USN

Internal Assessment Test 1 – June 2024

Sub:	**Machine Learning**			Sub Code:	**21AI63**	Branch:	**AInDS**	
Date:		Duration:	**90 minutes**	Max Marks:	**50**	Sem	**VI**	**OBE**

<table>
<tr><th colspan="3"><u>Answer any FIVE Questions</u></th><th>MARKS</th><th>CO</th><th>RBT</th></tr>
<tr><td>1</td><td>a</td><td>Solve by using Find-S Algorithm
<table>
<tr><th>Sr. No</th><th>CGPA</th><th>Interactive</th><th>Practical Knowledge</th><th>Communication Skill</th><th>Logical Thinking</th><th>Internship Done</th><th>Job Offer</th></tr>
<tr><td>1</td><td>>=9</td><td>Y</td><td>Excellent</td><td>Good</td><td>Fast</td><td>Y</td><td>Y</td></tr>
<tr><td>2</td><td>>=9</td><td>Y</td><td>Good</td><td>Good</td><td>Slow</td><td>Y</td><td>Y</td></tr>
<tr><td>3</td><td>>=8</td><td>N</td><td>Good</td><td>Good</td><td>Slow</td><td>N</td><td>N</td></tr>
<tr><td>4</td><td>>=9</td><td>Y</td><td>Good</td><td>Good</td><td>Slow</td><td>N</td><td>Y</td></tr>
</table></td><td>[5]</td><td>CO1</td><td>L2</td></tr>
<tr><td></td><td>b</td><td>List and Explain Perspectives and Issues in Machine Learning</td><td>[5]</td><td>CO1</td><td>L1</td></tr>
<tr><td>2</td><td></td><td>How you will design a Learning System? Explain with example</td><td>[10]</td><td>CO1</td><td>L2</td></tr>
<tr><td>3</td><td></td><td>Apply ID3 algorithm for constructing Decision Tree for the following training example
<table>
<tr><th>Outlook</th><th>Temperature</th><th>Humidity</th><th>Wind</th><th>PlayTennis</th></tr>
<tr><td>sunny</td><td>hot</td><td>high</td><td>weak</td><td>no</td></tr>
<tr><td>sunny</td><td>hot</td><td>high</td><td>strong</td><td>no</td></tr>
<tr><td>overcast</td><td>hot</td><td>high</td><td>weak</td><td>yes</td></tr>
<tr><td>rain</td><td>mild</td><td>high</td><td>weak</td><td>yes</td></tr>
<tr><td>rain</td><td>cool</td><td>normal</td><td>weak</td><td>yes</td></tr>
<tr><td>rain</td><td>cool</td><td>normal</td><td>strong</td><td>no</td></tr>
<tr><td>overcast</td><td>cool</td><td>normal</td><td>strong</td><td>yes</td></tr>
<tr><td>sunny</td><td>mild</td><td>high</td><td>weak</td><td>no</td></tr>
<tr><td>sunny</td><td>cool</td><td>normal</td><td>weak</td><td>yes</td></tr>
<tr><td>rain</td><td>mild</td><td>normal</td><td>weak</td><td>yes</td></tr>
<tr><td>sunny</td><td>mild</td><td>normal</td><td>strong</td><td>yes</td></tr>
<tr><td>overcast</td><td>mild</td><td>high</td><td>strong</td><td>yes</td></tr>
<tr><td>overcast</td><td>hot</td><td>normal</td><td>weak</td><td>yes</td></tr>
<tr><td>rain</td><td>mild</td><td>high</td><td>strong</td><td>no</td></tr>
</table></td><td>[10]</td><td>CO4</td><td>L3</td></tr>
<tr><td>4</td><td>a</td><td>Explain different types of Machine Learning and Main Challenges of Machine Learning</td><td>[5]</td><td>CO1</td><td>L1</td></tr>
<tr><td></td><td>b</td><td>Explain the concept of Entropy and Information Gain</td><td>[5]</td><td>CO4</td><td>L1</td></tr>
<tr><td>5</td><td></td><td>Define Following Terms:
a) Concept Learning
b) Version Space
c) Consistent Hypothesis
d) General Boundary
e) Specific Boundary</td><td>[10]</td><td>CO1</td><td>L1</td></tr>
<tr><td>6</td><td></td><td>Explain Candidate Elimination Algorithm with Example</td><td>[10]</td><td>CO1</td><td>L2</td></tr>
</table>

