Seventh Semester B.E. Degree Examination, Dec.2019/Jan.2020

Machine Learning

CYPHER LUNATIC
Back Benchers Association

Time: 3 hrs. Max. Marks: 80

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

Module-1

1	a.	What do you mean by well-posed learning problem? Explain with example.	**(04 Marks)**
	b.	Explain the various stages involved in designing a learning system in brief.	**(08 Marks)**
	c.	Write Find_S algorithm and discuss the issues with the algorithm.	**(04 Marks)**

1A)

WELL-POSED LEARNING PROBLEMS

Definition: A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

To have a well-defined learning problem, three features needs to be identified:

1. The class of tasks
2. The measure of performance to be improved
3. The source of experience

Examples

1. ***Checkers game:*** A computer program that learns to play ***checkers*** might improve its performance as measured by its ability to win at the class of tasks involving playing checkers games, through experience obtained by playing games against itself.

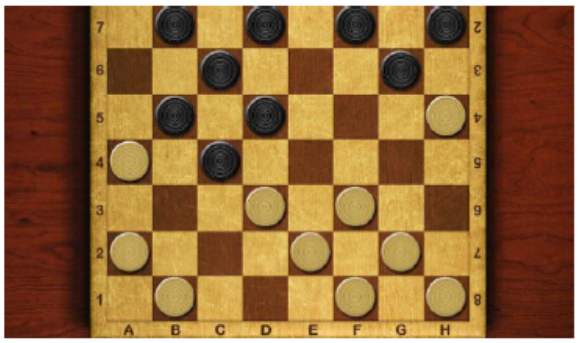


Fig: Checker game board

***A checkers learning problem*:**

- Task T: playing checkers
- Performance measure P: percent of games won against opponents
- Training experience E: playing practice games against itself

2. ***A handwriting recognition learning problem:***
   - Task T: recognizing and classifying handwritten words within images
   - Performance measure P: percent of words correctly classified
   - Training experience E: a database of handwritten words with given classifications

3. ***A robot driving learning problem*:**
    - Task T: driving on public four-lane highways using vision sensors
    - Performance measure P: average distance travelled before an error (as judged by human overseer)
    - Training experience E: a sequence of images and steering commands recorded while observing a human driver

1B)

DESIGNING A LEARNING SYSTEM

The basic design issues and approaches to machine learning are illustrated by designing a program to learn to play checkers, with the goal of entering it in the world checkers tournament

1. Choosing the Training Experience
2. Choosing the Target Function
3. Choosing a Representation for the Target Function
4. Choosing a Function Approximation Algorithm
    1. Estimating training values
    2. Adjusting the weights
5. The Final Design

1. Choosing the Training Experience

- The first design choice is to choose the type of training experience from which the system will learn.
- The type of training experience available can have a significant impact on success or failure of the learner.

There are three attributes which impact on success or failure of the learner

1. Whether the training experience provides ***direct or indirect feedback*** regarding the choices made by the performance system.

2. The degree to which the ***learner controls the sequence of training examples***

3. How well it represents the ***distribution of examples*** over which the final system performance P must be measured

2. Choosing the Target Function

The next design choice is to determine exactly what type of knowledge will be learned and how this will be used by the performance program.

Let's consider a checkers-playing program that can generate the legal moves from any board state.

Let's consider a checkers-playing program that can generate the legal moves from any board state.
The program needs only to learn how to choose the best move from among these legal moves. We must learn to choose among the legal moves, the most obvious choice for the type of information to be learned is a program, or function, that chooses the best move for any given board state.

1. Let ***ChooseMove*** be the target function and the notation is

$$\textit{ChooseMove} : B \rightarrow M$$

2. An alternative target function is an ***evaluation function*** that assigns a ***numerical score*** to any given board state
   Let the target function V and the notation

$$V : B \rightarrow R$$

3. Choosing a Representation for the Target Function

Let's choose a simple representation - for any given board state, the function ***c*** will be calculated as a linear combination of the following board features:

- x_1: the number of black pieces on the board
- x_2: the number of red pieces on the board
- x_3: the number of black kings on the board
- x_4: the number of red kings on the board
- x_5: the number of black pieces threatened by red (i.e., which can be captured on red's next turn)
- x_6: the number of red pieces threatened by black

Thus, learning program will represent as a linear function of the form

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

Where,

- w_0 through w_6 are numerical coefficients, or weights, to be chosen by the learning algorithm.
- Learned values for the weights w1 through w6 will determine the relative importance of the various board features in determining the value of the board
- The weight w_0 will provide an additive constant to the board value

4. Choosing a Function Approximation Algorithm

In order to learn the target function **f** we require a set of training examples, each describing a specific board state b and the training value $V_{train}(b)$ for b.

