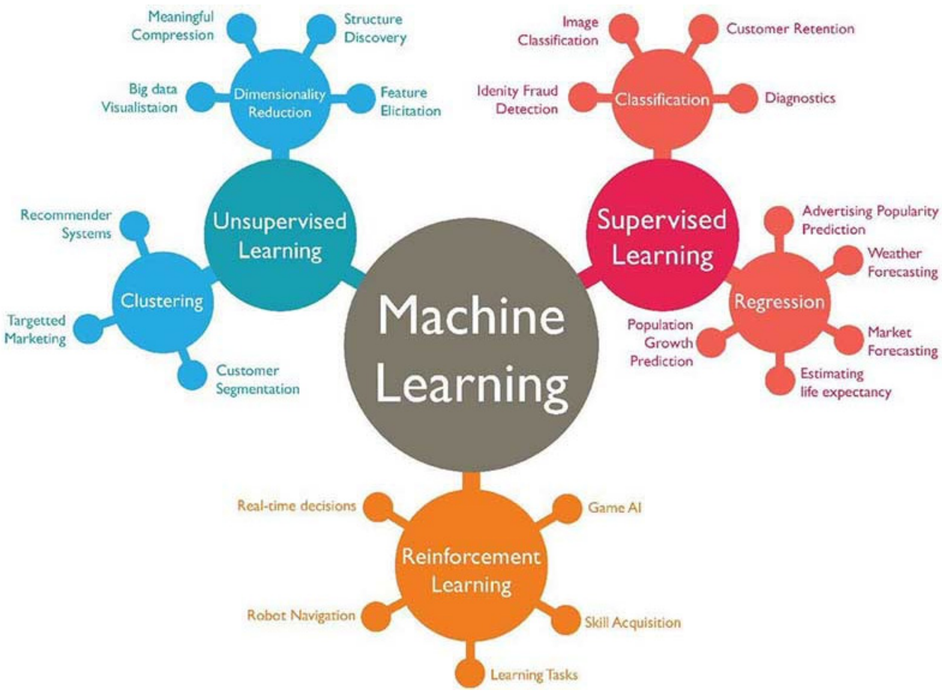
Unit 1 Machine Learning

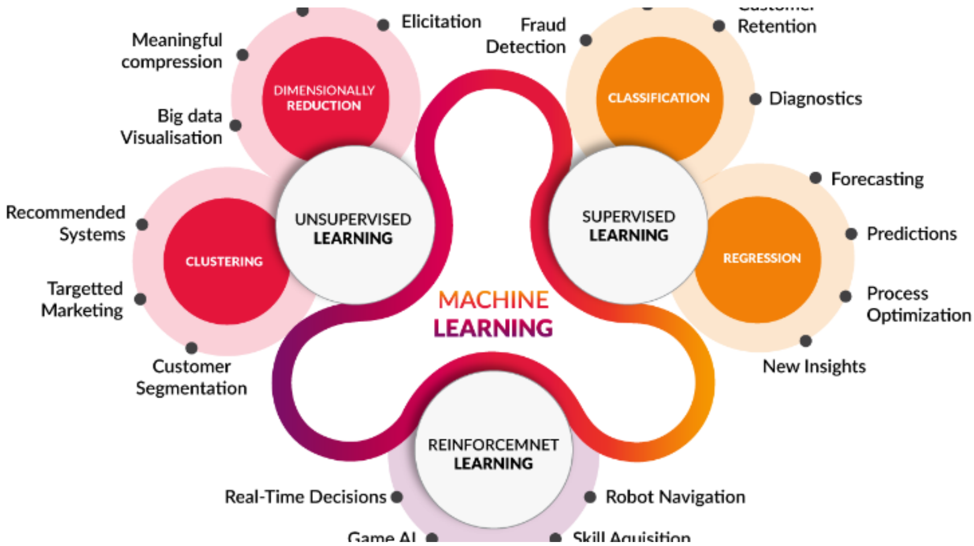


For Micro Notes by the Student

Chapter - 1

INTRODUCTION TO MACHINE LEARNING:

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.



Well Posed Learning Problems with Examples:

- ➢ Learning = Improving with experience at some task.
- ➢ Improve over task T.
  - With respect to performance measure P. Based on experience E.
- ➢ What are T, P, E ? How do we formulate a machine learning problem

- **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

- **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

- **A robot driving learning problem:**

  Task T : driving on public four-lane highways using vision sensors
  Performance measure P : average distance traveled before an error (as judged by human overseer)

Training experience E : a sequence of images and steering commands recorded while observing a human driver.

For Micro Notes by the Student

Designing a Learning System:

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

1. Choosing the Training Experience

q). Whether the training experience provides direct or indirect feedback regarding the choices made by the performance system:

Example: –

Direct training examples in learning to play checkers consist of individual checkers board states and the correct move for each. – Indirect training examples in the same game consist of the move sequences and final outcomes of various games played in which information about the correctness of specific moves early in the game must be inferred indirectly from the fact that the game was eventually won or lost – credit assignment System.

b).The degree to which the learner controls the sequence of training examples:

Example:

The learner might rely on the teacher to select informative board states and to provide the correct move for each – The learner might itself propose board states that it finds particularly confusing and ask the teacher for the correct move.
Or the learner may have complete control over the board states and (indirect) classifications, as it does when it learns by playing against itself with no teacher present.

c). How well it represents the distribution of examples over which the final system performance P must be measured: In general learning is most reliable when the training examples follow a distribution similar to that of future test examples.

Example:

If the training experience in play checkers consists only of games played against itself, the learner might never encounter certain crucial board states that are very likely to be played by the human checkers champion. (Note however that the most current theory of machine learning rests on the crucial assumption that the distribution of training examples is identical to the distribution of test examples)

For Micro Notes by the Student

2.Choosing the Target Function:

a).To determine what type of knowledge will be learned and how this will be used by the performance program:

Example:

In play checkers, it needs to learn to choose the best move among those legal moves: Choose Move:
B → M, which accepts as input any board from the set of legal board states B and produces as output some move from the set of legal moves M.

b). Since the target function such as Choose Move turns out to be very difficult to learn given the kind of indirect training experience available to the system, an alternative target function is then an evaluation function that assigns a numerical score to any given board state, V: B → R.

3. Choosing a Representation for the target function:

Given the ideal target function V, we choose a representation that the learning system will use to describe V' that it will learn:

Example: –

In play checkers,

V'(b) = w 0 + w 1 x 1 + w 2x 2 + w 3x 3 + w 4x 4 + w 5x 5 + w 6x 6 –

where wi is the numerical coefficient or weight to determine the relative importance of the various board features and xi is the number of i-th objects on the board.

4. Choosing a Function Approximation Algorithm

- Each training example is given by where Vtrain(b) is the training value for a board b.
- Estimating Training Values: Vtrain (b) ← V' (Successor (b)).
- Adjusting the weights: To specify the learning algorithm for choosing the weights wi to best fit the set of training examples {},
- which minimizes the squared error E between the training values and the values predicted $E = \sum_{<b, Vtrain(b)> \in \text{training examples}} (V_{train}(b) - v'(b))^2$ by the hypothesis.
- To minimize E, the following rule is used: LMS weight update rule
  For each training example Use the current weights to calculate
  V'(b) For each weight wi ,
  update it as wi ← wi+ η (Vtrain(b) – V'(b)) xi.