CI **CCI** **HOD**

Q 1 a) Solve by using Find-S Algorithm

Sr. No	CGPA	Interactive	Practical Knowledge	Communication Skill	Logical Thinking	Internship Done	Job Offer
1	>=9	Y	Excellent	Good	Fast	Y	Y
2	>=9	Y	Good	Good	Slow	Y	Y
3	>=8	N	Good	Good	Slow	N	N
4	>=9	Y	Good	Good	Slow	N	Y

Solution:

Page No.:

Step 1 Initialize 'h' to most specific hypothesis. There are 6 attributes so for each attribute we initially fill ϕ

$h = < \phi \ \phi \ \phi \ \phi \ \phi \ \phi >$

Step 2 Generalize initial hypothesis for first +ve instance. I1 is +ve instance so generalize most specific hypothesis.

I1 - ≥ 9 yes Ex Good fast yes

$h = < \geq 9$ Y Ex G F Y $>$

Step 3. Scan the next instance I_2. It is +ve so include it & math each attribute if nomatch then put '?'

I2 - ≥ 9 Y G G F Y

$h = < \geq 9$ Y ? G F Y $>$

Scan I3 It is negative so exclude it & hypothesis remains same without any change

I4 - ≥ 9 Y G G slow N

$h = < \geq 9$ Y ? G ? ? $>$

So this is final hypothesis generated by Find-S algorithm

Q 1 b) List and Explain Perspectives and Issues in Machine Learning [5] **Perspective 2M and Issues 3M**

Solution:

- One useful perspective on machine learning is that it involves searching a very large space of possible hypotheses to determine one that best fits the observed data and any prior knowledge held by the learner.
- The learner's task is thus to search through this vast space to locate the hypothesis that is most consistent with the available training examples.
- **Issues in Machine Learning**
- What algorithms exist for learning general target functions from specific training examples?
- In what settings will particular algorithms converge to the desired function, given sufficient training data?
- Which algorithms perform best for which types of problems and representations?
- How much training data is sufficient?
- What general bounds can be found to relate the confidence in learned hypotheses to the amount of training experience and the character of the learner's hypothesis space?
- When and how can prior knowledge held by the learner guide the process of generalizing from examples?
- Can prior knowledge be helpful even when it is only approximately correct?
- What is the best strategy for choosing a useful next training experience, and
- how does the choice of this strategy alter the complexity of the learning problem?

Q 2) How you will design a Learning System? Explain with example [10] 2M each Step

Steps are as follows:

Choosing the Training Experience

Choosing the Target Function

Choosing a Representation for the Target Function

Choosing a Function Approximation Algorithm

The Final Design

Choosing the Training Experience

- The first design choice we face is to choose the type of training experience from which our system will learn.
- The type of training experience available can have a significant impact on success or failure of the learner
- One key attribute is whether the training experience provides direct or indirect feedback regarding the choices made by the performance system.
- For example, in learning to play checkers, the system might learn from ***direct*** training examples consisting of individual checkers board states and the correct move for each.
- Alternatively, it might have available only ***indirect*** information consisting of the move sequences and final outcomes of various games played.
- A second important attribute of the training experience is the degree to which the learner controls the sequence of training examples.
- For example, the learner might rely on the teacher to select informative board states and to provide the correct move for each.

Choosing the Target Function

Step 2 — Choose Target Fu

Target function → $V(b)$

Board state → b

legal moves set → B

b → $V(b) = 100$ win state

b → $V(b) = -100$ loss state

$V(b) = 0$ draw state

if b is not final state in the game then $V(b) = V(b')$ where b' is the best final board state that can be achieved starting from b and playing optimally.

Choosing a Representation for the Target Function

Step 3 — choosing Representation for Target function

x_1 = no. of black pieces on board

x_2 = –"– Red –"–

x_3 = no. of black kings on board

x_4 = –"– Red –"–

x_5 = no. of black pieces threatened by red

x_6 = –"– Red –"– black

$$\hat{V}(b) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_6x_6$$

where w_1 to w_6 are coefficients or weights chosen by learning algorithm

w_0 = additive constant to board value.

Choosing a Function Approximation Algorithm

Step 4 choosing function Approximation Algorithm

1. Estimating training values
2. Adjusting the weights

In this step we need a set of training examples.

