CMRIT
CMR INSTITUTE OF TECHNOLOGY, BENGALURU.
ACCREDITED WITH A+ GRADE BY NAAC

Internal Assessment Test 1 Scheme and Solutions– October 2022

Sub:	Artificial Intelligence and Machine Learning – Set 2			Sub Code:	18CS71	Branch:	ISE	
Date:	20-10-2022	Duration:	90 Minutes	Max Marks: 50		Sem / Sec:	7 A,B,C	OBE

Answer any FIVE FULL Questions		Marks Distribution	Max Marks
1	• Candidate elimination algorithm • Inductive bias explanation	5 M 5 M	10 M
2	• Find-S algorithm • Problem: Initializing specific hypothesis and generic hypothesis • Finding the maximally specific hypothesis	4 M 1 M 5 M	10 M
3	• Initializing specific hypothesis and generic hypothesis • For each positive and negative instances: Finding the maximally specific and generic hypothesis	2 M 8 M	10 M
4	• Defining Consistent Hypothesis • Defining Version Space • Defining List-then-Eliminate Algorithm	2 M 2 M 6 M	10 M
5	• Calculating the overall entropy • Calculating individual attribute gain values along with entropy • Identifying root node and sub nodes • Constructing final decision tree	2 M 4 M 2 M 2 M	10 M
6	• Defining perceptron • Diagram • Discussing the Training rule	2 M 3 M 5 M	10 M

SN	Question	Marks	CO	BT
1	Write the algorithm for Candidate Elimination and explain the Inductive bias.	10	CO1	L1

Solution:

Candidate Elimination Algorithm:

1. Initialize G to the set of maximally general hypotheses in H
2. Initialize S to the set of maximally specific hypotheses in H
3. For each training example d, do
   a. If d is a positive example
      - Remove from G any hypothesis inconsistent with d,
      - For each hypothesis s in S that is not consistent with d,
        - Remove s from S
        - Add to S all minimal generalizations h of s such that h is consistent with d, and some member of G is more general than h
        - Remove from S, hypothesis that is more general than another hypothesis in S

   b. If d is a negative example
      - Remove from S any hypothesis inconsistent with d
      - For each hypothesis g in G that is not consistent with d
        - Remove g from G
        - Add to G all minimal specializations h of g such that h is consistent with d, and some member of S is more specific than h
        - Remove from G any hypothesis that is less general than another in G

Inductive Bias:

1. A Biased Hypothesis Space

- Suppose we wish to assure that the hypothesis space contains the unknown target concept.
- The obvious solution is to enrich the hypothesis space to include every possible hypothesis.
- Consider EnjoySport example in which we restricted the hypothesis space to include only conjunctions of attribute values.

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Cool	Change	Yes
2	Cloudy	Warm	Normal	Strong	Cool	Change	Yes
3	Rainy	Warm	Normal	Strong	Cool	Change	No

- Most specific hypothesis consistent with the first two examples
- It incorrectly covers the third (negative) training example
- The problem is that we have biased the learner to consider only conjunctive hypotheses.
- In this case we require a more expressive hypothesis space.

2. An unbiased learner

- The obvious solution to be a unbiased learner– design hypothesis space H to represent ***every teachable concept***;
- It should capable of representing every possible subset of the instances X. In general, the set of all subsets of a set X is called the ***power-set*** of X.
- In general, number of distinct subsets is $2^{|X|}$.