Each training example is an ordered pair of the form (b, $V_{train}(b)$).

Estimating Training Values

Need to assign specific scores to intermediate board states

• Approximate intermediate board state b using the learner's current approximation of the next board state following b

$$V_{train}(b) \leftarrow \hat{V}(Successor(b))$$

• Simple and successful approach

• More accurate for states closer to end states

Adjusting the Weights

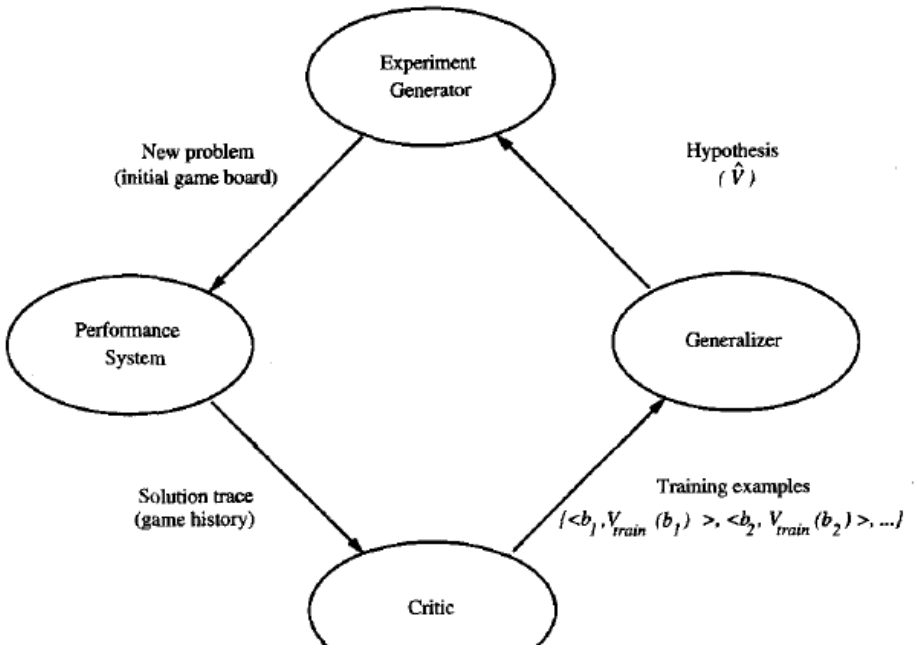
• Choose the weights wi to best fit the set of training examples