It is a ordered pair $(b, V(b))$

eg. following training example describes a board state 'b' in which black has won the game. for which target fu^n value $V_{train}(b) = +100$

It is associated with values assigned to intermediate states.

* Adjusting the weights

a) we need (require) an algo. that will incrementally refine the weights as new training example become available

b) such algo is called LMS training rule

LMS: Least mean square

Rule is as follows

$$w_i \leftarrow w_i + \eta \, (V_{train}(b) - \hat{V}(b)) \, x_i$$

For each training example $(b, V_{train}(b))$ use current weights to calculate $V_{bar}(b)$

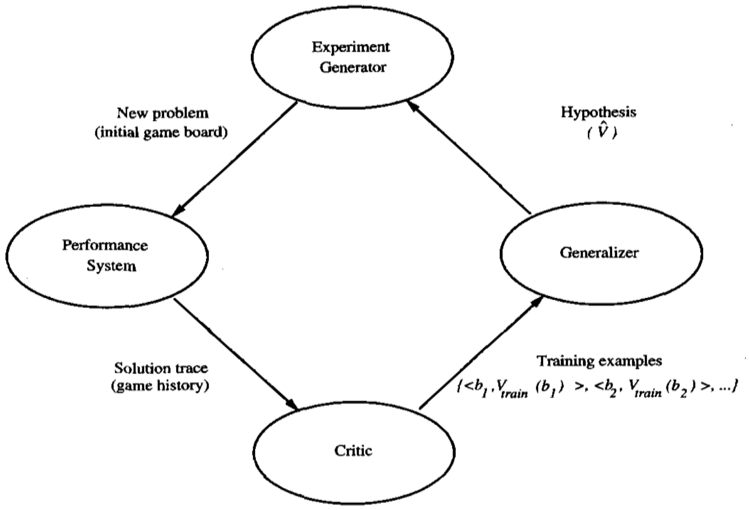
η is small constant

i) When $V_{train}(b) - V_{bar}(b) = 0$ then no weight change.

ii) if $V_{train}(b) - V_{bar}(b) = +ve$ then each weight is increased.

iii) If any feature x_i is zero then also no weight change

The Final Design



- Performance System is the module that must solve the given performance task, in this case playing checkers, by using the learned target function(s).
- It takes an instance of a new problem (new game) as input and produces a trace of its solution (game history) as output.
- **The Critic** takes as input the history or trace of the game and produces as output a set of training examples of the target function
- The **Generalizer** takes as input the training examples and produces an output hypothesis that is its estimate of the target function.
- **Experiment Generator** takes as input the current hypothesis (currently learned function) and outputs a new problem (i.e., initial board state) for the Performance System to explore.
- Its role is to pick new practice problems that will maximize the learning rate of the overall system.

Q 3) Apply ID3 algorithm for constructing Decision Tree for the following training example

Outlook	Temperature	Humidity	Wind	PlayTennis
sunny	hot	high	weak	no
sunny	hot	high	strong	no
overcast	hot	high	weak	yes
rain	mild	high	weak	yes
rain	cool	normal	weak	yes
rain	cool	normal	strong	no
overcast	cool	normal	strong	yes
sunny	mild	high	weak	no
sunny	cool	normal	weak	yes
rain	mild	normal	weak	yes
sunny	mild	normal	strong	yes
overcast	mild	high	strong	yes
overcast	hot	normal	weak	yes
rain	mild	high	strong	no

Solution:

Basic algo extension - C4.5

Page No.:

Decision Tree ID3 Algorithm

Here first we have to find max^m information gain (which attribute gives max^m information), consider that record / attribute as a root node & then build the tree

values (outlook) = Sunny, Overcast, Rain

But first we have to calculate Entropy of whole dataset

It is denoted as 'S' & formula is

$S = [9+, 5-]$ total = 14

9+ = 9 responses are +ve

5- = 5 " -ve

$$S = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} =$$

0.3676

0.2840

-0.6428 ×

-0.191

-0.3571 ×

-0.4971

-0.1227

$\boxed{S = 0.94}$

Now calculate Entropy of sunny, overcast & Rain

S_{sunny} ← total 5 [2+, 3-]

$$Entro(S_{sunny}) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}$$

$= 0.971$

$S_{overcast}$ ← total 4 [4+ 0-]

$$S_{over} = -\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}$$

$= 0$

S_{rain} ← total = 5 [3+, 2-]

$$Ent(S_{rain}) = -\frac{3}{5}\log_2\frac{3}{4} - \frac{2}{5}\log_2\frac{1}{4}$$

$= 0.971$

$$Gain(S, outlook) = Entropy(S) - \sum \left|\frac{S_v}{S}\right|$$

$$= Entropy\ S - \frac{5}{14} \times Entro\ S_{sunny} - \frac{4}{14} Entro$$

$$- \frac{5}{14} Entro(S_{rain})$$

$$= 0.94 - \frac{5}{14} \times 0.971 - \frac{4}{14} \times 0 - \frac{5}{14} \times 0.971$$