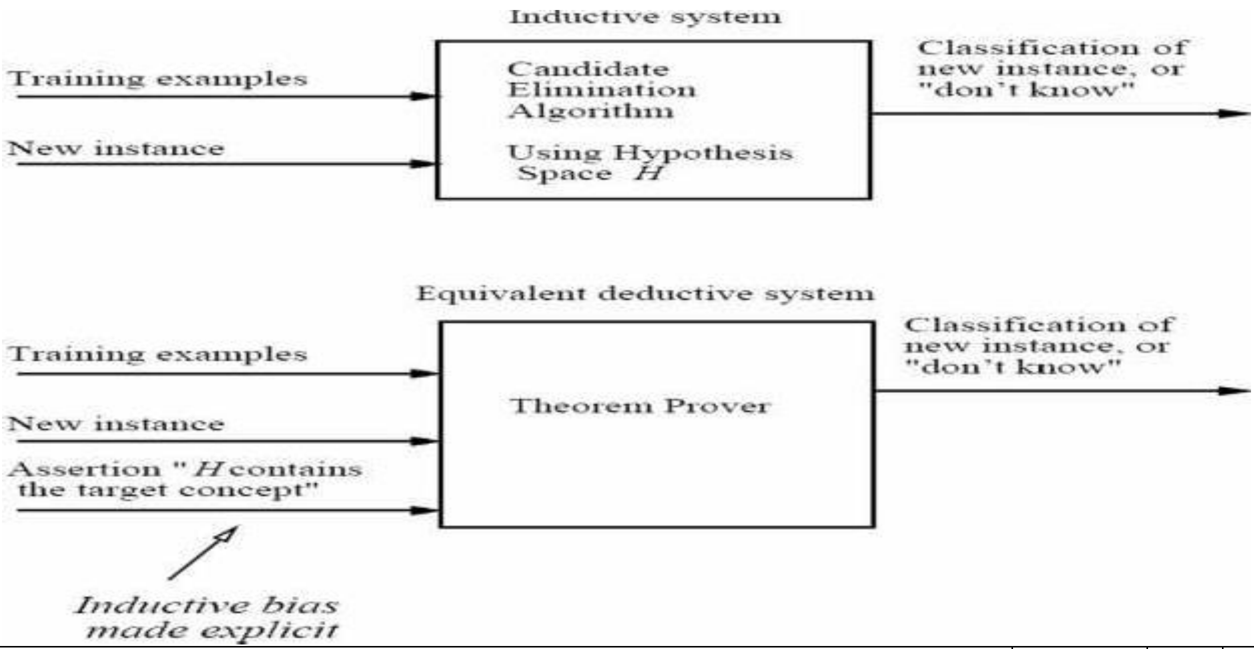
- Thus, there are 2^{96}, or approximately distinct target concepts that could be defined over this instance space and that our learner might be called upon to learn.

Our conjunctive hypothesis space is able to represent only 973 of these-a very biased hypothesis space indeed!

- Let us reformulate the *Enjoysport* learning task
- Let H’ represent every subset of instances; that is, let H’ correspond to the power set of X.
- One way to define such an H' is to allow arbitrary disjunctions, conjunctions, and negations of our earlier hypotheses.
- For instance, the target concept "Sky = Sunny or Sky = Cloudy" could then be described as $\langle Sunny, ?, ?, ?, ?, ?\rangle \vee \langle Cloudy, ?, ?, ?, ?, ?\rangle$

3. The Futility of Bias-Free Learning

- CEA generalizes observed training examples because it was biased by the implicit assumption that the target concept could be represented by a conjunction of attribute values.
  - If this assumption is correct (and the training examples are error-free), its classification of new sample will also be correct.
  - If this assumption is incorrect, however, it is certain that the CEA will mis-classify at least some instances from X.



