Well Proposed Learning Problems

ML means

Model

without Human Intervention, by Experience model it should learn

In machine learning what is learning problem

Definition:- A Computer Program is said to learn from Experience (E) w.r.t some Class of tasks (T) and performance Measure (P), if its performance at tasks in T, as Measured by P, improved with Experience (E).

For Ex * One Scenary is drawing

1 - bird, Tree, Sun, Mountain, River

* My Main Target is to draw Scenary Picture

Small Task what is that?

1) Tree should know How to draw
2) Sun " " " " "
3) Bird.

These are Class of Task (T)

* With out mistake drawing That is called (P)

* from that finally what we know, i.e Not drawn Properly is called Experience (E)

Next Time will draw Perfectly

* By Task what we learn is Called (E), to develope Computer Program.

* First Problem identified - Next Creating model

* How learning Problem Correct means due to 3 features.

Features for a well-defined learning Problem:-

1) Class of Tasks
2) measure of performance to be improved
3) Source of Experience.

for Ex Checkers learning Problem:- What is T, P, E will see

Task T : Playing checkers → we perform Check (i.e) T

Performance Measure P : % of games won against opponents

[If Not win With Highest Rank means developed model is Not good]

Tracking Experience E : Playing Practice games against itself.
[with more Example, Training and by Experience ↑ Performance].

A Handwriting Recognition learning Problem:-

Handwriting Classify is Task

Task T : Recognizing and classifying handwritten words with in Images.

Performance Measure P: % of words Correctly classified

Training Experience E : A database of handwritten words with given Classifications

[Too much database, Too much Handwritten words with proper Classify Can ↑ Performance].

A Robot driving learning Problem:- Drive - Task.

Task T: Driving on public four-lane highway using vision sensor's

Performance Measure P : Avg distance Travelled before an Error

E : A sequence of images and steering commands recorded while observing a human driver. [Human - Command Registered User ↑ (Ex RoBo)]

Find T, P, E.

* Ludo Learning Algorithm
* Matching Pictures in the gallery of an User's Mobile Phone

Why Machine learning?

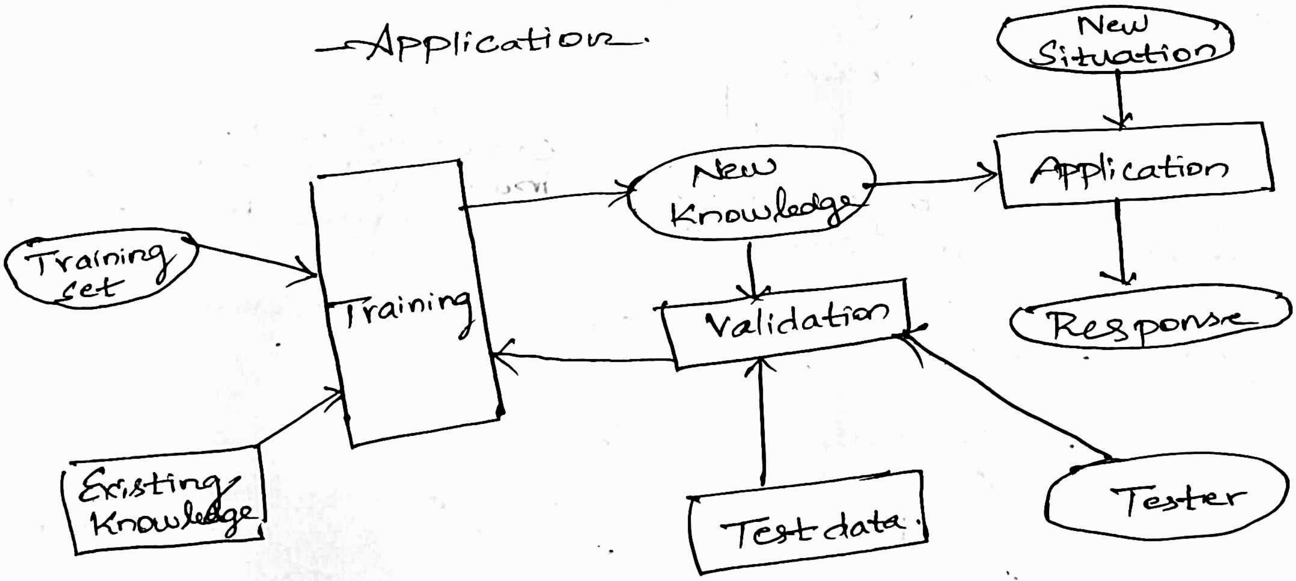
- To learn and build learning Machines (model develop)
- Pgming Computers to optimize a performance Criterion using Example data / past Experience