$= 0.2464$

In similar way calculate Entropy Inf^m Gain for attribute Temp, Humidity, wind

Temp = S = 0.94

$S_{Hot} = 1.0$ - if +ve & -ve are equal then value = 1.0

$S_{mild} = 0.9183$

$S_{cool} = 0.8113$

Page No.:

$\boxed{Gain(Temp) = 0.0289}$

Humidity =

$S_{High} = 0.9852$

$S_{normal} = 0.5916$

Gain (Humidity) = 0.1516

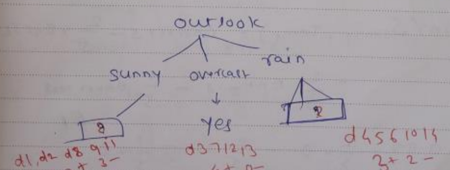
Attribute wind

S_{strong} - 1

S_{weak} - 0.8113

$\boxed{Gain (wind) = 0.0478}$

Now compare all 4 attributes gain max^m gain by outlook = 0.2464



New dataset becomes excluding overlook

day	Temp	Humidity	wind	play Tennis
1	Hot	High	weak	No
2	Hot	High	Strong	No
8	mild	High	weak	No
9	cool	Normal	weak	Yes
11	mild	Normal	strong	Yes

$S_{sunny} = 5 [2+, 3-]$

$$Entropy(S_{sunny}) = -\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right)$$

$= 0.97$

Temp Hot, mild, cool

$S_{hot} = 2 [0+, 2-] = 0$

Entropy $(S_{hot}) = 0$

$S_{mild} = 2 [1+, 1-] = 1$

$S_{cool} = 1 [1+, 0-] = 0$

$$Gain(S_{sunny}, Temp) = Entropy\ S - \frac{2}{5} \times Ent\ S_{hot}$$

$$- \frac{2}{5} Ent\ mild - \frac{1}{5} Entro\ cool$$

$$= 0.97 - \frac{2}{5} \times 0 - \frac{2}{5} \times 1 - \frac{1}{5} \times 0$$

$$= 0.97 - 0 - \frac{2}{5} - 0$$

$= 0.97 - 0.4 =$

$= 0.57$

attribute Humidity — High Normal

$S_{high} = 3\ [0+, 3-] = 0$

$S_{normal} = 2\ [2+, 0-] = 0$

$Gain = 0.97 - \frac{3}{5} \times S_{high} - \frac{2}{5} S_{normal}$

$= 0.97 - \frac{3}{5} \times 0 - \frac{2}{5} \times 0$

$\boxed{Gain = 0.97}$

attribute wind - strong, weak

$S_{strong} = 2\ [1+, 1-] = 1$

$S_{weak} = 3\ [1+, 2-]$

$S = -\frac{1}{3}\log_2 \frac{1}{3} - \frac{2}{3}\log_2 \frac{2}{3}$

$= 0.9183$

$Gain = 0.97 - \frac{2}{5} \times 1 - \frac{3}{5} \times 0.9183$

$= 0.97 - 0.4 - 0.6 \times 0.9183$

$\boxed{Gain = 0.019}$

∴ Tree becomes as follows

Outlook

Sunny — overcast — Rain

overcast → Yes

Rain → ?

