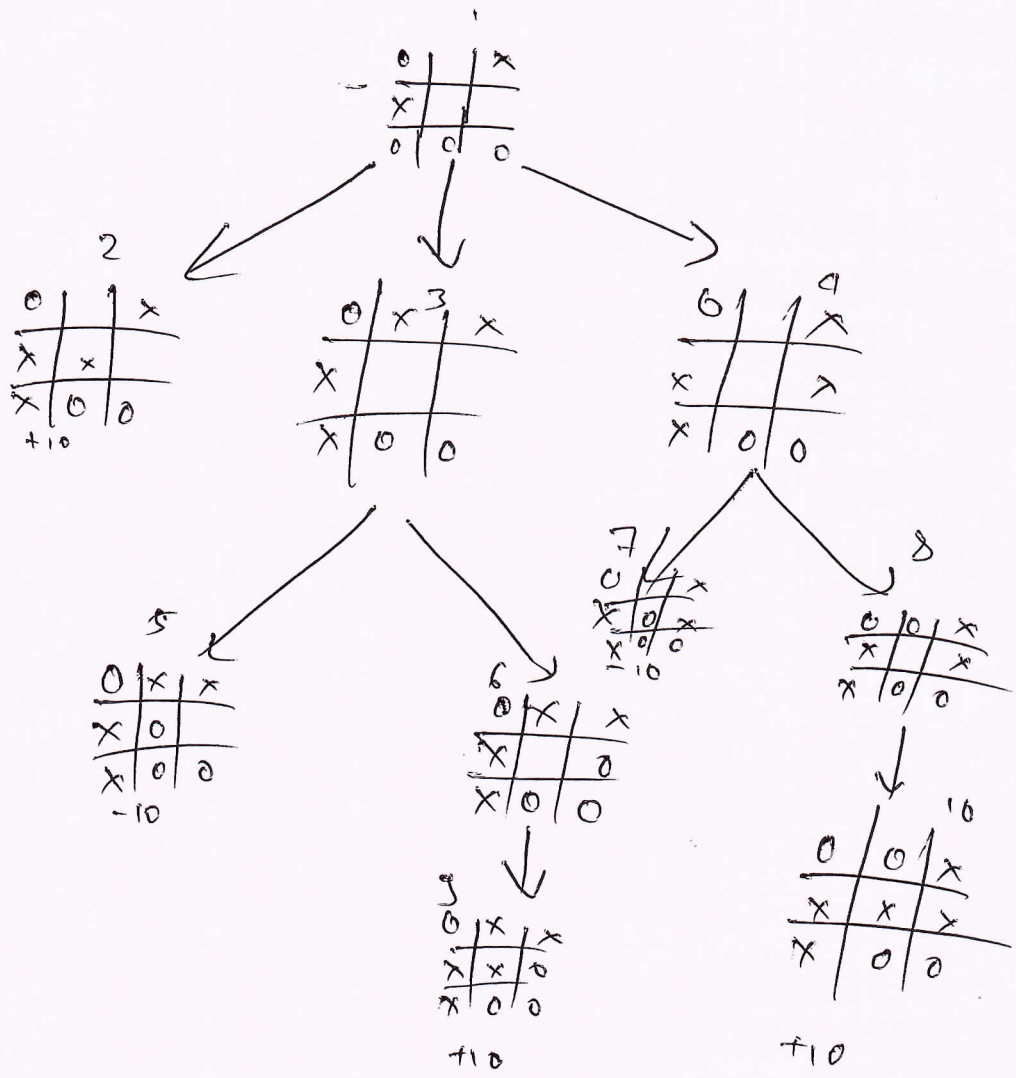
→ Scientific analysis

→ Medical diagnosis

→ Financial analysis

(2) Explain Minimax procedure of tic-tac-toe.

- The key to minimax algorithm is back & forth between two players, where the player whose "turn it is" desires to pick the move with the maximum score. In turn, the scores for each of the available moves are determined by the opposing player deciding which of its available moves has the minimum score.



Assume 'X' is the turn taking player.

- If the game is over, return the score from X's perspective.
- Otherwise get a list of new game states for every possible move.
- Create a scores list.
- For each of these states add the minimax result of that state to the scores list.
- If it's X's turn, return the maximum score from the list.
- If it's O's turn, return the minimum score from the list.

③ Q i)