5. The Final Design:

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

For Micro Notes by the Student

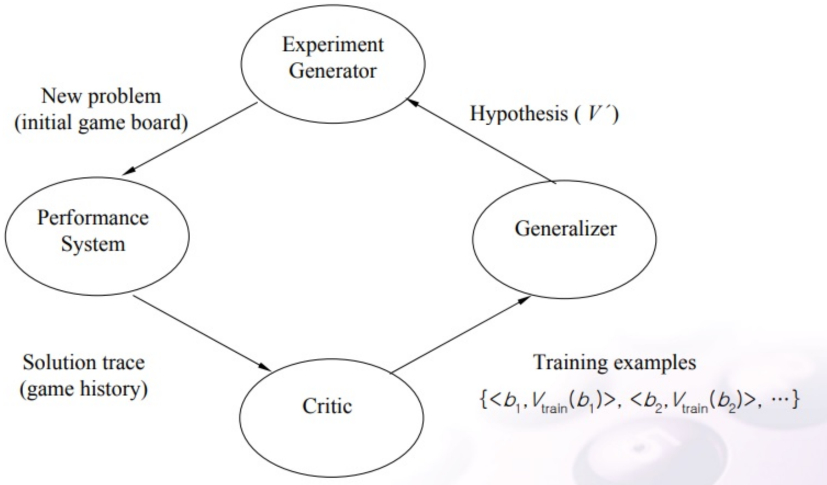


Figure 1.1 Final design of the checkers learning program

systems. The four are explained with characteristics below.

- Performance System: To solve the given performance task by using the learned target function(s). It takes an instance of a new problem (new game) as input and a trace of its solution (game history) as output.

- Critic: To take as input the history or trace of the game and produce as output a set of training examples of the target function.

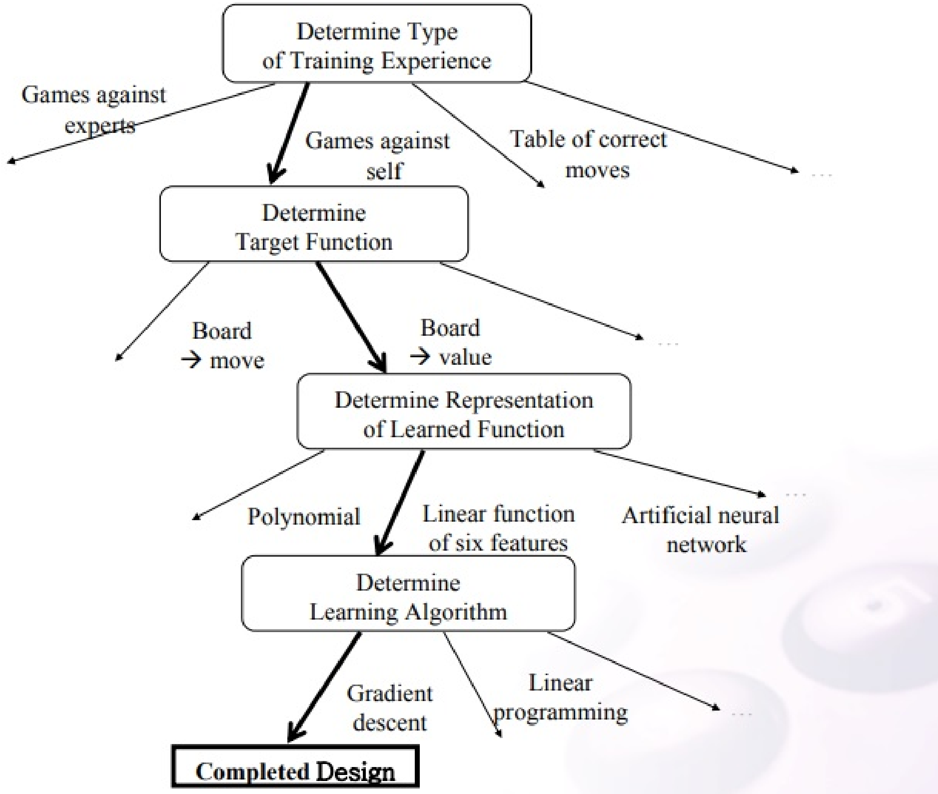
- Generalizer: To take as input the training examples and produce an output hypothesis that is its estimate of the target function. It generalizes from the specific training examples, hypothesizing a general function that covers these examples and other cases beyond the training examples.

- Experiment Generator: To take as input the current hypothesis (currently learned function) and outputs a new problem (i.e., initial board state) for Performance System to explore. Its role is to pick new practice problems that will maximize the learning rate of the overall system.

Together, the design choices we made for our checkers program produce specific imitations for the performance systems, critic, generalize and experiment generator. Many machine learning systems can be usefully characterized in terms of these four generic modules.

Choices in Designing the Checkers Learning Problem

For Micro Notes by the Student



Issues in Machine Learning:

- What algorithms exist for learning general target functions from specific training examples ?
- How does the number of training examples influence accuracy ?
- When and how can prior knowledge held by the learner guide the process of generalizing from examples ?
- 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 ?
- How can the learner automatically alter its representation to improve its ability to represent and learn the target function ?

For Micro Notes by the Student

Chapter - 2

CONCEPT LEARNING AND THE GENERAL-TO-SPECIFIC ORDERING

Contents of Concept Learning:

- ✓ Learning from examples
- ✓ General-to-specific ordering over hypotheses
- ✓ Version spaces and candidate elimination algorithm
- ✓ Picking new examples

The Concept Learning Problem Concept:

- ➢ Subset of objects defined over a set
- ➢ Boolean-valued function defined over a set
- ➢ Set of instances
- ➢ Syntactic definition Concept Learning
- ➢ Inferring a Boolean-valued function from training examples of its
- ➢ input and output Automatic inference of the general definition of some concept, given examples labelled as members or non-members of the concept .

Given:

A set E = {e1, e2, . . . , en} of training instances of concepts, each labeled with the name of a concept C1, C2, . . . , Ck to which it belongs

Determine:

The definitions of each of C1, C2, . . . , Ck which correctly cover E. Each definition is a concept description

Example: Learning Conjunctive Boolean Concepts Instances space: {0, 1} n
Concept is binary function c : {0, 1} n → {0, 1}

Inputs: n-bit patterns
Outputs: 0 or 1
C = set of all c which have conjunctive representation
Learning task: Identify a conjunctive concept that is consistent with the examples

C . Learning Conjunctive Boolean Concepts

Learning algorithm:

1. Initialize: L = {x1, x¯1, . . . , xn, x¯n}
2. Predict the label on the input X based on the conjunction of literals in L
3. If a mistake is made, eliminate the offending literals from L

Theorem: The above algorithm is guaranteed to learn any conjunctive Boolean concept given a non-contradictory sequence of examples in a noise-free environment. The bound on the number of mistakes is n + 1.