With D, E → ↑P → pgm develop

- learning Algorithms ← - - - - - - - - - - - - Various Alg
(Step By Step known, ↑P and getting o/p)

How machines learn?

- Training (While Robo develop giving Training)
- Validation (Rotate heads, Arms)
- Application.



Phases of ML.

—×—.

Designing a learning System:-

"To Avoid learning problem How learning System Designing"

4-steps are there.

- Choosing the training Experience
- " " Target fn.
- " " Representation for the Target fn.
- " " function Approximation Algorithm.
- The final Design

1. Choosing the Training Experience:

Training Experience < Success / failure.

3 attributes for TE

1. Types of feedback
2. Learning strategy
3. Diversity of Training.

1. TOF

→ Direct - Every board State and Correct move for each. (A – B Correct, A – C incorrect) success can easily identity like directly informed

→ Indirect - move Sequences and final outcome

- Problem of Credit Assignment.

Sequence-1 | A - B - C - D - E }
Sequence-2 | A - F - G - H - I } Win / loss directly Sucess Not possible

Learning Strategy:-

→ Relies on teacher - Expert guidance (A - B Success, A - C Non-Success w.r.t by Teacher).

→ No Teacher - Autonomous. (we Ourself want to identify. after game over will Come to know we did Some mistake due to that Not won)

Diversity of Training:-

- E might not be similar for every Situation
- Large Training data.

[Using large Time of facing Problems we will got Experience and Solve the problem easily].

"This is how Choosing TE".

A Checkers Learning Problem:-

T, P, E already known.

To Complete the design of learning System, We must Choose

1) The Exact Type of knowledge to be learned.
2) Representation for this target knowledge
3) Which Learning Mechanism Required to design.

2. Choosing the Target function:-

Choosing the best move from the class of legal moves: the Program must learn.

Function - ChooseMove

Notation - $\boxed{\text{ChooseMove} : B \rightarrow M}$

where, B - Set of legal Board states

M - Set of legal moves.

Choose move is difficult to learn when it gives indirect feedback.

Alternate Target fn:-

Assigning Numerical value in to the State

Target fn : V

Notation : $\boxed{V : B \rightarrow \alpha}$

Where
B — Set of legal board state
α — Set of real Numbers

The Target value of $V(b)$ for an arbitary board states b in B, is defined below

① if b is a final board state that is won,

then $V(b) = 100$

② lost,

then $V(b) = -100$

③ drawn,

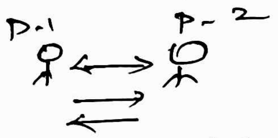
then $V(b) = 0$

④ If b is not a final state in the game

$V(b) = V(b')$

↓

Best final board state Starting from b assuming both Players play optimally.



[In 4 Situation value is Not possible to Compute]

[1 - 3] ✓ Value can Compute

$\boxed{\text{[4] × Value Can't Compute}}$,,

⇓

Called Non Operational definition

To Compute 4^{th} State We need to Convert

↓

Operational Description

↓

With learning Algorithm - Approx Target fn (Solving).

⇓

Solution (i.e) for approximation.

3. Choosing a Representation for Target fn :-

How we are representing means "$\hat{V}$"

V - ideal Target fn

$\hat{V}$ - my pgm How much it learnt

To Represent $\hat{V}$:- 1) Large table with values like won, loss.

2) Artificial Neural N/w

3) Quadrature PolyNomial fn

4) Collection of Rules that match.