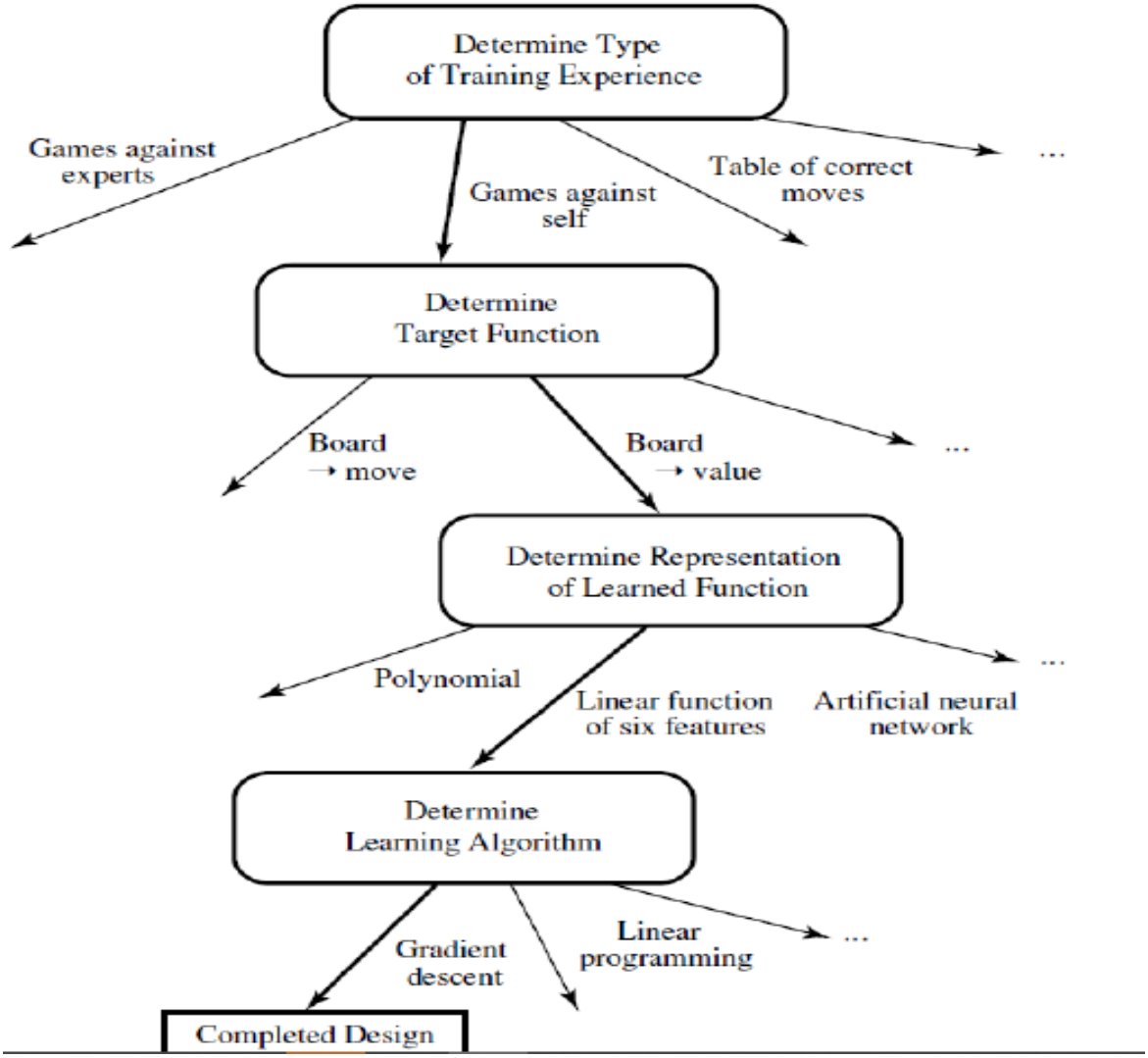
• Minimize the squared error E between the train values and the values predicted by the hypothesis

$$E \equiv \sum_{\langle b, V_{train}(b) \rangle \in training examples} \left(V_{train}(b) - \hat{V}(b) \right)^2$$

5. The Final Design

The final design of checkers learning system can be described by four distinct program modules that represent the central components in many learning systems





1C)

FIND-S: FINDING A MAXIMALLY SPECIFIC HYPOTHESIS

FIND-S Algorithm

1. Initialize h to the most specific hypothesis in H
2. 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
3. Output hypothesis h

Limitations of Find-S Algorithm

There are a few limitations of the Find-S algorithm listed down below:

There is no way to determine if the hypothesis is consistent throughout the data.

Inconsistent training sets can actually mislead the Find-S algorithm, since it ignores the negative examples.

Find-S algorithm does not provide a backtracking technique to determine the best possible changes that could be done to improve the resulting hypothesis.

---

2 a. List the issues in machine learning. **(04 Marks)**

b. Consider the given below training example which finds malignant tumors from MRI scans.

Example	Shape	Size	Color	Surface	Thickness	Target concept
1	Circular	Large	Light	Smooth	Thick	Malignant
2	Circular	Large	Light	Irregular	Thick	Malignant
3	Oval	Large	Dark	Smooth	Thin	Benign
4	Oval	Large	Light	Irregular	Thick	Malignant
5	Circular	Small	Light	Smooth	Thick	Benign

Show the specific and general boundaries of the version space after applying candidate elimination algorithm. (Note: Malignant is +ve, Benign is –ve). **(08 Marks)**

c. Explain the concept of inductive bias in brief. **(04 Marks)**

2A)

Issues in Machine Learning

The field of machine learning, and much of this book, is concerned with answering questions such as the following

- 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?

- 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?
- What is the best way to reduce the learning task to one or more function approximation problems? Put another way, what specific functions should the system attempt to learn? Can this process itself be automated?
- How can the learner automatically alter its representation to improve its ability to represent and learn the target function?