- Target function $f \in F$: unknown to the learner
- Hypothesis $h \in H$, about what f might be H — Hypothesis space
- Instance space X: domain of f, h
- Output space Y : range of f, h
- Example: an ordered pair (x, y), $x \in X$ and $f(x) = y \in Y$
- F and H may or may not be the same!
- Training set E: a multi-set of examples
- Learning algorithm L: a procedure which given some E, outputs an $h \in H$ 5.

Dimensions of Concept Learning Representation:

1. Instances

• Symbolic
• Numeric

2. Hypotheses (i.e., concept description)

• Attribute-value (propositional logic)
• Relational (first-order logic)

3. Semantic associated with both representations Level of learning

• Symbolic
• Sub-symbolic Method of learning

1. Bottom-up (covering)
2. Top-down

Case Study: Concept of EnjoySport
What is the general concept?

Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

Representing Hypotheses (Many possible representations) Here, h is conjunction of constraints on attributes.

Each constraint can be

• a specific value (e.g., Water = Warm)
• don't care (e.g., "Water =?")
• no value allowed (e.g.,"Water=ϕ")

For example:

Sky	Air Temp	Humid	Wind	Water	Forecast
Sunny	?	?	Strong	?	Same

Prototypical Concept Learning Task • Given: – Instances X: Possible days, each described by the attributes Sky, AirTemp, Humidity, Wind, Water, Forecast – Target function c: EnjoySport : X → {0, 1} –

Hypotheses H: Conjunctions of literals. E.g. h?, Cold, High, ?, ?, ?
Training examples D: Positive and negative examples of the target function hx1, c(x1)i, . . . hxm, c(xm)i

• Determine:
A hypothesis h in H such that h(x) = c(x) for all x in D.

The inductive learning hypothesis:
Any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over other unobserved examples. Instance, Hypotheses, and More-General-
Than h = h = h = 2 h h 3 h Instances X
Hypotheses H
Specific General 1 x 2 x x = x = 1 1 2 1 2 3
Find-S Algorithm
1. Initialize h to the most specific hypothesis in H
2. For each positive training instance x

• For each attribute constraint ai in h If the constraint
ai in h is satisfied by x
Then do nothing
Else
replace ai in h by the next more general constraint that is satisfied by x 3.
Output hypothesis h Hypothesis Space Search by
Find-S Instances X Hypotheses
H Specific General 1 x 2 x x 3 x 4 h 0 h 1 h 2,3 h 4 + + + x = , + 4 x = , + 1 x =,+2 x =, - 3 h = 1 h = 2 h = 4 h = 3 0 h = -.

For Micro Notes by the Student

For Micro Notes by the Student

ACE

Engineering College

(NBA ACCREDITED B.TECH COURSES: EEE, ECE & CSE)

Ankshapur (V), Ghatkesar, R.R.Dist. 501 301

Department of Computer Science Engineering

Action Plan -- Academic year 2018 – 2019

Name of the faculty	:	Mr.G.Sreenivasulu Associate Professor
Name of the Subject	:	**Machine Learning**
Class and Section	:	IV B.Tech CSE- B
Semester	:	I (ODD Semester)
No. of lectures per week	:	5+1 (Tutorial)

1	A) Define Concept Learning.	(2 Marks)
	B)What is the difference between Large Concept Space and TargetSpace?	(3 Marks)
	C)Sketch the Machine Learning Process Flow with a Diagram.	(10 Marks)
2	A)Define Machine Learning. List some instances of ML.	(2 Marks)
	B)Differentiate the different types of Learning.	(3 Marks)
	C)Generalize the Task of Concept Learning with an example.	(5 Marks)
	D)Explain the Version Space.	(5 Marks)
3	A)What is an Inductive Bias?	(2 Marks)
	B)What are the advantages and disadvantages of Machine Learning?	(3 Marks)
	C)Demonstrate the Candidate Elimination Algorithm in detail.	(10 Marks)
4	A)Why Machine Learning is so important?	(2 Marks)
	B)State are the two phases of Machine Learning? Describe.	(3 Marks)
	C)Illustrate the Inductive Learning Hypothesis with an example of 4 attributes, and 9 values of your choice.	(10 Marks)
5	A)State the Candidate Elimination Algorithm	(2 Marks)
	B)Write the algorithm for a desired choice of Hypothesis in Learning process.	(3 Marks)
	C)Explain in detail about the General-To-Specific Ordering with an example.	(10 Marks)
6	A)Differentiate “More-General-Or-Equal” & “S”	(2 Marks)
	B)What do you mean by Success Criteria in Machine Learning?	(3 Marks)
	C)What are the different issues in data sources? Explain in detail.	(5 Marks)
	D)What is a PAC Model? Explain in Detail.	(5 Marks)

For Micro Notes by the Student

1. A). Define Concept Learning.
B). What is the difference between Large Concept Space and Target Space?
C). Sketch the Machine Learning Process Flow with a Diagram.

A)

Concept learning, also known as **category learning, concept attainment**, and **concept formation**, is defined as "the search for and listing of attributes that can be used to distinguish exemplars from non-exemplars of various categories".

It is a Boolean-valued function from training examples of its input and output.A concept is an idea of something formed by combining all its features or attributes which construct the given concept. Every concept has two omponents:

1. Attributes: features that one must look for to decide whether a data instance is a positive one of the concept.
2. A rule: denotes what conjunction of constraints on the attributes will qualify as a positive instance of the concept.

B)

Concept learning can be viewed as the task of searching through a large space of hypotheses. Let X be the set of arbitrary collection of well-defined objects or instances. Then the set of all possible concepts that can be made is said to be as the "Large concept Space" whereas for any concept we consider. That is, for any no. of training examples data set, there exist a specific Set of Hypothesis consistent to the given training examples data sets. That we can call as the Target Space. To illustrate more about this Let we consider the following example,

Suppose we have 4 Attributes, in which is a Boolean valued, and the concept is stated as "Malignant Tumor".

Instance space X Sky (Sunny/Cloudy/Rainy) AirTemp (Warm/Cold) Humidity (Normal/High) Wind (Strong/Weak) Say for example:

For a given object I considered the following features:

Shape: Circular / Oval Size: Small / Large

Color: Dark / Lighter Surface : Smooth / Irregular

Then the no. of possible intendances or objects can be made from the above Boolean valued attributes are (2 x 2 x 2 x 2 = 16 = 2^4). Hence there will be 16 possible featured objects exists

Then the no. of ways you can be divide the above all objects in to two subsets possible concepts:

Then no. of all possible concepts can be X = 2^16 (65,536) : => This is the size of Large Concept Space.

i.e., If we have d binary features the total no. of all possible concepts can be 2^2^d.