i) List all the production rules for the water jug problem

production rules for water jug problem

①	(x,y) if $x<4$	$\rightarrow (4,y)$	Fill the 4-gallon jug.
②	(x,y) if $(y<3)$	$\rightarrow (x,3)$	Fill the 3-gallon jug.
③	(x,y) if $x>0$	$\rightarrow (x-d,y)$	Pour some water out of the 4-gallon jug.
④	(x,y) if $y>0$	$\rightarrow (x,(y-d))$	Pour some water out of the 3-gallon jug
⑤	(x,y) if $x>0$	$\rightarrow (0,y)$	Empty the 4-gallon jug on the ground.
⑥	(x,y) if $y>0$	$\rightarrow (x,0)$	Empty the 3-gallon jug on the ground
⑦	(x,y) if $x+y\geq 4 \& y>0$	$\rightarrow (4,y-(4-x))$	Pour water from the 3-gallon jug into 4-gallon jug until the 4 gallon jug is full.
⑧	(x,y) if $x+y\geq 3 \& x>0$	$\rightarrow (x-(3-y),3)$	Pour water from the 4-gallon jug into 3-gallon jug until the 3-gallon jug is full.
⑨	(x,y) if $x+y\leq 4 \& y>0$	$\rightarrow (x+y,0)$	Pour all the water from 3-gallon jug into 4-gallon jug
⑩	(x,y) if $x+y\leq 3 \& x>0$	$\rightarrow (0,x+y)$	Pour all the water from 4-gallon jug into 3-gallon jug.
⑪	$(0,2)$	$\rightarrow (2,0)$	Pour 2 gallons from the 3-gallon jug into 4-gallon jug.
⑫	$(2,y)$	$\rightarrow (0,y)$	Empty the 2 gallon in the 4-gallon jug on ground.

④ Q a)

Illustrate Slot & filler structure method in Question & Answering System.

Consider the sentence "She found a red one She really liked" can be represented as:

Event2:
instance : Finding
tense : Past
agent : Mary
Object : Thing 1

Thing 1
instance : Coat
Color : Red

Event 2
instance : Liking
tense : Past
Modifier : much
Object : Thing 1

The entities in the above representation derive their meaning from their connections to other entities.

Input Question - The input question in Character form

StructQuestion - A structured representation of the content of the user's question.

Algorithm:

- Convert the InputText into Structured form using knowledge contained in EnglishKnow.
- To answer a Question do:-
  i) convert question to structured form
  ii) Match the structured form against structured text.
  iii) Return as the answer those parts of the text that match the requested segment of question.

Q2 b) Explain Hill Climbing issues which terminates algorithm without finding a goal state or getting to a state from which no better state can be generated.

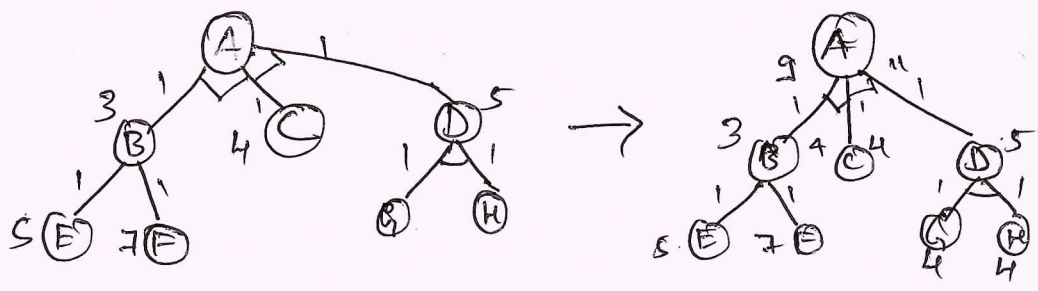
(7) Both basic & steepest - ascent hill climbing fail to find solutio Either algorithm may terminate not by finding a goal state bu by getting to a state from which no better states can be generated. This will happen if program has reached either a local maximum, a plateau or a ridge.

- A local maximum is a state that is better than all its neighbors but is not better than some other states farther away.

- A plateau is a flat area of the search space in whic a whole set of neighboring states have the same value.

- A ridge is a special kind of local maximum. It is an area of search space that is higher than surrounding areas & that itself has a slope.

(8) c) Apply AO* algorithm for the following graph & find final path.



(6) Module-2

Q3 a) Convert the following statement into its Equivalent Predicate Logic form.

i) Marcus was a man
- man(Marcus)

ii) Marcus was a Pompeian
- Pompeian(Marcus)

iii) All Pompeians were Romans.
- $\forall x$: Pompeian(x) $\rightarrow$ Roman(x)

iv) Caesar was a ruler
- ruler(Caesar)

v) All Romans were either loyal to Caesar or hated him
$\forall x$: Roman(x) $\rightarrow$ loyalto(x, Caesar) $\vee$ hate(x, Caesar)

vi) Everyone is loyal to someone
$\forall x$: $\rightarrow$ y: loyalto(x, y)

vii) People only try to assassinate rulers they are not loyal to.
$\forall x$: $\forall y$: person(x) $\wedge$ ruler(y) $\wedge$ tryassassinate(x, y) $\rightarrow$ $\neg$ loyalto(x, y)

viii) Marcus tried to assassinate Caesar
tryassassinate(Marcus, Ceasar)

Q3 b) List the issues on knowledge representation

① Important attribute
② Relationships among attributes
- Inverses

(7)
- Existence in an is a hierarchy
- Techniques for reasoning about values
- Single-valued attributes.

(5) Choosing the Granularity of Representation

(4) Representing Set of objects

(5) Finding the right structures as needed
- Selecting an initial structure
- Revising the choice when necessary

Q3 c) Construct maximally specific hypothesis for the following training examples

Example	Sky	Airtemp	Humidity	Wind	Water	forecast	Enjoy sport
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

→ $h \leftarrow \langle \phi, \phi, \phi, \phi, \phi, \phi \rangle$

$h \leftarrow \langle$Sunny, Warm, Normal, Strong, Warm, Same$\rangle$

$h \leftarrow \langle$Sunny, Warm, ?, Strong, Warm, Same$\rangle$

$h \leftarrow \langle$Sunny, Warm, ?, Strong, ?, ?$\rangle$

Q4 (a) Apply Candidate elimination algorithm for the dataset given above in Q.3(c).