2B) PROBLEM NOT AVAILABLE REFER WHATEVER

2C)

INDUCTIVE BIAS

The fundamental questions for inductive inference

1. What if the target concept is not contained in the hypothesis space?
2. Can we avoid this difficulty by using a hypothesis space that includes every possible hypothesis?
3. How does the size of this hypothesis space influence the ability of the algorithm to generalize to unobserved instances?
4. How does the size of the hypothesis space influence the number of training examples that must be observed?

These fundamental questions are examined in the context of the CANDIDATE-ELIMINTION algorithm

---

Module-1

1 a. Define machine learning. Describe the steps in designing learning system. (08 Marks)
b. Write Find-S algorithm and explain with example. (04 Marks)
c. Explain List-Then-Eliminate algorithm. (04 Marks)

1A)

1. What is Machine learning? Give Example

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed . For example: Robots are programmed so that they can perform the task based on data they gather from sensors.

Steps in designing REPEATED

2b)repeated FIND -S example

To illustrate this algorithm, assume the learner is given the sequence of training examples from the ***EnjoySport*** task

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 first step of FIND-S is to initialize h to the most specific hypothesis in H

  h - (Ø, Ø, Ø, Ø, Ø, Ø)

- Consider the first training example

  x_1 = <Sunny Warm Normal Strong Warm Same>, +

Observing the first training example, it is clear that hypothesis *h* is too specific. 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>

- Consider the second training example

  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>

- Consider the third training example

  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.

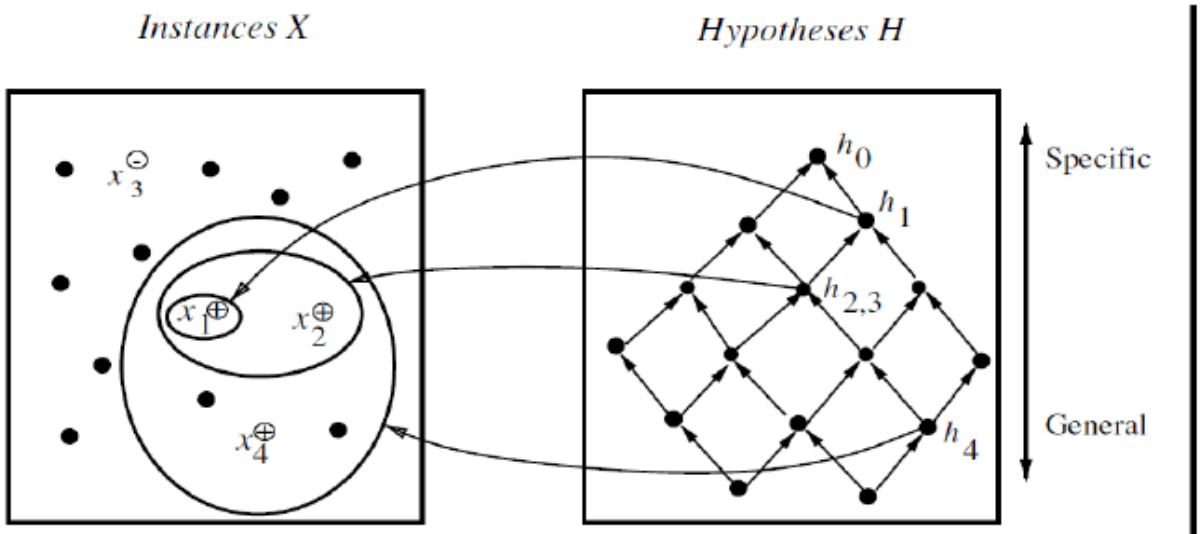
  h_3 = < Sunny Warm ? Strong Warm Same>

- Consider the fourth training example

  x_4 = <Sunny Warm High Strong Cool Change>, +

  The fourth example leads to a further generalization of h

  h_4 = < Sunny Warm ? Strong ? ? >



x_1 = <*Sunny Warm Normal Strong Warm Same*>, +
x_2 = <*Sunny Warm High Strong Warm Same*>, +
x_3 = <*Rainy Cold High Strong Warm Change*>, -
x_4 = <*Sunny Warm High Strong Cool Change*>, +

h_0 = <∅, ∅, ∅, ∅, ∅, ∅>
h_1 = <*Sunny Warm Normal Strong Warm San*
h_2 = <*Sunny Warm ? Strong Warm Same*>
h_3 = <*Sunny Warm ? Strong Warm Same*>
h_4 = <*Sunny Warm ? Strong ? ?* >