In Conjunctive concept: (Hypothesis) we need to find the the possible alternatives for each attribute as __ ^ __ ^ __ ^ __

Now the possible How many conjunctive concepts can there be? (if you have 4 binary features) (Either first feature or Second or no feature) –

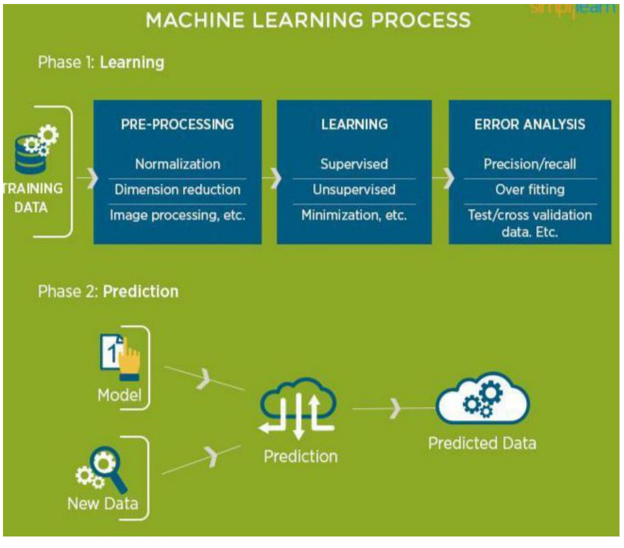
So 3 ^ d + 1 = 3^4 + 1 = 82 (Look the possible concepts which are 2^2^d): This is the size of Target space.

We made our problem much simpler..

C)

Machine Learning with a constant evolution of the field there has been subsequent increase in the model and development based on its demand and the influence of other fields like analytics and BigData. Machine learning has also changed the way data extraction, and interpretation is done by involving automatic sets of generic methods that have replaced traditional statistical techniques.

The process flow depicted here represents how machine learning works



Whatever the Data has to be process for Machine Learning, It is important to build a Model.

There are mainly two phases in ML. they are:

1. Learning
2. Prediction.

Under Learning, We have 3 phases are there.

1. Preprocessing
2. Learning Error
3. Analysis.

Preprocessing requires the Data Pre-Processing. This required Training Data. A Training data is used to train an algorithm. Generally, training data is a certain percentage of an overall dataset along with testing set. As a rule, the better the training data, the better the algorithm or classifier performs.

Once a model is trained on a training set, it's usually evaluated on a test set. Oftentimes, these sets are taken from the same overall dataset, though the training set should be labeled or enriched to increase an algorithm's confidence and accuracy.

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm.

Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

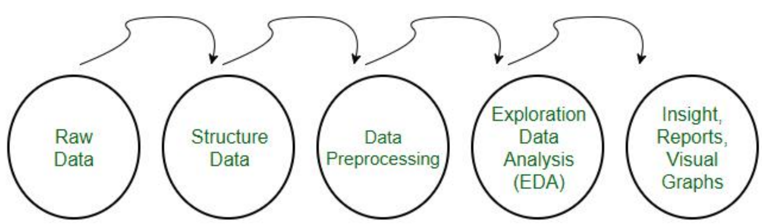


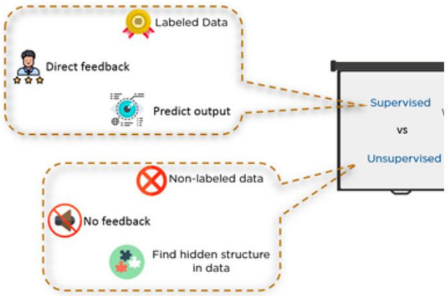
Fig: Data Preprocessing Steps

Learning :: Choosing a model

The next step in our workflow is choosing a model. There are many models that researchers and data scientists have created over the years. Some are very well suited for image data, others for sequences (like text, or music), some for numerical data and others for text-based data.

Learning Task can be accomplished in different ways like Supervised Learning, Unsupervised Learning and Inferential Learning.

For Micro Notes by the Student



Error Analysis and Tradeoffs

There are multiple types of errors associated with machine learning and predictive analytics. The primary types are in-sample and out-of-sample errors. In-sample errors (aka re-substitution errors) are the error rate found from the training data, i.e., the data used to build predictive models.

One very important point to note is that prediction performance and error analysis should only be done on test data, when evaluating a model for use on non-training or new data (out-of-sample).

Generally speaking, model performance on training data tends to be optimistic, and therefore data errors will be less than those involving test data. There are tradeoffs between the types of errors that a machine learning practitioner must consider and often choose to accept.

For binary classification problems, there are two primary types of errors. Type 1 errors (false positives) and Type 2 errors (false negatives). It's often possible through model selection and tuning to increase one while decreasing the other, and often one must choose which error type is more acceptable. This can be a major tradeoff consideration depending on the situation.

In the Phase 2: We have now ready with the model after making it ready for test the data and interpret the computations. The Model with consider the Test data and process it for the Prediction, It depend on the model for the output, in such a way that it can produce the prediction Data and results.

2.

A). Define Machine Learning. List some instances of ML.

B). Differentiate the different types of Learning.

C). Generalize the Task of Concept Learning with an example.

D). Explain the Version Space.

A)

Definition: Machine learning is a core sub-area of artificial intelligence enables computers to get the ability into a mode of self-learning and improve from experience without being explicitly programmed.
Machine learning focuses on the development of computer programs that can access data and use it learn for themselves when exposed to new data, these computer programs are enabled to learn, grow, change, and develop by them.
By building precise models, an organization has a better chance of identifying profitable opportunities – or avoiding unknown risks.
To better understand the uses of machine learning, consider some of the instances where machine learning is applied: the self-driving Google car, cyber fraud detection, online recommendation engines—like friend suggestions on Face book, Netflix showcasing..

B) Differentiate the different types of Learning. Supervised Learning

Supervised learning relies on data where the true class of the data is revealed. For example, if we want to teach the computer to distinguish between pictures of cats and dogs. We will run the algorithm on lots of pictures of cats and dogs. In order to supervise the algorithm to learn the right way to classify images, we will label the pictures as cats and dogs. Once our algorithm learns how to classify images we can use it on new data and predict labels (cat or dog in our case) on unseen images.

Unsupervised Learning :Unsupervised learning means that the learning algorithm does not have any labels attached to supervise the learning. We just provide algorithm with a large amount of data and characteristic of each observation. Imagine there are no labels to the images of cats and dogs in above example. In such case the algorithm itself cannot decide what a face is, but it can divide the data into groups. You can employ unsupervised learning (for e.g. clustering) to separate the images in two groups based on some inherent features of the pictures like color, size, shape etc.

Reinforcement Learning
Another well-known class of ML problems is called reinforcement learning.
This class of problems focus on the end outcome to learn. Let's illustrate by an example of learning to play chess. As input to this problem ML algorithm receives information about whether a game played was won or lost. So ML does not have every move in the game labelled as successful or not, but only has the result of the whole game. The more games the algorithm plays, the more it learns about the winning moves.
C)

For Micro Notes by the Student

Concept learning, also known as category learning, concept attainment, and concept formation, as “the search for and listing of attributes that can be used to distinguish exemplars from non-exemplars of various categories”.
A Concept is a subset of objects or events defined over a larger set [Example: The concept of a bird is the subset of all objects (i.e., the set of all things or all animals) that belong to the category of bird. Alternatively, a concept is a Boolean-valued function defined over this larger set. i.e., Acquire the Ability to categorize a well-defined object belonging to a concept (Category) or not.