S_0: $\langle \phi, \phi, \phi, \phi, \phi, \phi \rangle$

S_1 : $\langle$Sunny, Warm, Normal, Strong, Warm, Same$\rangle$

S_2 : $\langle$Sunny, Warm, ?, Strong, Warm, Same$\rangle$

G_0, G_1, G_2 : $\langle ?, ?, ?, ?, ?, ? \rangle$

(8)

S_2, S_3: $\langle$Sunny, Warm, ?, Strong, Warm, Same$\rangle$

G_3: $\langle$Sunny, ?, ?, ?, ?, ?$\rangle$ $\langle$?, Warm, ?, ?, ?, ?$\rangle$ $\langle$?, ?, ?, ?, ?, Same$\rangle$

G_2 $\langle ?, ?, ?, ?, ?, ? \rangle$

S_3 : $\langle$Sunny, Warm, ?, Strong, Warm, Same$\rangle$

S_4 : $\langle$Sunny, Warm, ?, Strong, ?, ?$\rangle$

G_4 : $\langle$Sunny, ?, ?, ?, ?, ?$\rangle$ $\langle$?, Warm, ?, ?, ?, ?$\rangle$

G_3 : $\langle$Sunny, ?, ?, ?, ?, ?$\rangle$ $\langle$?, Warm, ?, ?, ?, ?$\rangle$ $\langle$?, ?, ?, ?, ?, Same$\rangle$

Final version space

S_4 : $\langle$Sunny, Warm, ?, Strong, ?, ?$\rangle$

$\langle$Sunny, ?, ?, Strong, ?, ?$\rangle$ $\langle$Sunny, Warm, ?, ?, ?, ?$\rangle$ $\langle$?, Warm, ?, Strong, ?, ?$\rangle$

G_4 : $\langle$Sunny, ?, ?, ?, ?, ?$\rangle$ $\langle$?, Warm, ?, ?, ?, ?$\rangle$

Classify new instance

i) $\langle$Sunny, Warm, Normal, Strong, Cool, Change$\rangle$ ⇒ Yes

ii) $\langle$Rainy, Cold, Normal, High, Warm, Same$\rangle$ ⇒ No.

Q5 (b) What are Horn clauses? write a declarative & procedural representation. List syntactic difference between Logic & PROLOG.

A Horn clause is a clause containing at most one positive literal. A definite clause contains ~~atleast~~ exactly one positive literal. Ex:- [¬child, ¬mail, Boy]

A declarative representation is one in which knowledge is specified, but the use to which that knowledge is to be put is not given.

A procedural representation is one in which control information that is necessary to use the knowledge is considered to be embedded in knowledge itself.

Eg.

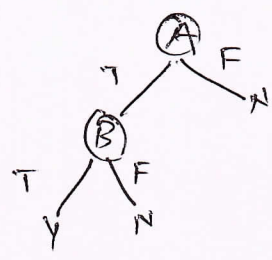
Declarative	Procedural
$\forall x: pet(x) \wedge small(x) \rightarrow apartment(x)$	apartment(x) :- pet(x), small(x)
$\forall x: (cat(x) \vee dog(x)) \rightarrow pet(x)$	pet(x) :- cat(x) pet(x) :- dog(x)

Syntactic difference between Logic & PROLOG

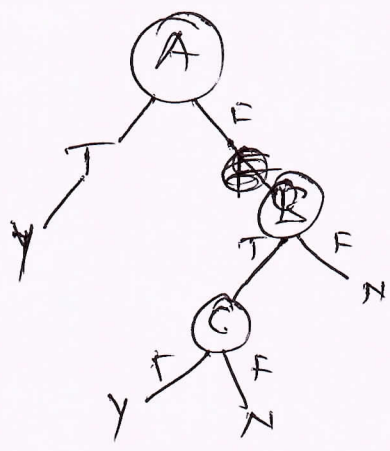
Logic	PROLOG
i) Variables are explicitly quantified	i) Quantification is provided implicitly by the way variable interpreted.
ii) Explicit symbols for AND ($\wedge$), OR($\vee$)	ii) Explicit symbol only for AND (,) but not for OR.
iii) P implies q is written as $P \rightarrow Q$	iii) P implies q is written as q :- P

Q5 (a) Give decision tree to represent the following Boolean functions: Module -5