2	Describe the steps in FIND-S algorithm and apply the algorithm for the dataset given below.	10	CO2	L3

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

Solution:

FIND-S Algorithm

1.Initialize *h* to the most specific hypothesis in *H*

1. For each positive training instance *x*

- For each attribute constraint a_i in *h*
  - If the constraint a_i is satisfied by *x*
    Then do nothing
  - Else replace a_i in *h* by the next more general constraint that is satisfied by *x*

2. Output hypothesis *h*

Problem Solution

The first step of FIND-S is to initialize ***h*** to the most specific hypothesis

h - (Ø, Ø, Ø, Ø, Ø, Ø)

1st instance x_1 = *<Sunny Warm Normal Strong Warm Same>, +*

Observing the first training example, it is clear that our hypothesis is too specific. In particular, none of the "Ø" constraints in h are satisfied by this example, so each is replaced by the next more general constraint that fits the example

h_1 = <Sunny Warm Normal Strong Warm Same>

2nd instance x_2 = <Sunny, Warm, High, Strong, Warm, Same>, +

The second training example forces the algorithm to further generalize h, this time substituting a "?' in place of any attribute value in h that is not satisfied by the new example

h_2 = <Sunny Warm ? Strong Warm Same>

3rd instance x3 = <Rainy, Cold, High, Strong, Warm, Change>, -

Upon encountering the third training the algorithm makes no change to h. The FIND-S algorithm simply ignores every negative example.

h3 = < Sunny Warm ? Strong Warm Same>

4th instance x4 = <Sunny,Warm, High, Strong, Cool, Change>, +

The fourth example leads to a further generalization of h

h4 = < Sunny Warm ? Strong ? ? >

3	Apply the Candidate Elimination algorithm for the dataset given below.	10	CO2	L3

Origin	Manufacturer	Color	Decade	Type	Example Type
Japan	Honda	Blue	1980	Economy	Positive
Japan	Toyota	Green	1970	Sports	Negative
Japan	Toyota	Blue	1990	Economy	Positive
USA	Chrysler	Red	1980	Economy	Negative
Japan	Honda	White	1980	Economy	Positive
Japan	Toyota	Green	1980	Economy	Positive
Japan	Honda	Red	1990	Economy	Negative

Solution:

Candidate Elimination Algorithm:

S0= 0,0,0,0,0
G0=?,?,?,?,?

1st instance: S1= Japan, Honda, Blue,1980,Economy +ve
G1=?,?,?,?,?

2nd instance: Japan, Toyota, Green, 1970,Sports -ve
S2= Japan, Honda, Blue,1980,Economy
Try to make each ? with different possible pairs
G2= (USA,?,?,?,?)(?, Honda,?,?,?)(?, Chrysler,?,?,?)(?,?,Blue,?,?)(?,?, Red,?,?) (?,?,White,?,?) (?,?,?,1980,?) (?,?,?,1990,?)(?,?,?,?,Economy)

2nd, 4th, 7th and 9th pairs are consistant with S2 and rest all are inconsistent. Hence we eliminate them.
Final G2= (?,Honda,?,?,?) (?,?,Blue,?,?)(?,?,?, 1980,?)(?,?,?,?,Economy)

3rd instance: Japan,Toyota,Blue,1990,Economy +ve
S3=Japan,?,Blue,?,Economy
Make G3 more consistant pairs by removing less consistant pairs with Specific hypothesis.
G3 ={?,?,blue,?,?}{?,?,?,?,Economy} because first and 3rd pair is not consistant with specific hypothesis.

4th instance: USA,Chrysler,Red,1980,Economy -ve
S4= Japan,?, Blue,?,Economy

G4= (Japan,?,Blue,?,?)(?,Honda,Blue,?,?)(?,Toyota,Blue,?,?)(?,?,Blue,1970,?)(?,?,Blue,1990,?)(?,?,Blue,?,sports)(Japan,?,?,?,Economy)(?,Honda,?,?,Economy)(?,Toyota,?,?,Economy)(?,?,Blue,?,Economy)(?,?,Green,?,Economy)(?,?,White,?,Economy)(?,?,?,1970,Economy)(?,?,?,1980,Economy)
G4= (Japan,?,Blue,?,?)(Japan,?,?,?,Economy)(?,?,Blue,?,Economy)

5th instance: Japan,Honda,White,1980,Economy +ve
S5= Japan,?,?,?,Economy
G5= (Japan,?,?,?,Economy) bcz 1st and 3rd pairs are not consistant with 5th instance

6th instance: Japan,Toyota,Green,1980,Economy +ve
S6= Japan,?,?,?, Economy
G6=Japan,?,?,?,Economy

7th instance: Japan,Honda,Red,1990,Economy -ve
Whatever G6 and S6 we have got, it is consistant with negative instance. So S cannot be generalized and G cannot be specialized further.