Example: a function defined over all animals whose value is true for birds and false for every other animal]. Given a set of examples labeled as members or non-members of a concept, concept-learning consists of automatically inferring the general definition of this concept.
In other words, concept-learning consists of approximating a Boolean-valued function from training

examples of its input and output.
Example of a Concept Learning task:
Concept: Good Days for Water Sports (values: Yes, No)

Attributes/Features:
Sky (values: Sunny, Cloudy, Rainy)
AirTemp (values: Warm, Cold)
Humidity (values: Normal, High)
Wind (values: Strong, Weak)
Water (Warm, Cool)

Forecast (values: Same, Change)

- Example of a Training Point:
- <Sunny, Warm, High, Strong, Warm, Same, Yes> (Yes is a class label)

Concept Data Object
c x

x – Belong to a concept c 1 or +1 or T-- Not Belonging to concept c 0 or -1 or F
X = all possible concepts

Conjunction of constraints on each attribute where:
“?” means “any value is acceptable”
“0” means “no value is acceptable”
Finding Malignant Tumors from MRI SCANS:

For Micro Notes by the Student

Attributes:
Shape: Circular / Oval
Size: Small / Large
Color: Dark / Lighter
Surface : Smooth / Irregular
Thickness: Thin / Thick

Concept :
Malignant Tumor
Example of a hypothesis: <?, Cold, High,?,?,?>
(If the air temperature is cold and the humidity high then it is a good day for water sports)
Say for example:

For a given object I considered the following features:
Shape: Circular / Oval Size: Small / Large

Color: Dark / Lighter Surface : Smooth / Irregular
(2 x 2 x 2 x 2 = 16 = 2^4)

Hence there will be 16 possible featured objects exists
How many ways you can be divide the above all objects in to two subsets possible concepts:
Then no. of all possible concepts can be X = 2^16 (65,536)

i.e., If we have d binary features the total no. of all possible concepts can be 2^2^d
Inductive bias:
- First make an assumption that would like to target and reduce the concept Size is called Inductive bias
Reduction large concept space to small Target concept space. Conjunctive concept: (Hypothesis)
If x<- (Circular, Small, Dark, Smooth) then x1 < -(1, 0, 1, 0)
__ ^ __ ^ __ ^ __
? ^ ? ^ ? ^ ? – Accept All (16 possibilities) (phi) - Reject all - 1

Now the possible How many conjunctive concepts can there be? (if you have 4 binary features) (Either first feature or Second or no feature) –
So 3 ^ d + 1 = 3^4 + 1 = 82 (Look the possible concepts which are 2^2^d) We made our problem much simpler..
This is our Hypothesis space…saying that our target concept lies here.

For Micro Notes by the Student

Simple Algorithm:

1. Start with h = 0
2. Use Next input {x, c(x)}
3. If c(x) = 0, goto step 2
4. h<-h^x (pairwise-and **)
5. If more samples: Goto step 2
6. Stop   ** Pairwise rules has to be specified.

D)
The Candidate-Elimination algorithm represents the set of all hypotheses consistent with the observed training examples. • This subset of all hypotheses is called the version space with respect to the hypothesis space H and the training examples D, because it contains all plausible versions of the target concept.

VSHD≡{hϵH|Consistant(h,D)}

The set of items/objects over which the concept is defined is called the set of instances and denoted by X. The concept or function to be learned is called the target concept and denoted by c. It can be seen as a boolean valued function defined over X and can be represented as c: X -> {0, 1}.
If we have a set of training examples with specific features of target concept C, the problem faced by the learner is to estimate C that can be defined on training data.
H is used to denote the set of all possible hypotheses that the learner may consider regarding the identity of the target concept. The goal of a learner is to find a hypothesis H that can identify all the objects in X so that h(x) = c(x) for all x in X.
An algorithm that supports concept learning requires:
Training data (past experiences to train our models)
Target concept (hypothesis to identify data objects)
Actual data objects (for testing the models)
Inductive Learning Hypothesis
As we discussed earlier, the ultimate goal of concept learning is to identify a hypothesis H identical to target concept C over data set X with the only available information about C being its value over X. Our algorithm can guarantee that it best fits the training data. In other words:
"Any hypothesis found approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over other unobserved examples."

3. A). What is an Inductive Bias?
B). What are the advantages and disadvantages of Machine Learning?
C). Demonstrate the Candidate Elimination Algorithm in detail.

For Micro Notes by the Student

A)
Inductive bias: Inductive Bias is the set of assumptions a learner uses to predict results given inputs it has not yet encountered.
- First make an assumption that would like to target and reduce the concept size is called Inductive bias. i.e., Reduction large concept space to small Target concept space.

B)
Advantages:
- Primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.
- A form of artificial intelligence, automatically produce models that can analyze bigger, more complex data and deliver faster, more accurate results – even on a very large scale.
- By building precise models, an organization has a better chance of identifying profitable opportunities – or avoiding unknown risks.

To better understand the uses of machine learning, consider some of the instances where machine learning is applied: the self-driving Google car, cyber fraud detection, online recommendation engines— like friend suggestions on Face book, Netflix showcasing.

i. Machine learning has the major challenge called Acquisition. Also, based on different algorithms data need to be processed. And, it must be processed before providing as input to respective algorithms. Thus, it has a significant impact on results to be achieved or obtained.
ii. As we have one more term interpretation. That it results is also a major challenge. That need to determine the effectiveness of machine learning algorithms.

C)
Candidate Elimination Algorithm will

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 CANDIDATE-ELIMINATION algorithm computes the version space containing all hypotheses from

For Micro Notes by the Student

H that are consistent with an observed sequence of training examples. It begins by initializing the version space to the set of all hypotheses in H; that is, by initializing the G boundary set to contain the most general hypothesis in H

G0, G1, G2: {<?, ?, ?, ?, ?, ?>

and initializing the S boundary set to contain the most specific (least general) hypothesis S0 : < φ,φ,φ,φ,φ,φ>

These two boundary sets delimit the entire hypothesis space, because every other hypothesis in H is both more general than So and more specific than Go. As each training example is considered, the S and G boundary sets are generalized and specialized, respectively, to eliminate from the version space any hypotheses found inconsistent with the new training example. After all examples have been processed, the computed version space contains all the hypotheses consistent with these examples and only these hypotheses.