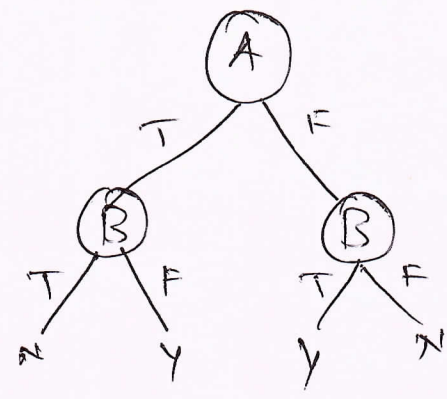
i) $A \wedge \neg B$



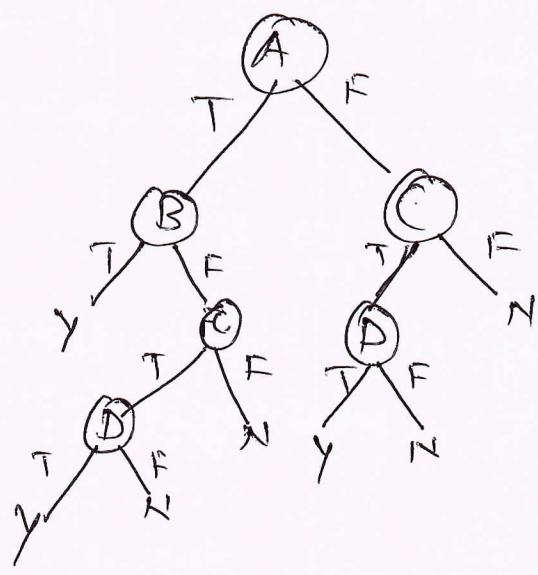
ii) $A \vee [B \wedge C]$



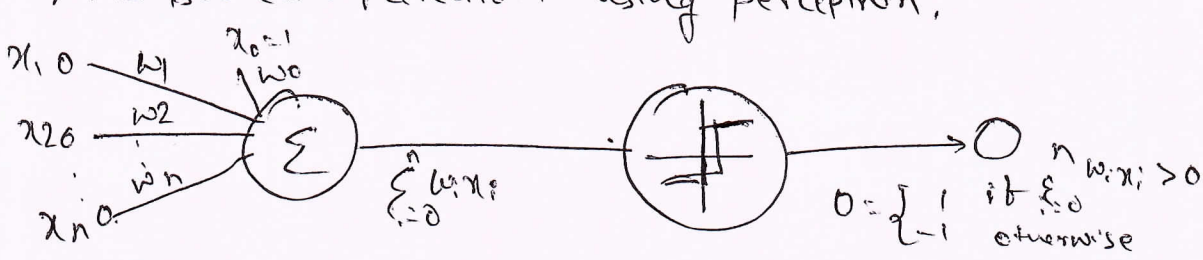
iii) A XOR B



iv) $[A \wedge B] \vee [C \wedge D]$

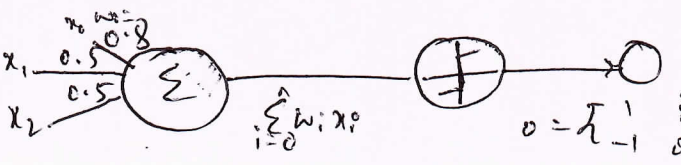


Q6 b) Explain Perceptron with appropriate diagram. Represent AND Boolean function using perceptron.



The perceptron outputs a 1 for instances lying on one side of hyperplane & outputs a -1 for instances lying on other side.

AND function:



6(c) Write Back propagation algorithm

Algorithm:

BACKPROPAGATION (training examples, η, n_{in}, n_{hid}, n_{out})

Each training example is a pair of the form $(\vec{x}, \vec{t})$ $x \Rightarrow$ vector of network input values & $\vec{t} \Rightarrow$ vector of target network output values.

$\eta \Rightarrow$ learning rate

$n_{in} \Rightarrow$ No. of network inputs

$n_{hidden} \Rightarrow$ No. of units in hidden layer.

$n_{out} \Rightarrow$ no. of output units.

The input from i to j is denoted as x_{ji} & weights from i to j is denoted as w_{ji}.

- Create a feed-forward network with n_{in} inputs, n_{hidden} hidden units & n_{out} output units.
- Initialize all network weights to small random numbers
- Until the termination condition is met, DO
  - For each $(\vec{x}, \vec{t})$ in training examples, DO

Propagate the input forward through the network

1. Input the instance $\vec{x}$ to the network and compute output O_u of every unit u in the network.

Propagate the errors backwards through the network

2. For each network output unit k, calculate error δ_k

$$\delta_k \leftarrow O_k(1-O_k)(t_k - O_k)$$

3. For each hidden unit h, calculate error term δ_h

$$\delta_h \leftarrow O_h(1-O_h) \sum_{k \in outputs} w_{kh}\delta_k$$

4. Update each network weight w_{ji}

$$w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$$

Module - 4

Q7 a) A patient takes a lab test and the result comes back Positive. The test returns a correct +ve result in only 98% of the cases in which the disease is actually present and a correct negative result in only 97% of the cases in which the disease is not present. Further, 0.8% of entire population have the cancer. Does a patient have cancer or not?

$P(\text{disease}) = 0.008$ $\quad P(\neg\text{disease}) = 0.992$

$P(+|\text{disease}) = 0.98$ $\quad P(-|\text{disease}) = 0.02$

$P(+|\neg\text{disease}) = 0.03$ $\quad P(-|\neg\text{disease}) = 0.97.$

$\therefore P(+|\text{disease})\,P(\text{disease}) = 0.98 \times 0.008 = 0.0078$

$P(+|\neg\text{disease})\,P(\neg\text{disease}) = (0.03) \times (0.992) = 0.298$

Thus $h_{MAP} = \neg\text{disease} \Rightarrow$ disease does not have cancer.