1c)

The LIST-THEN-ELIMINATION 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 $h(x) \neq c(x)$
3. Output the list of hypotheses in *VersionSpace*

The LIST-THEN-ELIMINATE Algorithm

OR

2 a. List out any 5 applications of machine learning. **(05 Marks)**

b. What do you mean by hypothesis space, instance space and version space? **(03 Marks)**

c. Find the maximally general hypothesis and maximally specific hypothesis for the training examples given in the table using candidate elimination algorithm. **(08 Marks)**

Day	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

2a)

Some successful applications of machine learning

- Learning to recognize spoken words
- Learning to drive an autonomous vehicle
- Learning to classify new astronomical structures
- Learning to play world-class backgammon

1. Image Recognition: Image recognition is one of the most common applications of machine learning. It is used to identify objects, persons, places, digital images, etc.
2. Speech Recognition **While using Google, we get an option of "**Search by voice**," it comes under speech recognition, and it's a popular application of machine learning.**
3. Traffic prediction: **If we want to visit a new place, we take help of Google Maps, which shows us the correct path with the shortest route and predicts the traffic conditions.**
4. Self-driving cars: **One of the most exciting applications of machine learning is self-driving cars. Machine learning plays a significant role in self-driving cars. Tesla, the most popular car manufacturing company is working on self-driving car.**
5. Email Spam and Malware Filtering: **Whenever we receive a new email, it is filtered automatically as important, normal, and spam. We always receive an important mail in our inbox with the important symbol and spam emails in our spam box, and the technology behind this is Machine learning.**

2b)

Hypothesis space 'h' is described by a conjunction of constraints on the attribute, the constraints may General hypothesis "?" (any value is acceptable), Specific hypothesis "φ" (a specific value or no value is accepted).

Instance Space: It is a subset of all possible example or instance.

Version Space: The Version Space denotes VSHD (with respect to hypothesis space H and training example D) is the subset of hypothesis from H consistent with training example in D.

2c)

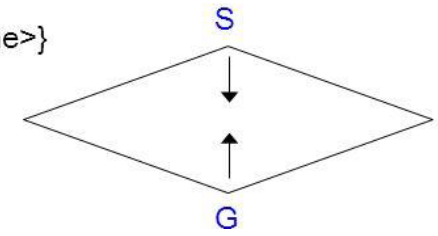
Candidate-Elimination Algorithm

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

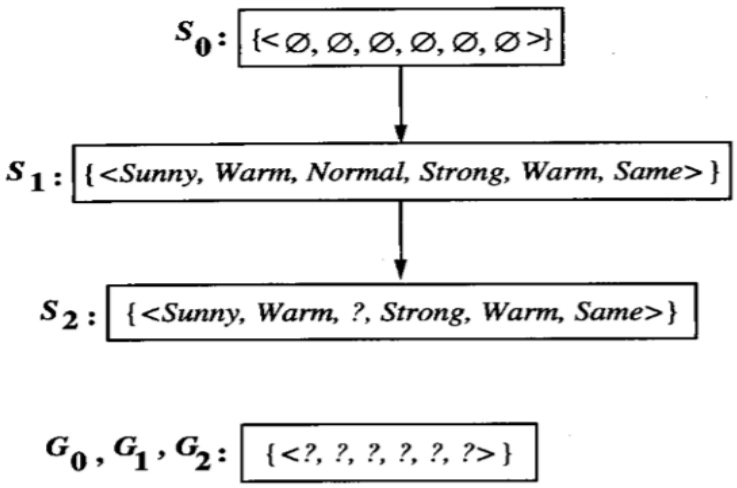
S_0 = {<∅, ∅, ∅, ∅, ∅, ∅>}
G_0 = {<?, ?, ?, ?, ?, ?>}

S_1 = {<Sunny, Warm, Normal, Strong, Warm, Same>}
G_1 = {<?, ?, ?, ?, ?, ?>}

S_2 = {<Sunny, Warm, ?, Strong, Warm, Same>}
G_2 = {<?, ?, ?, ?, ?, ?>}



Trace1:



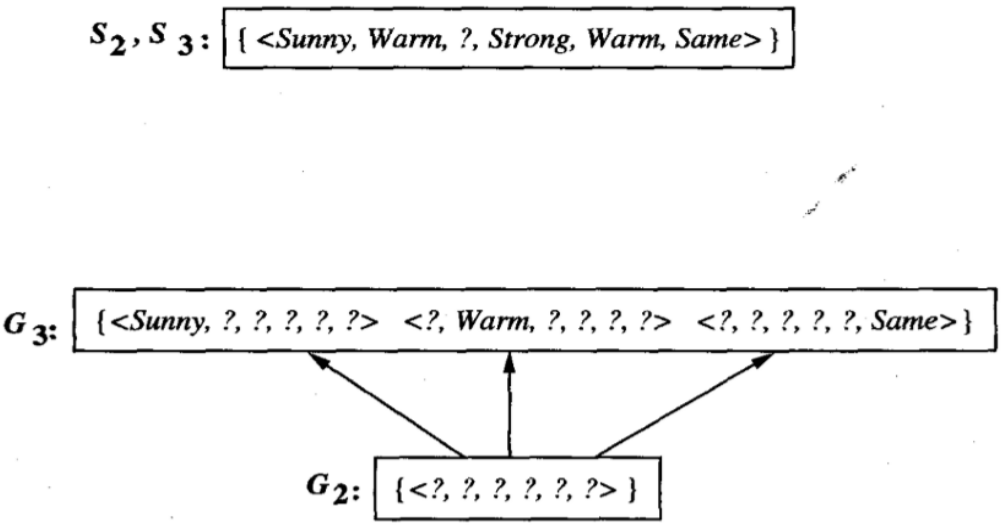
Training examples:

1. <*Sunny, Warm, Normal, Strong, Warm, Same*>, *Enjoy Sport = Yes*

2. <*Sunny, Warm, High, Strong, Warm, Same*>, *Enjoy Sport = Yes*

CANDIDATE-ELIMINATION Trace 1. S_0 and G_0 are the initial boundary sets corresponding to the most specific and most general hypotheses. Training examples 1 and 2 force the S boundary to become more general, as in the FIND-S algorithm. They have no effect on the G boundary.

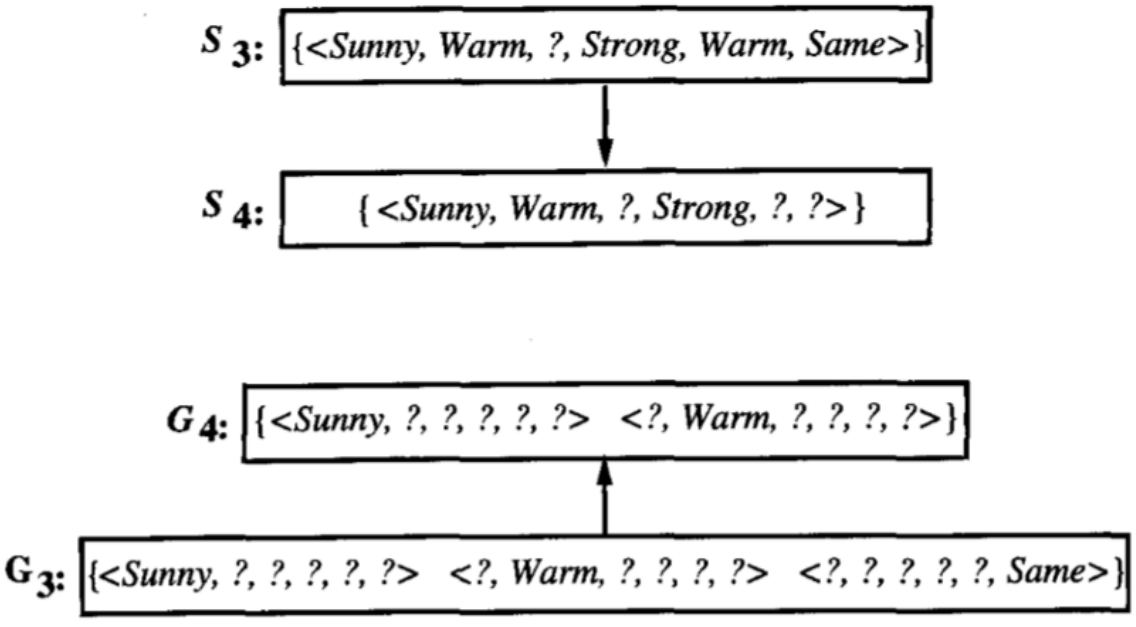
Trace 2:



Training Example:

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

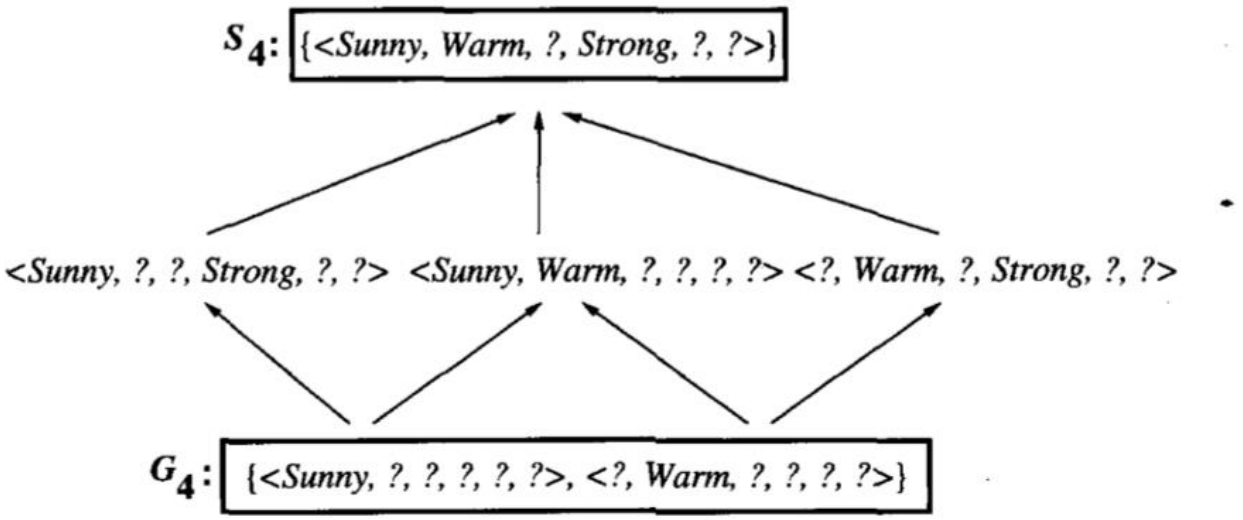
CANDIDATE-ELIMINATION Trace 2. Training example 3 is a negative example that forces the G_2 boundary to be specialized to G_3. Note several alternative maximally general hypotheses are included in G_3.



Training Example:

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

CANDIDATE-ELIMINATION Trace 3. The positive training example generalizes the S boundary, from S_3 to S_4. One member of G_3 must also be deleted, because it is no longer more general than the S_4 boundary.



The final version space for the *EnjoySport* concept learning problem and training examples described earlier.

Module-1

1	a.	Specify the learning task for 'A checkers learning problem'.	(03 Marks)
	b.	Discuss the following with respect to the above,	
		(i) Choosing the training experience.	
		(ii) Choosing the target function and	
		(iii) Choosing a function approximation algorithm.	(09 Marks)
	c.	Comment on the issues in machine learning.	(04 Marks)

1a) repeated

1b)repeated(first paper 1b) sub answers

1c) 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?

• What is the best way to reduce the learning task to one or more function approximation problems? Put another way, what specific functions should the system attempt to learn? Can this process itself be automated?
• How can the learner automatically alter its representation to improve its ability to represent and learn the target function?

OR

2 a. Write candidate elimination algorithm. Apply the algorithm to obtain the final version space for the training example. **(10 Marks)**

Sl. No.	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

b. Discuss about an unbiased Learner. **(06 Marks)**

2a)repeated previous paper

2b)

An Unbiased Learner

- The solution to the problem of assuring that the target concept is in the hypothesis space H is to provide a hypothesis space capable of representing every teachable concept that is representing every possible subset of the instances X.
- The set of all subsets of a set X is called the power set of X

  - In the ***EnjoySport*** learning task the size of the instance space X of days described by the six attributes is 96 instances.
  - Thus, there are 2^{96} distinct target concepts that could be defined over this instance space and learner might be called upon to learn.
  - The conjunctive hypothesis space is able to represent only 973 of these - a biased hypothesis space indeed

  - Let us reformulate the ***EnjoySport*** learning task in an unbiased way by defining a new hypothesis space H' that can represent every subset of instances
  - The target concept "Sky = Sunny or Sky = Cloudy" could then be described as

(Sunny, ?, ?, ?, ?, ?) v (Cloudy, ?, ?, ?, ?, ?)