When the first training example is presented (a positive example in this case), the CANDIDATE-ELIMINATION algorithm checks the S boundary and finds that it is overly specific-it fails to cover the positive example. The boundary is therefore revised by moving it to the least more general hypothesis that

covers this new example. This revised boundary is shown as S1. No update of the G boundary is needed in response to this training example because

Go correctly covers this example. When the second training example (also positive) is observed, it has a similar effect of generalizing S further to S2, leaving G again unchanged (i.e., G2 = GI = GO). Notice the processing of these first Training examples:

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

S0 : < φ,φ,φ,φ,φ,φ>
S1 : {<Sunny, Warm, Normal, Strong, Warm, Same>
S2 : {<Sunny, Warm,?, Strong, Warm, Same>
G0, G1, G2: {<?, ?, ?, ?, ?, ?>

As illustrated by these first two steps, positive training examples may force the S boundary of the version space to become increasingly general. Negative training examples play the complimentary role of forcing the G boundary to become increasingly specific. Consider the third training example, shown in Figure. This negative example reveals that the G boundary of the version space is overly

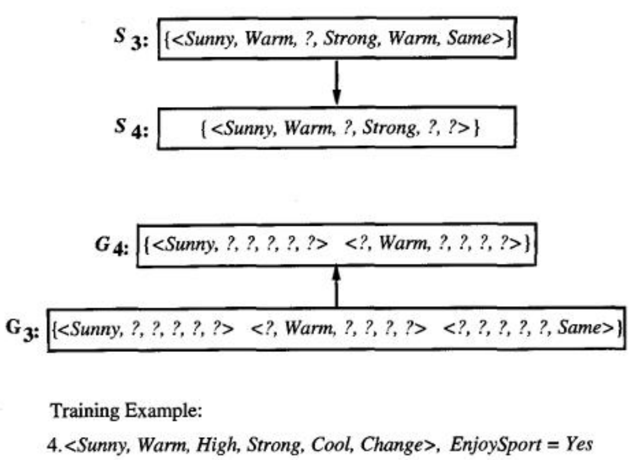
For Micro Notes by the Student

general; that is, the hypothesis in G incorrectly predicts that this new example is a positive example. The hypothesis in the G boundary must therefore be specialized until it correctly classifies this new negative example. As shown in Fig there are several alternative minimally more specific hypotheses. All of these become members of the new G3 boundary set. Given that there are six attributes that could be specified to specialize G2, why are there only three new hypotheses in G3?

For example, the hypothesis h = (?, ?, Normal, ?, ?, ?) is a minimal specialization of G2 that correctly labels the new example as a negative example, but it is not included in G0. The reason this hypothesis is excluded is that it is inconsistent with the previously encountered positive examples. The algorithm determines this simply by noting that h is not more general than the current specific boundary, S0. In fact, the S boundary of the version space forms a summary of the previously encountered positive examples that can be used to determine whether any given

hypothesis bounded by S4 and G4, is shown in Figure This learned version space is independent of the sequence in which the training examples are presented (because in the end it contains all hypotheses consistent with the set of examples).

As further training data is encountered, the S and G boundaries will move monotonically closer to each other, delimiting a smaller and smaller version space of candidate hypotheses



4.
A) Why Machine Learning is so important?
B)State are the two phases of Machine Learning? Describe.
C)Illustrate the Inductive Learning Hypothesis with an example of 4 attributes, and 9 values of your choice.

A)

For Micro Notes by the Student

Machine Learning in this era playing a vital that can make the model quickly, cheaply and automatically process and analyze huge volumes of complex data, machine learning is critical to countless new and future applications.
Machine learning is used to handle multi-dimensional and multi-variety data in dynamic environments. Machine learning allows time cycle reduction and efficient utilization of resources.
As machine learning has many wide applications. Such as banking and financial sector, healthcare, retail, publishing etc.

B)
Whatever the Data has to be process for Machine Learning, It is important to build a Model.
There are mainly two phases in ML.
They are 1. Learning 2. Prediction.
Under Learning, We have 3 phases are there.
Pre-Processing Learning
Error Analysis.
Preprocessing requires the Data Pre-Processing. This required Training Data. A Training data is used to train an algorithm. Generally, training data is a certain percentage of an overall dataset along with testing set. As a rule, the better the training data, the better the algorithm or classifier performs.
Once a model is trained on a training set, it's usually evaluated on a test set. Oftentimes, these sets are taken from the same overall dataset, though the training set should be labeled or enriched to increase an algorithm's confidence and accuracy.

C)
Concept learning, also known as category learning, concept attainment, and concept formation, as "the search for and listing of attributes that can be used to distinguish exemplars from non-exemplars of various categories".

Example of a Concept Learning task:
Concept: Good Days for Water Sports (values: Yes, No)

Attributes/Features:
Sky (values: Sunny, Cloudy, Rainy)
AirTemp (values: Warm, Cold)
Humidity (values: Normal, High)
Wind (values: Strong, Weak)
Water (Warm, Cool)

Forecast (values: Same, Change)

- Example of a Training Point:
- <Sunny, Warm, High, Strong, Warm, Same, Yes> (Yes is a class label)

Concept Data Object
c x

x – Belong to a concept c 1 or +1 or T
-- Not Belonging to concept c 0 or -1 or F

X = all possible concepts

Conjunction of constraints on each attribute where:
“?” means “any value is acceptable”
“0” means “no value is acceptable”

Concept :

Simple Algorithm:

1. Start with h = 0
2. Use Next input {x, c(x)}
3. If c(x) = 0, goto step 2
4. h<-h^x (pairwise-and **)
5. If more samples: Goto step 2
6. Stop .

Example of a hypothesis: <?, Cold, High,?,?,?>
(If the air temperature is cold and the humidity high then it is a good day for water sports)
(2 x 2 x 2 x 2 x 2 x 2 = 64 = 2^6)

Hence there will be 64 possible featured objects exists
How many ways you can be divide the above all objects in to two subsets possible concepts:
Then no. of all possible concepts can be X = 2^64
i.e., If we have d binary features the total no. of all possible concepts can be 2^2^d

Inductive bias:

- First make an assumption that would like to target and reduce the concept Size is called Inductive bias
Reduction large concept space to small Target concept space. Conjunctive concept: (Hypothesis)
If x<- (Circular, Small, Dark, Smooth) then x1 < -(1, 0, 1, 0, 0, 1)
__ ^ __ ^ __ ^ __
? ^ ? ^ ? ^ ? – Accept All (64 possibilities) (phi) - Reject all - 1

Now the possible How many conjunctive concepts can there be? (if you have 4 binary features) (Either first feature or Second or no feature) –
So 3 ^ d + 1 = 3^6 + 1 (Look the possible concepts which are 2^2^d) We made our problem much simpler..
This is our Hypothesis space…saying that our target concept lies here.
5.
A) State the Candidate Elimination Algorithm
B) Write the algorithm for a desired choice of Hypothesis in Learning process.
C) Explain in detail about the General-To-Specific Ordering with an example.

A)
The candidate-Elimination algorithm computes the version space containing all (and only those) hypotheses from H that are consistent with an observed sequence of training examples.
The candidate elimination algorithm incrementally builds the version space given a hypothesis space H and a set E of examples. The examples are added one by one; each example possibly shrinks the version space by removing the hypotheses that are inconsistent with the example. The candidate elimination algorithm does this by updating the general and specific boundary for each new example

B)
Simple Algorithm Inductive biasing for desired choice of Hypothesis
1. Start with h = 0
2. Use Next input {x, c(x)}
3. If c(x) = 0, goto step 2
4. h<-h^x (pairwise-and **)
5. If more samples: Goto step 2
6. Stop ** Pairwise rules has to be specified.