Q7 (b) Derive Brute force MAP learning & also mention assumption made in this process.

Assumptions

1. The training data D is noise free ($d_i = c(x_i)$)
2. The target concept C is contained in the hypothesis space H
3. We have no a priori reason to believe that any hypothesis is more probable than any other.

$$P(h) = \frac{1}{|H|} \quad \forall h \in H$$

$P(D|h)$ ⇒ Probability of observing target values $D = \langle d_1 \ldots d_m \rangle$ for instances $\langle x_1 \ldots x_m \rangle$

$$P(D|h) = \begin{cases} 1 & \text{if } d_i = h(x_i) \ \forall\ d_i \text{ in } D \\ 0 & \text{otherwise} \end{cases}$$

W.K.T $$P(h|D) = \frac{P(D|h)\,P(h)}{P(D)}$$

When h is inconsistent with D, $P(D|h) = 0$

$$\therefore P(h|D) = \frac{0 \cdot P(h)}{P(D)} = 0$$

When h is consistent with D, $P(D|h) = 1$

$$\therefore P(h|D) = \frac{1 \cdot \frac{1}{|H|}}{P(D)} = \frac{1 \cdot \frac{1}{|H|}}{\frac{|VS_{H,D}|}{|H|}} = \frac{1}{|VS_{H,D}|} \quad \text{if } h \text{ is consistent with } D$$

Where $VS_{H,D}$ is subset of hypotheses from H that are consistent with D.

$$P(D) = \sum_{h_i \in H} P(D|h_i)\,P(h_i) = \sum_{h_i \in VS_{H,D}} 1 \cdot \frac{1}{|H|} + \sum_{h_i \notin VS_{H,D}} 0 \cdot \frac{1}{|H|}$$

$$= \sum_{h_i \in VS_{H,D}} 1 \cdot \frac{1}{|H|} = \frac{|VS_{H,D}|}{|H|}$$

$$\therefore P(h|D) = \begin{cases} \frac{1}{|VS_{H,D}|} & h \text{ is consistent with } D \\ 0 & \text{otherwise} \end{cases}$$

Q 7 a) Explain Minimum Description length Principle (MDL).

Minimum Description length Principle

- It is Bayesian perspective on Occam's razor.
- Motivated by interpreting the definition of h_{MAP} in the light of basic concepts from Information theory.

$$h_{MAP} = \underset{h \in H}{\operatorname{argmax}}\ P(D|h)\,P(h)$$

Which can be expressed in terms of maximizing the $\log_2$

$$h_{MAP} = \underset{h \in H}{\operatorname{argmax}}\ \log_2 P(D|h) + \log_2 P(h)$$

or

alternatively, minimizing the negative of this quantity

$$h_{MAP} = \underset{h \in H}{\operatorname{argmin}}\ -\log_2 P(D|h) - \log_2 P(h) \quad \text{— (1)}$$

The equation (1) can be interpreted as statement that short hypotheses are preferred

- $-\log_2 P(h)$: the description length of h under the optimal encoding for the hypothesis space H, $L_{C_H}^{(h)} = -\log_2 P(h)$, C_H ⇒ optimal code for hypothesis space H.
- $-\log_2 P(D|h)$: The description length of the training data D given hypothesis h, under optimal encoding from space H: $L_{C_{D|h}}^{(D|h)} = -\log_2 P(D|h)$, $C_{D|h}$ is the optimal code for describing data D

$$\therefore h_{MAP} = \underset{h \in H}{\operatorname{argmin}}\ L_{C_H}(h) + L_{C_{(D|h)}}^{(D|h)}$$

Q 8 b) Explain Naive Bayes Classifier & Bayes Belief Network

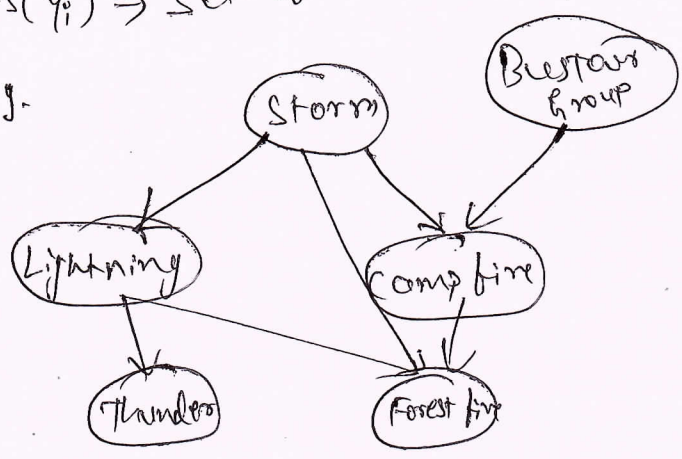
The naive Bayes classifier applies to learning tasks where each instance is described by conjunction of attribute values & where target function take any value from finite set V.

The joint probability for any desired assignment of values $(y_1, \ldots y_n)$ to the tuple of network variables $(Y_1, \ldots Y_n)$ is given by:

$$P(y_1, \ldots y_n) = \prod_{i=1}^{n} P(y_i | \text{Parents}(Y_i))$$

Parents(Y_i) ⇒ set of immediate predecessors of Y_i in the network

e.g.