Sunny → Humidity

Humidity: high → d1 d2 d8 → NO; low → d9 d11 → Yes

Day	Temp	Hum	wind	Play tennis
d4	mild	High	w	Y
d5	cool	Normal	w	Y
d6	cool	Normal	S	N
d10	mild	Normal	w	Y
d14	mild	High	S	N

Temp - mild cool

$S_{rain} = 5\ [3+, 2-]$

$Ent\ S_{rain} = -\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2 \frac{2}{5} = 0.97$

$S_{mild} = 3\ [2+, 1-]$

$= -\frac{2}{3}\log_2\left(\frac{2}{3}\right) - \frac{1}{3}\log_2\left(\frac{1}{3}\right)$

$= 0.9183$

$S_{cool} = 2\ [1+, 1-] = 1.0$

$Gain = 0.97 - \frac{3}{5} \times 0.9183 - \frac{2}{5} \times 1.0$

$= 0.0192$

Page No.:

attribute Humidity (High, Normal)

$S_{high} = 2\ [1+, 1-] = 1.0$

$S_{normal} = 3\ [2+, 1-] = 0.9183$

$Gain = 0.97 - \frac{2}{5} \times 1 - \frac{3}{5} \times 0.9183$

$= 0.0192$

attribute wind - (strong, weak)

$S_{strong} = 2\ [0+, 2-] = 0$

$S_{weak} = 3\ [3+, 0-] = 0$

$Gain = 0.97 - \frac{2}{5} \times 0 - \frac{3}{5} \times 0$

$\boxed{Gain = 0.97}$

check wind colm then it is showing weak 3 record & all are yes & strong 2 record & all are no. so we can put directly Y or N at leaf node so updated tree becomes

Outlook

S — O — R

O → Yes

S → Humidity: H → N, N → Y

R → wind: strong → N, weak → Yes

Q 4 a) Explain different types of Machine Learning and Main Challenges of Machine Learning [Types 3 M Challenges 2 M]

- There are so many different types of Machine Learning systems that it is useful to classify them in broad categories based on:

- Whether or not they are trained with human supervision (supervised, unsupervised, semi-supervised, and Reinforcement Learning)
- Whether or not they can learn incrementally on the fly (online versus batch learning)
- Whether they work by simply comparing new data points to known data points, or instead detect patterns in the training data and build a predictive model, much like scientists do (instance-based versus model-based learning)
- In *supervised learning*, the training data you feed to the algorithm includes the desired solutions, called *labels*
- In *unsupervised learning*, as you might guess, the training data is unlabeled.
- The system tries to learn without a teacher.
- Some algorithms can deal with partially labeled training data, usually a lot of unlabeled data and a little bit of labeled data. This is called *semisupervised learning*
- Reinforcement Learning: The learning system, called an *agent* in this context, can observe the environment, select and perform actions, and get *rewards* in return (or *penalties* in the form of negative rewards
- It must then learn by itself what is the best strategy, called a *policy*, to get the most reward over time.
- A policy defines what action the agent should choose when it is in a given situation.
- **Main Challenges of ML**
- Insufficient Quantity of Training Data
- Nonrepresentative Training Data
- Poor-Quality Data
- Irrelevant Features
- Overfitting the Training Data
- Underfitting the Training Data
- Testing and Validating

Q 4 b) Explain the concept of Entropy and Information Gain

Solution

In order to define information gain precisely, we begin by defining a measure commonly used in information theory, called ***entropy,*** that characterizes the (im)purity of an arbitrary collection of examples.

Given a collection S, containing positive and negative examples of some target concept, the entropy of S relative to this boolean classification is

$$Entropy(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

where p, is the proportion of positive examples in S and p, is the proportion of negative examples in S. In all calculations involving entropy we define 0 log 0 to be 0.

To illustrate, suppose S is a collection of 14 examples of some Boolean concept, including 9 positive and 5 negative examples (we adopt the notation [9+, **5-]** to summarize such a sample of data). Then the entropy of S relative to this boolean classification is

$$\begin{aligned} Entropy([9+, 5-]) &= -(9/14) \log_2(9/14) - (5/14) \log_2(5/14) \\ &= 0.940 \end{aligned}$$

Given entropy as a measure of the impurity in a collection of training examples, we can now define a measure of the effectiveness of an attribute in classifying the training data. The measure we will use, called ***information gain,*** is simply the expected reduction in entropy caused by partitioning the examples according to this attribute. More precisely, the information gain, ***Gain(S, A)*** of ***an*** attribute **A,** relative to a collection of examples ***S,*** is defined as