When Training data ↑ α $\hat{V}$

↓

[If I need $\hat{V}$ clearly we should ↑ Training data]

A Simple Representation :-

x_1 → The No of Block Pieces on the Board

x_2 → " " Red " " " "

x_3 → " " Black Kings " "

x_4 → " " red Kings " "

x_5 → Black pieces threatened by red

x_6 → " " Red " " " Black

for that simple Error (Back page) Representing $\hat{V}(b)$ as a linear fn. to Calculate (i.e)

$$\boxed{\hat{V}(b) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 + w_6 x_6}$$

w_0 to w_6 → weight chosen from learning algorithm.

2min Partial Design:-

Task T : Playing Checkers

Performance P : % of games won

E : games played against itself

→ Specification of learning Task

Target fn V : Board → α

Target fn Representation :

→ implementation

$$\hat{V}(b) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 + w_6 x_6$$

4. Choosing a fn Approximation Algorithm:

To learn Target fn V we required,

b – a specific board state (like won, loss).

Training values also required

$V_{train}(b)$ – The training value for b

Training Example : $(b, V_{train}(b))$

(State, Train values)

Ex Black won the game (Note $x_2 = 0$) means No Red Coins.

$((x_1 = 3, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 0, x_6 = 0)$ ~~+~~ $100)$ → won (so 100)

To choose algorithm two things required.

1) Estimating Training Values

Easy :- Assign a value to a board state
[End of the game] we can easily Assign

Difficult :- Assign a value to intermediate Board state.

Rule for Estimating training values

$$V_{train}(b) \leftarrow V'(\text{Successor}(b)) \longrightarrow (1)$$

$\downarrow$

Next Board State

Next state also dont know, How we come to know,

When we realized next game chances to win. (If nearer to 100)

2) Adjusting the weight's.

— Choose the weights w_i

✓ Depends weight o/p will good, when means No Error

The fn we using is

$$E = \sum \Big(\underbrace{V_{train}(b)}_{\text{Our's}} - \underbrace{\hat{V}(b)}_{\text{Original}} \Big)^2$$

When original - Ours automatically Error $\downarrow$

How weight assigning means, with out Error we need to assign

(i.e) Least mean Square algorithm.

for each weight w_i, update it

New value ↓ ; Old value ↓

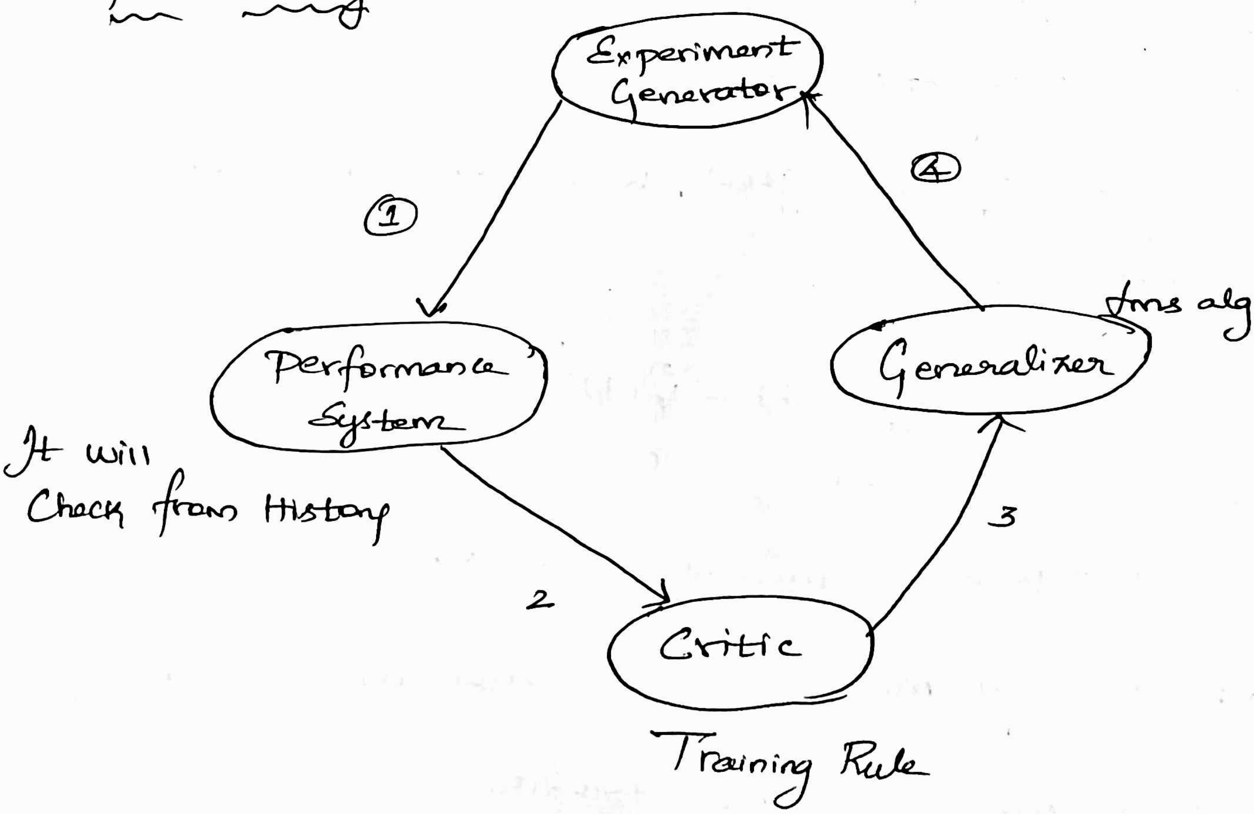
$$\boxed{w_i \leftarrow w_i + \underbrace{\eta}_{\text{Constant}} \underbrace{(V_{train}(b) - \hat{V}(b))}_{\text{Error.}} x_i}$$