	S,B	S,¬B	¬S,B	¬S,¬B
C	0.4	0.1	0.8	0.2
¬C	0.6	0.9	0.2	0.8

campfire

Q8 c) Write EM Algorithm

The EM Algorithm first initializes the hypothesis:

$$h = \langle \mu_1, \mu_2 \rangle$$

μ_1 & μ_2 ⇒ arbitrary initial values.

It then iteratively re-estimates h by repeating the following two steps until the procedure converges to a stationary value for h.

Step 1: Calculate the expected value $E[z_{ij}]$ of each hidden variable z_{ij}, assuming the current hypothesis $h = \langle \mu_1, \mu_2 \rangle$ holds.

Step 2: Calculate a new maximum likelihood hypothesis $h' = \langle \mu_1', \mu_2' \rangle$ assuming the value taken on by each hidden variable z_{ij} is its expected value $E[z_{ij}]$ calculated in step 1. Then replace the hypothesis $h = \langle \mu_1, \mu_2 \rangle$ by the new hypothesis $h' = \langle \mu_1', \mu_2' \rangle$ & iterate.

$$E[z_{ij}] = \frac{P(x = x_i | \mu = \mu_j)}{\sum_{n=1}^{2} P(x = x_i | \mu = \mu_n)} = \frac{e^{-\frac{1}{2\sigma^2}(x_i - \mu_j)^2}}{\sum_{n=1}^{2} e^{-\frac{1}{2\sigma^2}(x_i - \mu_n)^2}}$$

$$\mu_j \leftarrow \frac{1}{m} \sum_{i=1}^{m} E[z_{ij}] x_i$$

The Bayesian approach to classifying new instance is to assign most probable target value, V_{MAP} given attribute values $(a_1, a_2, \ldots a_n)$ that describe the instance

$$V_{MAP} = \underset{V_j \in V}{\operatorname{argmax}} P(V_j | a_1, a_2, \ldots a_n)$$

Using Bayes thm.

$$V_{MAP} = \underset{V_j \in V}{\operatorname{argmax}} \frac{P(a_1, a_2, a_3 \ldots a_n | V_j) P(V_j)}{P(a_1, a_2, \ldots a_n)}$$

$$= \underset{V_j \in V}{\operatorname{argmax}} P(a_1, a_2, \ldots a_n | V_j) P(V_j)$$

- The Naive Bayes classifier is based on assumption that the attribute values are conditionally independent given the target value.

i.e. $P(a_1, a_2, \ldots a_n | V_j) = \prod_i P(a_i | V_j)$

∴ Naive Bayes Classifier:

$$V_{NB} = \underset{V_j \in V}{\operatorname{argmax}} P(V_j) \prod_i P(a_i | V_j)$$

Bayesian Belief Networks

A Bayesian Belief network represents the joint probability distribution by specifying a set of conditional independence assumptions.

- Bayesian Networks are represented by directed acyclic graph, together with sets of local conditional probabilities.
- Each variable in the joint space is represented by a node in the Bayesian network.
- The network arcs represents the assertion that the variable is conditionally independent of its non-descendants in the network given its immediate predecessors in the network.
- A conditional probability table (CPT) is given for each variable, describing the probability distribution for that variable given the values of its immediate predecessors.

General Statement of EM Algorithm

Step 1: Estimation (E) step: Calculate $Q(h'|h)$ using current hypothesis h & the observed data X to estimate the probability distribution over Y.

$$Q(h'|h) \leftarrow E\left[\ln P(y|h')\, h, X\right]$$

Step 2: Maximization (M) step: Replace hypothesis h by the hypothesis h' that maximizes this Q function.

$$h \leftarrow \underset{h'}{\operatorname{argmax}}\, Q(h'|h)$$

When the function Q is continuous, the EM algorithm converges to a stationary point of the likelihood function $P(y|h')$.

Module 5

Q9 a) Explain K-NN algorithm

- The most basic instance-based method is K-NN learning.
- The nearest neighbors of an instance are defined in terms of the standard Euclidean distance.
- Let instance x be described by the feature vector

$$(a_1(x), a_2(x), \ldots a_n(x))$$

$a_r(x)$ denotes the value of the r^{th} attribute of instance X.

- Then distance between two instances x_i & x_j is defined to be $d(x_i, x_j)$ where,

$$d(x_i, x_j) \equiv \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2}$$

Discrete-valued target function - K NN

Training algorithm:

- For each training example $\langle x, f(x) \rangle$, add the example to the list training_examples.

Classification algorithm:

- Given a query instance x_q to be classified.
  - Let $x_1 \ldots x_k$ denote the k instances from training examples that are nearest to x_q
  - Return

$$\hat{f}(x_q) \leftarrow \underset{v \in V}{\operatorname{argmax}} \sum_{i=1}^{k} \delta(v, f(x_i))$$

where $\delta(a,b) = 1$ if $a = b$ & where $\delta(a,b) = 0$ otherwise

Real-valued target function

Training algorithm:

- For each training example $\langle x, f(x) \rangle$, add the example to the list training_examples.