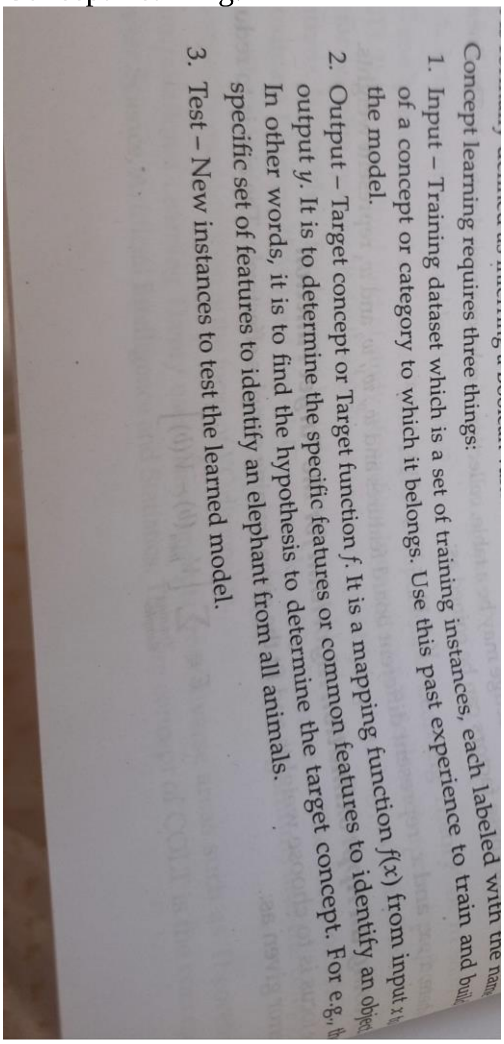
$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

where ***Values(A)*** is the set of all possible values for attribute A, and ***S,*** is the subset of S for which attribute A has value v (i.e., ***S,*** = ***{s E SIA(s)*** = ***v))***

Q 5) Define Following Terms:

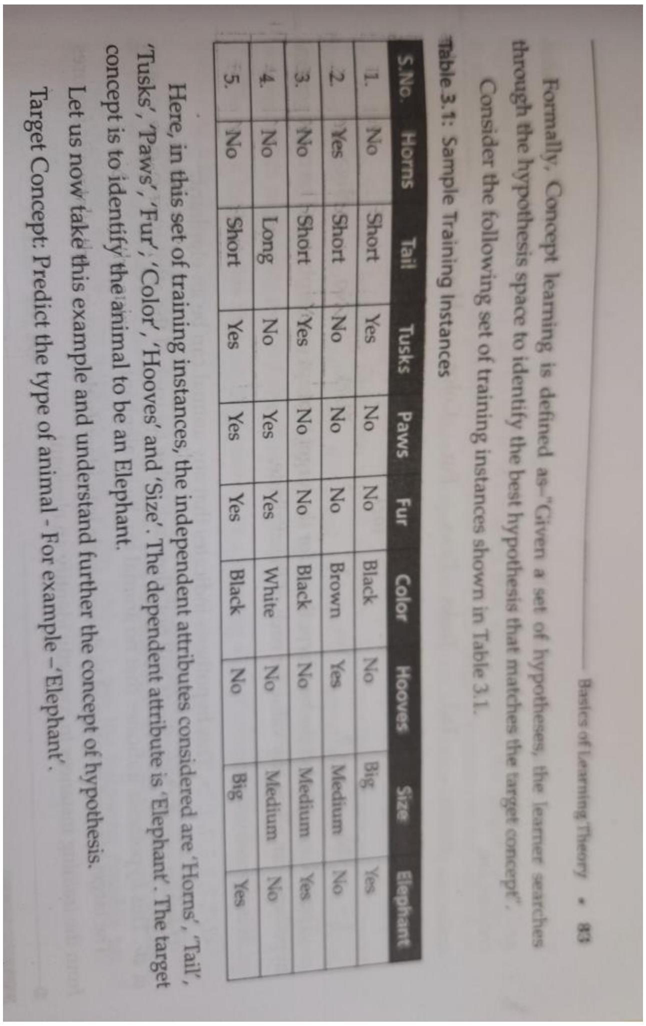
a) Concept Learning
b) Version Space
c) Consistent Hypothesis
d) General Boundary
e) Specific Boundary

Concept Learning:



Concept learning requires three things:

1. Input – Training dataset which is a set of training instances, each labeled with the name of a concept or category to which it belongs. Use this past experience to train and build the model.
2. Output – Target concept or Target function *f*. It is a mapping function *f*(*x*) from input *x* to output *y*. It is to determine the specific features or common features to identify an object. In other words, it is to find the hypothesis to determine the target concept. For e.g., the specific set of features to identify an elephant from all animals.
3. Test – New instances to test the learned model.