When error = zero, No weights are changed.
$(w_i \leftarrow w_i + 0)$.

error = +ve, $w_i \leftarrow w_i +$ [Some small value]
↓
Weight will Increase

When $x_i = 0$, $w_i \leftarrow w_i + \eta(\text{Error})$
$= (w_i \leftarrow w_i + 0)$.
No weights are changed.

5. final Design



— x —

Perspective and issue in ML:-

Perspectives:- [(View) about ML]

* ML Involves Searching a Very large Space of Possible hypotheses to determine one that best fits the Observed data and any Prior knowledge held by learners.

- for Ex:- In Checkers learner, the hypothesis is located by searching through the vast Space with available training Ex.

Eg:-
Dress ← viewed in Amazon
* f.B suddenly link will come
* User data info, Observed got and Experienced knowledge with ↑ and finally Shown in facebook.
This is Called Perspective of ML

Hypothesis:-
After theorem Solve, we will get one Solution, By using that Solu we can do large Number of Problems.

Train Ex : $(V(b), V(b'))$
loss, win state
To calculate $V(b)$ we using Target fn

- LMS alg is used to ~~tune~~ the weights, each time the hypothesized evaluation fn predicts a Value that differs from the training Value.

LMS ⇒ weight assume
$V(b)$ ⇒ Representation
$w_0 + w_1x_1 + w_2x_2 \ldots$
Our weight ← Old weight will get
New weight will get

- Hypothesis Representation are appropriate for learning different kinds of Target fn.

"To get best o/p With machine So that we Bringing"

Issues:-

* What learning algorithm to be used.?

* How much learning data is sufficient?

* When and how prior knowledge can guide the learning Process?

* What is the best strategy for choosing a next training Experience?

* What is the best way to reduce the learning task to One or more fn approximation Problems?

* How can the learner automatically alter its representation to improve its learning ability?

—x—

Concept Learning Task:-

Introduction:-

learning:- learning Topic → Books ⇒ from different books Particular topics are understand

* Already Predefined Space are there, so many hypothesis available

Hypothesis means ↓

* Story reading ⇒ "Moral of finally the Story"

* My problem we giving only one line for solution, By using we will solve

* many hypothesis are there, which problem, which hypothesis required. That is Concept learning.

* Learning involves acquiring general Concepts from specific training Ex.

Ex Ref. - A problem of Searching through a Predefined Space of Potential hypothesis for the hypothesis that best fits the training Ex.