C)

By taking advantage of naturally occurring structure, we can design learning algorithms that exhaustively search even infinite hypothesis spaces without explicitly enumerating every hypothesis.
For instance, general-to-specific ordering
Any instance classified positive by will be classified positive by , therefore is more general than . Let and be boolean-valued functions defined over X. Then is more-general-than-or-equal-to if and
only if .

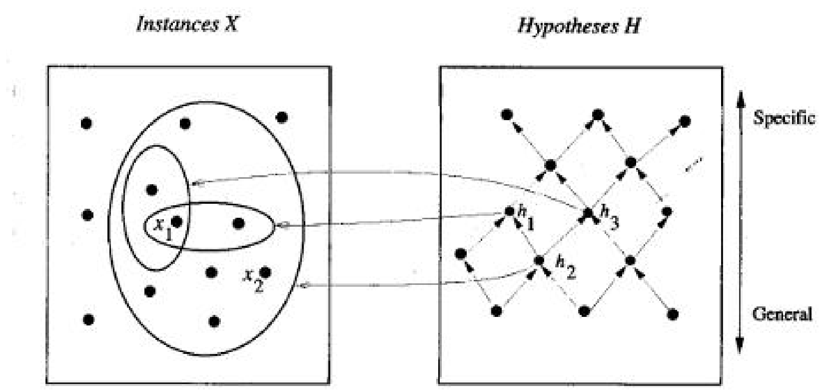
More-general-than and more-specific-than are also useful.

Many algorithms for concept learning organize the search through the hypothesis space by relying on a very useful structure that exists for any concept learning problem: a general-to-specific ordering of hypotheses. By taking advantage of this naturally occurring structure over the hypothesis space, we can design learning algorithms that exhaustively search even infinite hypothesis spaces without explicitly enumerating every hypothesis. To illustrate the general-to-specific ordering, consider the two hypotheses h1 = (Sunny, ?, ?, Strong, ?, ?)

h2 = (Sunny, ?, ?, ?, ?, ?)

Now consider the sets of instances that are classified positive by hl and by h2. Because h2 imposes fewer constraints on the instance, it classifies more instances as positive. In fact, any instance classified positive by hl will also be classified positive by h2. Therefore, we say that h2 is more general than hl. This intuitive "more general than" relationship between hypotheses can be defined more precisely as follows. First, for any instance x in X and hypothesis h in H, we say that x satisfies h if and only if h(x) = 1. We now define the more-general~han_or.-equal~o relation in terms of the sets of instances that satisfy the two hypotheses: Given hypotheses hj and hk, hj is more-general-thanm-equal do hk if and only if any instance that satisfies hk also satisfies hi.

To illustrate these definitions, consider the three hypotheses hl, h2, and h3 from our Enjoy sport example, shown in Figure.



= <Sunny, Warm, High, Strong, Cool, Same>
= <Sunny, Warm, High, Light, Warm, Same>

h_1= <Sunny, ?, ?, Strong, ?, ?>
h_2= <Sunny, ?, ?, ?, ?, ?>
h_3= <Sunny, ?, ?, ?, Cool, ?>

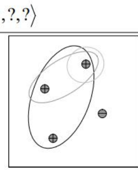
How are these three hypotheses related by the p, relation? As noted earlier, hypothesis h2 is more general than hl because every instance that satisfies hl also satisfies h2. Similarly, h2 is more general than h3. Note that neither hl nor h3 is more general than the other; although the instances satisfied by these two hypotheses intersect, neither set subsumes the other. Notice also that the p, and >, relations are defined independent of the target concept. They depend only on which instances satisfy the two hypotheses and not on the classification of those instances according to the target concept. Formally, the p, relation defines a partial

For Micro Notes by the Student

order over the hypothesis space H (the relation is reflexive, antisymmetric, and transitive).
Because h2 imposes fewer constraints, it will cover more instances x in X than both h1 and h3. h2 is more general than h1 and h3.

In a similar manner, we can define more-specific-than.
Find-S Algorithm:
1. Initialize h to the most specific hypothesis in H
2. For each positive training instance x
    For each attribute constraint ai in h
        If the constraint ai in h is satisfied by x Then do nothing
        Else
        replace ai in h by the next more general constraint that is satisfied by x
3. Output hypothesis have Many algorithms for concept learning organize the search by using a useful structure in the hypothesis space: general-to-specific ordering!
The following example will illustrate more about this.

$$h \leftarrow \langle \varnothing, \varnothing, \varnothing, \varnothing, \varnothing, \varnothing \rangle$$
$$x = \langle Sunny, Warm, Normal, Strong, Warm, Same \rangle$$
$$h \leftarrow \langle Sunny, Warm, Normal, Strong, Warm, Same \rangle$$
$$x = \langle Sunny, Warm, High, Strong, Warm, Same \rangle$$
$$h \leftarrow \langle Sunny, Warm, ?, Strong, Warm, Same \rangle$$
$$x = \langle Rainy, Cold, High, Strong, Warm, Change \rangle$$
$$h \leftarrow \langle Sunny, Warm, ?, Strong, Warm, Same \rangle$$
$$x = \langle Sunny, Warm, High, Strong, Cool, Change \rangle$$
$$h \leftarrow \langle Sunny, Warm, ?, Strong, ?, ? \rangle$$



6.
A) Differentiate “More-General-Or-Equal” & “S”
B) What do you mean by Success Criteria in Machine Learning?
C) What are the different issues in data sources? Explain in detail.
D) What is a PAC Model? Explain in Detail.

A) Many algorithms for concept learning organize the search through the hypothesis space by relying on a very useful structure that exists for any concept learning problem: a general-to-specific ordering of hypotheses. By taking advantage of this naturally occurring structure over the hypothesis space, we can design learning algorithms that exhaustively search even infinite hypothesis spaces without explicitly enumerating every hypothesis. To illustrate the general-to-specific ordering, consider the two hypotheses h1 = (Sunny, ?, ?, Strong, ?, ?)

h2 = (Sunny, ?, ?, ?, ?, ?)

h2 is said to be more-genral-Or-Equal-than h1.

Coming ot Find S:

The first step is to define the procedure for the Find-S for just one class. So that we can later apply it to each class. The whole algorithm can be reduced to a simpler one: instead of iterating and replacing one value at a time we can just look at the whole column vector of the attirbute and then define the resulting hypothesis as follows:

set value in position "i" of the hypothesis to value v if the column i of the dataset has only this value v
set value in position "i" to "?" otherwise

B) Consolidated a brief overview of some of the most important quality criteria on quality assurance for machine learning.

Quality criteria for machine learning

1. Accuracy: The most important quality characteristic of a machine learning algorithm is the accuracy of the category mapping or prediction. The accuracy that can be achieved depending on the specific problem, the model, as well as the type and quantity of input data.

2. Robustness: Another important quality trait is the robustness with respect to varying inputs. You must try to avoid two opposing problems: bias and over fitting.