Basics of Learning Theory • 83

Formally, Concept learning is defined as–"Given a set of hypotheses, the learner searches through the hypothesis space to identify the best hypothesis that matches the target concept".

Consider the following set of training instances shown in Table 3.1.

Table 3.1: Sample Training Instances

S.No.	Horns	Tail	Tusks	Paws	Fur	Color	Hooves	Size	Elephant
1.	No	Short	Yes	No	No	Black	No	Big	Yes
2.	Yes	Short	No	No	No	Brown	Yes	Medium	No
3.	No	Short	Yes	No	No	Black	No	Medium	Yes
4.	No	Long	No	Yes	Yes	White	No	Medium	No
5.	No	Short	Yes	Yes	Yes	Black	No	Big	Yes

Here, in this set of training instances, the independent attributes considered are 'Horns', 'Tail', 'Tusks', 'Paws', 'Fur', 'Color', 'Hooves' and 'Size'. The dependent attribute is 'Elephant'. The target concept is to identify the animal to be an Elephant.

Let us now take this example and understand further the concept of hypothesis.

Target Concept: Predict the type of animal - For example –'Elephant'.

Version Space:

A hypothesis *h* is **consistent** with a set of training examples *D* of target concept *c* if and only if *h(x)=c(x)* for each training example in *D.*

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

The **version space**, $VS_{H,D}$, with respect to hypothesis space *H* and training examples *D*, is the subset of hypotheses from *H* consistent with all training examples in *D.*

$$VS_{H,D} \equiv \{h \in H \mid Consistent(h, D)\}$$

Example Version Space

G: { <?,?,Round,?,?> <?,Triangle,?,?,?> }

<?,?,Round,?,Yes> <?,Triangle,Round,?,?> <?,Triangle,?,?,Yes>

S; { <?,Triangle,Round,?,Yes> }

Representing Version Spaces

The **General boundary**, G, of version space $VS_{H,D}$ is the set of its maximally general members.

The **Specific boundary**, S, of version space $VS_{H,D}$ is the set of its maximally specific members.

Consistent Hypothesis:

We need to take the combination of sets in ‘G’ and check that with ‘S’. When the combined set fields are matched with fields in ‘S’, then only that is included in the version space as consistent hypothesis

General Boundary:

The **General boundary**, G, of version space $VS_{H,D}$ is the set of its maximally general members.

Every member of the version space lies between General and Specific boundaries

$$VS_{H,D} = \{h \in H \mid (\exists s \in S)(\exists g \in G)(g \geq h \geq s)\}$$

where $x \geq y$ means x is more general or equal to y

Specific Boundary:

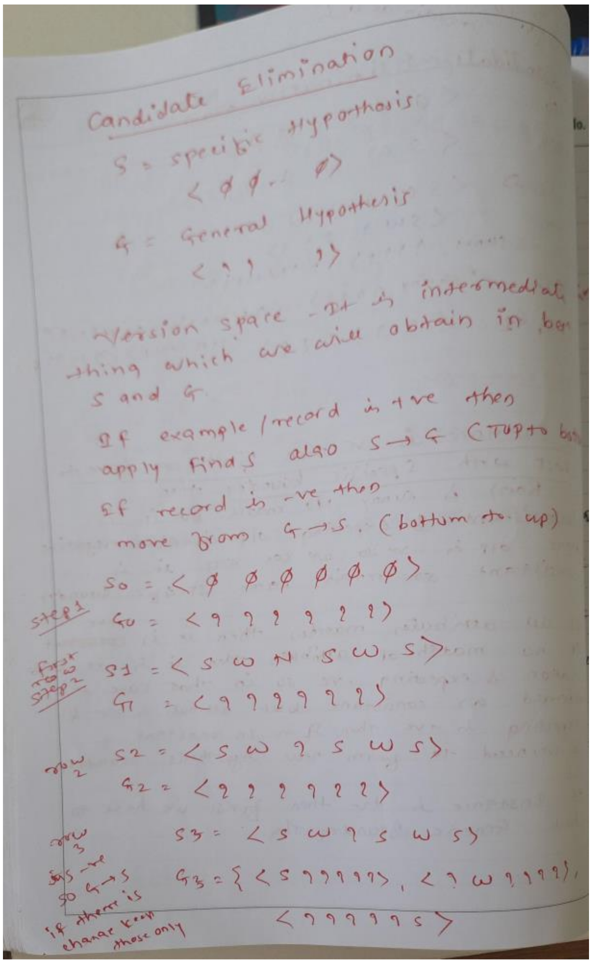
The **Specific boundary**, S, of version space $VS_{H,D}$ is the set of its maximally specific members.