How identifying means
- The problem of automatically inferring the general definition of Some Concept.

* Infering means We are thinking that from many things what we Concluded.

Definition :- Inferring a boolean - valued fn from training Examples of its i/p and o/p.

* Boolean valued fn means - 0 or 1, Yes or No from my Ex, if i use Boolean fn automatically generated means, (i.e) Concept learning.

* A Concept Can be viewed as describing Some Subset of Objects or events defined Over a larger set.

- A Concept Can be thought of as a boolean valued fn defined Over the

Ex learning the Concept for "Car" [the Subset of Vehicles that Constitute Car] Can be Considered as a boolean fn defined Over all Vehicles. Whose Value "true for car and false for other vehicle".

2. Concept learning Task:

Concept → Days on which my friend enjoys his favourite water spot.

Example	Sky	AirTemp	Humidity	Wind	Water	forecast	Enjoy Spot
1	Sunny	Warm	Normal	Strong	Warm	Same	yes
2	Sunny	Warm	High	"	"	"	yes
3	Rainy	Cold	High	"	"	Change	No
4	Sunny	Warm	High	"	Cool	"	yes

* Six attributes are given to learn about the Concept
 - enjoy spot = yes

3 (+)ve 1 - (-)ve.

* How in representation with hypothesis means
 1) ? any value is acceptable (eg) Sunny / Rainy

But we can't directly give Sky, it won't accept.

< ?, cold, high, ?, ?, ? >

Hypothesis enjoys his favourite spot Cold day with high humidity

* Two hypothesis are there
 1) general hypothesis (eg) Tea
 < ?, ?, ?, ?, ?, ? > ? (any value).
 2) Specific hypothesis (eg) Ginger, elachi Tea.
 < ϕ, ϕ, ϕ, ϕ, ϕ, ϕ > ϕ - No value

Notation

1) Instance - a Specific Situation
(x / X) ~~what~~

2) Hypothesis - a proposed Statement for Every instance.
(h/H) (Coming to Conclusion)

3) Target Concept - for Concept generating for

4) Training Example - By ~~various~~ attribute getting experience

—x—

{Concept Learning as a Search:-

"CL main goal to find hypothesis. for our Ex we have to find a Suitable hypothesis."

* The goal of this Search is to find the hypothesis that best fit the training Ex.

* The Selection of hypothesis representation Consist the designer of the learning algorithm implicity Defines the Space of all hypothesis that the Pgm can learn.

Ex Consider the instances X and hypothesis in the Enjoy spot learning task.

* Attribute can have the possible values.

"Take Tabulation from previous Topic":

		No of Attributes
Sky -	Sunny / Cloudy / Rainy	3
Air temp -	Warm / Cold	2
Humidity -	Normal / High	2
Wind -	Warm / Cool	2
Water -	Same / Change.	2

The Number of distinct instances, present in the instances space $X = 3 \times 2 \times 2 \times 2 \times 2$

$= 96.$

Syntactically Distinct hypothesis :- [W.K.T, In C, C++ Syntax Available]

		No. of Attributes.
Sky	- Sunny / Cloudy / Rainy / ? / ϕ	5
Air Temp	- Warm / Cold / ? / ϕ	4
Humidity	- Normal / High / ? / ϕ	4
Wind	- Strong / Week / ? / ϕ	4
Water	- Warm / Cool / ? / ϕ	4
forecast	- Same / change / ? / ϕ	4

No of Syntactically distinct hypothesis With in

$H = 5 \times 4 \times 4 \times 4 \times 4 \times 4 = 5120.$

Semantically distinct hypothesis :

Every hypothesis Containing One or more "ϕ" Symbols represents the empty set of instances (i.e Negative)

∴ No of Semantically distinct hypothesis With in

$H = \underbrace{1}_{\text{Specific Type}} + (4 \times 3 \times 3 \times 3 \times 3 \times 3) = 973$ (No value ϕ Removed in above)

General to Specific Ordering of hypothesis:-

$h_1 = \langle$ Sunny, ?, ?, Strong, ?, ? $\rangle$ 2 Att

$h_2 = \langle$ Sunny, ?, ?, ?, ?, ? $\rangle$ 1 Att.