Classification algorithm:

- Given a query instance x_q to be classified,
  - Let $x_1 \ldots x_k$ denote the k instances from training_examples that are nearest to x_q
  - Return

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} f(x_i)}{k}$$

Q9 b) Explain steps of Locally weighted Linear regression.

- Consider locally weighted regression in which the target function f is approximated near x_q using a linear function of form

$$\hat{f}(x) = w_0 + w_1 a_1(x) + \ldots + w_n a_n(x)$$

$a_i(x)$ denotes the value of the i^{th} attribute of the instance x.

Derive methods are used to choose weights that minimiz
the squared error summed over the set D of training exam
using gradient descent

$$E \equiv \frac{1}{2} \sum_{x \in D} \left(f(x) - \hat{f}(x) \right)^2$$

Which led us to the gradient descent training rule

$$\Delta w_j = \eta \sum_{x \in D} (f(x) - \hat{f}(x)) a_j(x)$$

$\eta \Rightarrow$ learning rate.

The simple way to redefine error criterion E to emphasize fitting local training examples. There are 3 possible criterias:

1) Minimize the squared error over just the k nearest nbrs:

$$E_1(x_q) \equiv \frac{1}{2} \sum_{x \in \text{k nearest nbrs of } x_q} (f(x) - \hat{f}(x))^2$$

2) Minimize the squared error over the entire set D of training examples, while weighting the error of each training example by some decreasing function K of its distance from x_q:

$$E_2(x_q) \equiv \frac{1}{2} \sum_{x \in D} (f(x) - \hat{f}(x))^2 K(d(x_q, x))$$

Combine (1) & (2)

$$E_3(x_q) \equiv \frac{1}{2} \sum_{x \in \text{k nearest nbrs of } x_q} (f(x) - \hat{f}(x))^2 K(d(x_q, x))$$

If we choose criterion 3 & re-derive the gradient descent rule, we obtain the following rule

$$\Delta w_j = \eta \sum_{x \in \text{k nearest nbrs of } x_q} K(d(x_q, x))(f(x) - \hat{f}(x)) a_j(x)$$

Q9 (c) Describe Radial basis functions with appropriate diagram

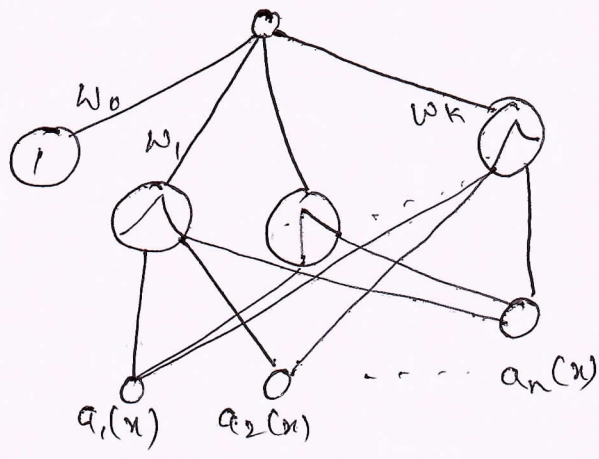
Given a set of training examples of the target function, RBF networks are typically trained in a two stage process.

1. The number 'k' of hidden units is determined & each hidden unit 'u' is defined by choosing the values of x_u that define its kernel function $K_u(d(x_u, x))$

2. The weights w_i are trained to maximize the fit of the network to the training data, using the global error criterion given by

$$E \equiv \frac{1}{2} \sum_{x \in D} (f(x) - \hat{f}(x))^2$$

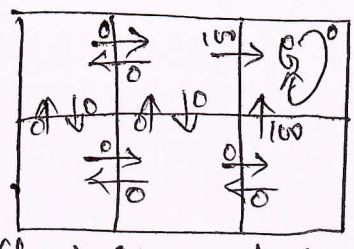
Because the kernel functions are held fixed during this stage the linear weight values w_i can be trained very efficiently



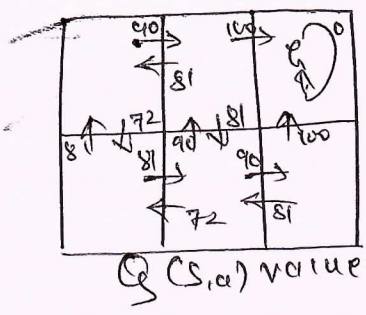
Several alternative methods have been proposed for choosing an appropriate no. of hidden units or, equivalently kernel functions.

- One approach is to allocate a Gaussian kernel function for each training example $(x_i, f(x_i))$ centring this Gaussian at point x_i. Each of these kernels may be assigned the same width σ^2. Given this approach, RBF network learns a global approximation to the target function in which each training example $(x_i, f(x_i))$ can influence the value of $\hat{f}$ only in the neighborhood of x_i.
- A second approach is to choose a set of kernel functions that is smaller than the number of training examples. This approach can be much more efficient than the first approach, especially when the no. of training examples is large.