Q 6) Explain Candidate Elimination Algorithm with Example

G = maximally general hypotheses in *H*
S = maximally specific hypotheses in *H*
For each training example *d*, do
- If ***d*** **is a positive example**
  - Remove from G any hypothesis that does not include *d*
  - For each hypothesis *s* in S that does not include *d*
    - Remove *s* from S
    - Add to S all minimal generalizations *h* of *s* such that
      1. *h* includes *d*, and
      2. Some member of G is more general than *h*
    - Remove from S any hypothesis that is more general than another hypothesis in S
- If ***d*** **is a negative example**
  - Remove from S any hypothesis that does include *d*
  - For each hypothesis *g* in G that does include *d*
    - Remove *g* from G
    - Add to G all minimal generalizations *h* of *g* such that
      1. *h* does not include *d*, and

2. Some member of S is more specific than *h*
Remove from G any hypothesis that is less general than another hypothesis in G

If G or S ever becomes empty, data not consistent (with H)



Candidate Elimination

S = specific Hypothesis
< ϕ ϕ .. ϕ>

G = General Hypothesis
< ? ? ... ?>

Version space - It is intermediate thing which we will obtain in between S and G

If example / record is +ve then apply Find S algo S → G (Top to bottom)

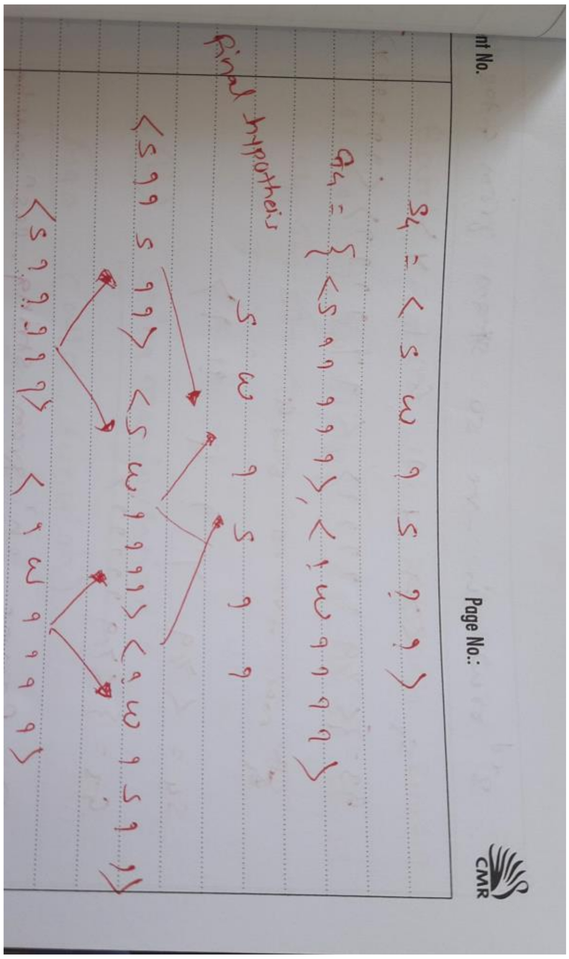
If record is -ve then move from G → S (bottom to up)

S0 = < ϕ ϕ ϕ ϕ ϕ ϕ>
step 1 G0 = <? ? ? ? ? ?>

First row step 2 S1 = <S W N S W S>
G1 = <? ? ? ? ? ?>

row 2 S2 = <S W ? S W S>
G2 = <? ? ? ? ? ?>

row 3 S3 = <S W ? S W S>
3rd is -ve so G → S if there is change keep those only
G3 = {<S ? ? ? ? ?>, <? W ? ? ? ?>, <? ? ? ? ? S>}



S4 = <S W ? S ? ?>

G4 = {<S ? ? ? ? ?>, <? W ? ? ? ?>}

Final hypothesis

S W ? S ? ?

<S ? ? S ? ?> <S W ? ? ? ?> <? W ? S ? ?>

<S ? ? ? ? ?> <? W ? ? ? ?>