4	Define Consistent Hypothesis and Version Space. Write a LIST-THEN-ELIMINATE algorithm.	10	CO1	L2

Solution:

Consistent Hypothesis: A hypothesis h is consistent with a set of training examples D if and only if h(x) = c(x) for each example (x, c(x)) in D.
Consistent (h, D) = (for all <x, c(x)> € D) h(x) = c(x))

Note difference between definitions of consistent and satisfies
• an example x is said to satisfy hypothesis h when h(x) = 1, regardless of whether x is a positive or negative example of the target concept.
• an example x is said to consistent with hypothesis h iff h(x) = c(x)

Version Space: The version space, denoted VSH,D with respect to hypothesis space H and training examples D, is the subset of hypotheses from H consistent with the training examples in D
VSH,D ={h € H | Consistent(h, D)}

LIST-THEN-ELIMINATE Algorithm:
The LIST-THEN-ELIMINATE algorithm first initializes the version space to contain all hypotheses in H and then eliminates any hypothesis found inconsistent with any training example.

1. **VersionSpace** c a list containing every hypothesis in H
2. For each training example, (x, c(x))
    remove from **VersionSpace** any hypothesis h for which h(x) ≠ c(x)
3. Output the list of hypotheses in **VersionSpace**

List-Then-Eliminate works in principle, so long as version space is finite.
However, since it requires exhaustive enumeration of all hypotheses in practice it is not feasible.

5	Create and explain the decision tree for the following transactions using ID3 algorithm.	10	CO2	L3

Day	A1	A2	A3	Classification
1	True	Hot	High	No
2	True	Hot	High	No
3	False	Hot	High	Yes
4	False	Cool	Normal	Yes
5	False	Cool	Normal	Yes
6	True	Cool	High	No
7	True	Hot	High	No
8	True	Hot	Normal	Yes
9	False	Cool	Normal	Yes
10	False	Cool	High	No

Solution:

Entropy(S) = 1 bcz all the samples are equal
Gain(S, A1) = Entropy(S) – {(5/10)*Entropy (Strue) + (5/10)* Entropy (Sfalse)}
Entropy (Strue) = -(1/5)log2(1/5) – (4/5)log2(4/5) = 0.72
Entropy (Sfalse) = -(4/5)log2(4/5)-(1/5)log2(1/5) = 0.72
Gain(S, A1) = 1 – {(5/10)*0.72 + (5/10)*0.72} = 0.29

Gain(S, A2) = Entropy(S) –{(5/10)*Entropy(Shot) +(5/10)*Entropy(SCool)}
Entropy (Shot)= -(2/5)log(2/5) – (3/5)log(3/5) = 0.966
Entropy (Scool) = -(3/5)log(3/5)-(2/5)log(2/5) = 0.966
Gain(S, A2) = 1- {(5/10)*0.966 + (5/10)*0.966} = 0.03

Gain(S, A3) = Entropy(S) –{(6/10)*Entropy (Shigh) + (4/10*Entropy (Snormal)}
Entropy(Shigh) = -(1/6)log(1/6) –(5/6)log(5/6) = 0.647
Entropy (Snormal) = 0 bcz all are belonging to same class
Gain(S, A3) = 1 – ((6/10)* 0.647 + (4/10)*0} = 0.61

Gain(S, A1) = 0.29
Gain(S, A2) =0.03
Gain(S, A3) = 0.61, so A3 will be chosen as root node