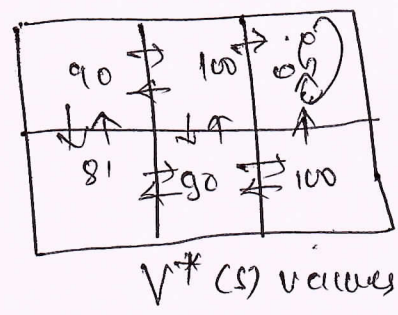
Q 10 (a) Illustrate the basic concept of Q-learning using Simple Deterministic World.



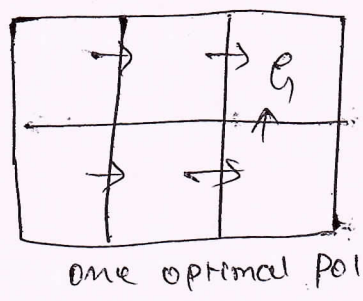
r(S,a) (immediate reward) values



Q(S,a) values



$V^*(S)$ values



one optimal policy.

The six grid squares represent 6 possible states, or locations, for the agent. Each arrow in the diagram represents a possible action the agent can take to move from one state to another.

The number associated with each arrow represents the immediate reward r(S,a) the agent receives if it executes the corresponding state-action transition.

G is the goal-state, as the only way the agent can receive reward, by entering this state. G is absorbing state.

Once the states, actions & immediate rewards are defined & once we choose a value for the discount factor γ, we can determine optimal policy π^* & its value function $V^*(S)$. Let us choose $\gamma = 0.9$

- the bottom figure shows optimal policy.

the diagram at the right of above figure shows value of V^* for each state. Consider bottom right side diagram. The value of V^* for this state is 100 because the optimal policy in this state selects the "move up" action that receives immediate reward of 100. Thereafter, the agent will remain in the absorbing state & receive no further rewards. Similarly, the value of V^* for the bottom center state is 90.

V^* is defined to be the sum of discounted future rewards over the infinite future. In this particular environment, once the agent reaches the absorbing state G its infinite future will consist of remaining in this state & receiving rewards of zero.

Q 10 b) Explain Q-Learning algorithm.

- Learning the Q function corresponds to learning the optimal policy.
- The key problem is finding a reliable way to estimate training values for Q.

$$V^*(S) = \max_{a'} Q(S,a')$$

$$Q(S,a) = r(S,a) + \gamma \max_{a'} Q(\delta(S,a),a')$$

Q learning algorithm

For each S,a initialize the table entry $\hat{Q}(S,a)$ to zero
Observe the current state S
Do Forever:
- Select an action a & execute it
- Receive immediate reward r
- Observe the new state S'
- Update the table entry for $\hat{Q}(S,a)$ as follows:

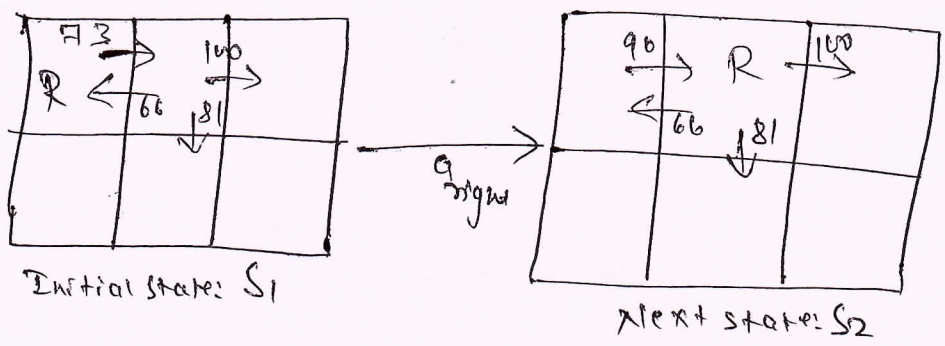
$$\hat{Q}(S,a) \leftarrow r + \gamma \max_{a'} \hat{Q}(S',a')$$

- $S \leftarrow S'$

- Q learning algorithm assumes deterministic rewards & actions. The discount factor γ may be any constant such that $0 \leq \gamma < 1$
- $\hat{Q}$ refer to the learner's estimate, or hypothesis, of the actual Q function.

Example

Consider a single action taken by an agent & the corresponding refinement to $\hat{Q}$ as shown below



- The agent moves one cell to the right in its grid world and receives an immediate reward of zero for this transition.
- Apply training rule

$$\hat{Q}(s,a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s',a')$$ to refine its estimate Q for the state action transition it just executed.

- According to the training rule, the new $\hat{Q}$ estimate for this transition is the sum of the received reward (zero) & the highest $\hat{Q}$ value associated with the resulting state (100), discounted by γ(0.9).

$$\hat{Q}(S_1, a_{right}) \leftarrow r + \gamma \max_{a'} \hat{Q}(S_2, a')$$
$$\leftarrow 0 + 0.9 \max[66, 81, 100]$$
$$\leftarrow 90$$

Q-Learning Algorithm converge toward a Q equal to true function under certain conditions:-

i) Assume the system is a deterministic MDP.

ii) Assume the immediate reward values are bounded

iii) Assume the agent selects actions in such a fashion that it visits every possible state action pair infinitely

Pranesh.K

HOD

HOD
Computer Science & Engineering
KLS Vishwanathrao Deshpande
Institute of Technology, Hâliyal.

Dean Academics