3. Bias (or under-fitting) occurs when a machine learning algorithm bases its decisions on too few input parameters. Additional information from other inputs is disregarded. There are some prominent examples of such bias in recent years.

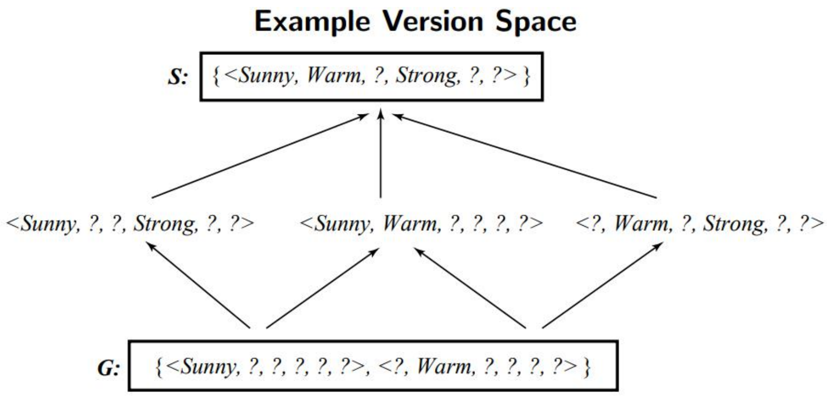
4. Over fitting (also over training) occurs when irrelevant or insignificant characteristics are included in the decision-making process.

5. Performance: The requirements in memory and computing capacity to complete pre-processing and training. Last but not least the time to calculate the actual result for each new input must meet requirements.

c)

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 hx,c(x)i in D. Consistent(h, D) ≡ (∀hx, c(x)i ∈ D) h(x) = c(x) Definition The version space, V SH,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. V SH,D ≡ {h ∈ H|Consistent(h, D)}

For Micro Notes by the Student



Representing Version Spaces

1. The General boundary, G, of version space V SH,D is the set of its maximally general members that are consistent with the given training set
2. The Specific boundary, S, of version space V SH,D is the set of its maximally specific members that are consistent with the given training set
3. Every member of the version space lies between these boundaries V SH,D = { h ∈ H|(∃ s ∈ S)(∃ g ∈ G)(g ≥ h ≥ s) } where x ≥ y means x is more general or equal to y

D)probably approximately correct (PAC):

We focus here on the problem of inductively learning an unknown target function, given only training examples of this target function and a space of candidate hypotheses. Within this setting, we will be chiefly concerned with questions such as how many training examples are sufficient to successfully learn the target function, and how many mistakes will the learner make before succeeding. As we shall see, it is possible to set quantitative bounds on these measures, depending on attributes of the learning problem such as:

The size or complexity of the hypothesis space considered by the learner The accuracy to which the target concept must be approximated

The probability that the learner will output a successful hypothesis The manner in which training examples are presented to the learner
For the most part, we will focus not on individual learning algorithms, but
Rather on broad classes of learning algorithms characterized by the hypothesis
Spaces they consider, the presentation of training examples, etc.
Our goal is to answer questions such as
Sample complexity. How many training examples are needed for a learner to converge (with high probability) to a successful hypothesis?

Computational complexity. How much computational effort is needed for a learner to converge (with high probability) to a successful hypothesis?

Mistake bound. How many training examples will the learner misclassify before converging to a successful hypothesis?

Note there are many specific settings in which we could pursue such questions. For example, there are various ways to specify what it means for the learner to be “successful.” We might specify that to succeed, the learner must output a hypothesis identical to the target concept. Alternatively, we might simply require that it output a hypothesis that agrees with the target concept most of the time, or that it usually output such a hypothesis. Similarly, we must specify how training examples are to be obtained by the learner. We might specify that training examples are presented by a helpful teacher, or obtained by the learner performing experiments, or simply generated at random according to some process outside the learner’s control. As we might expect, the answers to the above questions depend on the particular setting, or learning model, we have in mind.

The Problem Setting As in earlier chapters, let X refer to the set of all possible instances over which target functions may be defined. For example, X might represent the set of all people, each described by the attributes age (e.g., young or old) and height (short or tall).

Let C refer to some set of target concepts that our learner might be called upon to learn. Each target concept c in C corresponds to some subset of X, or equivalently to some boolean-valued function c : X + {0, 1). For example, one target concept c in C might be the concept “people who are skiers.” If x is a positive example of c, then we will write c(x) = 1; if x is a negative example, c(x) = 0.

We assume instances are generated at random from X according to some probability distribution D. For example, 2) might be the distribution of instances generated by observing people who walk out of the largest sports store in Switzer-land.

In general, D may be any distribution, and it will not generally be known to the learner. All that we require of D is that it be stationary; that is, that the distribution not change over time. Training examples are generated by drawing an instance x at random according to D, then presenting x along with its target value, c(x), to the learner.

The learner L considers some set H of possible hypotheses when attempting to learn the target concept. For example, H might be the set of all hypotheses describable by conjunctions of the attributes age and height.

For Micro Notes by the Student

After observing a sequence of training examples of the target concept c, L must output some hypothesis h from H, which is its estimate of c. To be fair, we evaluate the success of L by the performance of h over new instances drawn randomly from X according to D, the same probability distribution used to generate the training data.

Within this setting, we are interested in characterizing the performance of various learners L using various hypothesis spaces H, when learning individual target concepts drawn from various classes C. Because we demand that L be general enough to learn any target concept from C regardless of the distribution of training examples, we will often be interested in worst-case analyses over all possible target concepts from C and all possible instance distributions D.

The true error (denoted errorv(h)) of hypothesis h with respect to target concept c and distribution D is the probability that h will misclassify an instance drawn at random according to D.

Here the notation Pr indicates that the probability is taken over the instance x€D distribution V.

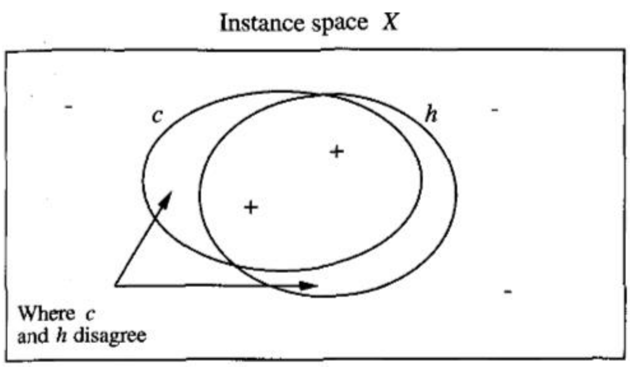


Figure shows this definition of error in graphical form. The concepts c and h are depicted by the sets of instances within X that they label as positive. The error of h with respect to c is the probability that a randomly drawn instance will fall into the region where h and c disagree (i.e., their set difference). Note we have chosen to define error over the entire distribution of instances-not simply over the training examples-because this is the true error we expect to encounter when actually using the learned hypothesis h on subsequent instances drawn from D. Note that error depends strongly on the unknown probability distribution that addresses this important special case.