A3
high Normal

1,2,3,6,7,10 Yes

Now considering A3 as the root node, next level root identification to be done whether it can be A1 or A2
Entropy(A3->High) = -(1/6)log(1/6)-(5/6)log(5/6) = 0.647
Gain(A3=High -> A1) = E(A3->High) – { (4/6) E(T) + (2/6) E(F)}
= 0.65 – {4/6 * 0 + (2/6) * 1} = 0.31
Gain(A3=High -> A2) = E(A3->High) – { (4/6) E(Hot) + (2/6) E(cool)}
E(hot) = -(1/4)log(1/4) – (3/4)log(3/4) = 0.81
E(Cool) = 0

A3
high Normal

1,2,3,6,7,10 Yes
A1
True False

No Yes

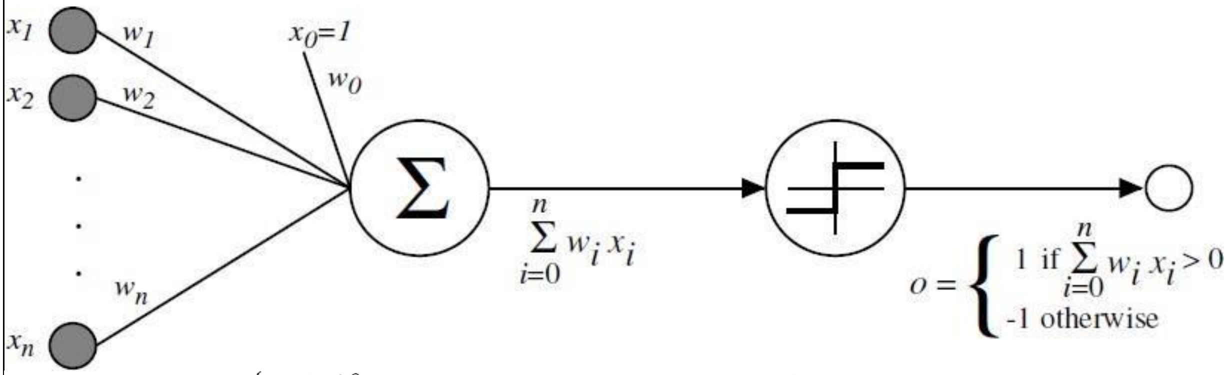
6	Explain the concept of a Perceptron with a neat diagram. Explain the Perceptron training rule.	10	CO1	L2

Solution:

One type of ANN system is based on a unit called a perceptron. Perceptron is a single layer neural network.

A perceptron takes a vector of real-valued inputs, calculates a linear combination of these inputs, then outputs a 1 if the result is greater than some threshold and -1 otherwise.

Given inputs x, the output O(x1, x2,….., xn) computed through perceptron is



$$o(x_1, \ldots, x_n) = \begin{cases} 1 & \text{if } w_0 + w_1 x_1 + \cdots + w_n x_n > 0 \\ -1 & \text{otherwise.} \end{cases}$$

The Perceptron Training Rule

The learning problem is to determine a weight vector that causes the perceptron to produce the correct + 1 or - 1 output for each of the given training examples.

To learn an acceptable weight vector

- Begin with random weights, then iteratively apply the perceptron to each training example, modifying the perceptron weights whenever it misclassifies an example.
- This process is repeated, iterating through the training examples as many times as needed until the perceptron classifies all training examples correctly.
- Weights are modified at each step according to the perceptron training rule, which revises the weight w_i associated with input xi according to the rule.
- The role of the learning rate is to moderate the degree to which weights are changed at each step. It is usually set to some small value (e.g., 0.1) and is sometimes made to decay as the number of weight-tuning iterations increases

$$w_i \leftarrow w_i + \Delta w_i$$

Where,

$$\Delta w_i = \eta(t - o)x_i$$

Here,

t is the target output for the current training example

o is the output generated by the perceptron

η is a positive constant called the ***learning rate***