The two hypothesis h_1 and h_2 are evaluated and that Shows h_2 imposes fewer Constraints on the instance.

$\boxed{\therefore\ h_2 \text{ is more general than } h_1}$

"Whatever in h_1 instance Solve, h_2 Instances also will Solve"

for Any instances Classified Positive by h_1 will also be Classified (+)ve by h_2.

W.K.T Boolean fn (+)ve means 1.

So that $\boxed{h(x) = 1.}$

Definition:- Let $\overset{h_2}{h_j}$ and $\overset{h_1}{h_K}$ be boolean-valued fn defined Over X. Then $\overset{h_2}{h_j}$ is more general than or equal to $\overset{h_1}{h_K}$.

(written $\overset{h_2 \geq h_1}{h_j \geq h_K}$) if and only if.

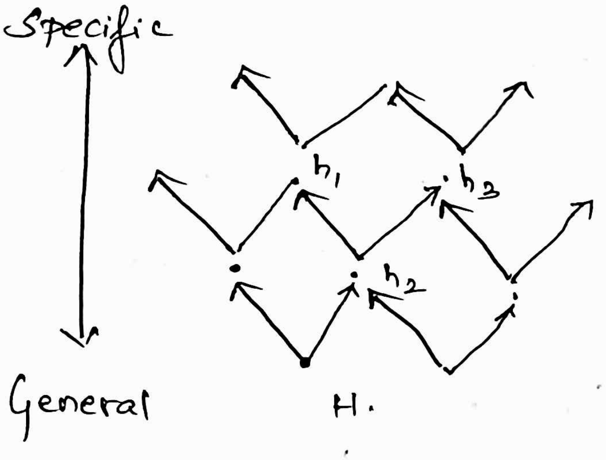
$$\boxed{(\forall_x \in X)\left[(h_K(x) = 1) \longrightarrow (h_2(x) = 1)\right]}$$

Overall x instances Set of instances. $h_1(x) = 1 \longrightarrow h_2(x) = 1$

* h_j is strictly more-general than h_K (written $h_j > h_K$) if and only if. $(h_j \geq_g h_K) \wedge (h_K \ngeq_g h_j)$

General hypothesis Should be greater than Specific hypothesis

(∧) and h_1 Should not greater than h_2



		N of att
h_1	= < Sunny, ?, ?, Strong, ?, ? >	2
h_2	= < Sunny, ?, ?, ?, ?, ? >	1
h_3	= < Sunny, ?, ?, ?, Cool, ? >	2.

* h_2 is more general than h_1 and h_3. Because h_2 Contains less Constraints.

* h_1, h_2, h_3 are Related, then only we will Select a Suitable (Specific) hypothesis from General hypothesis.

— x —

FIND S : find a maximally Specific hypothesis :-

S means Maximally Specific hypothesis

W.K.T - General Hypothesis ⇓ Value Replaced with ?

Specific hypothesis ⇓ Value Replaced with ϕ

for our Example :-
Now we are going to find S.

Algorithm :- 1) Initialize h to the most Specific Hypothesis in H (all value is ϕ)

2) for Each (+)ve training instance x

* for each attribute Constraint a_i in h

Att = hypothesis if not ↓ next page.

If the constraint a_i is Satisfied by x

Then do nothing ⟶ hypothesis value given to next general constraint

Else Replace a_i in h by the next more general constraint

* depends Example Instances ??? ? add.

* only take (+)ve instance

* (-)ve Instance it don't take

3) o/p hypothesis h.

Example:-

Task - Enjoy Spot.

Check.

Tabulation

Sky	Air...		

Step ①:-

h - most specific hypothesis

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

Step ②:- [Checking with (+)ve instances x]

Replace with the next general Constraint.

$h_1 \leftarrow$ < Sunny, Warm, Normal, Strong, Warm, Same >

Step ③:- [Checking Next (+)ve instances].

Since the value of third attribute Changes from "Normal" to "high". Replace with the Next general Constraint [?].

$h_2 \leftarrow$ < Sunny, Warm, ?, Strong, Warm, Same >

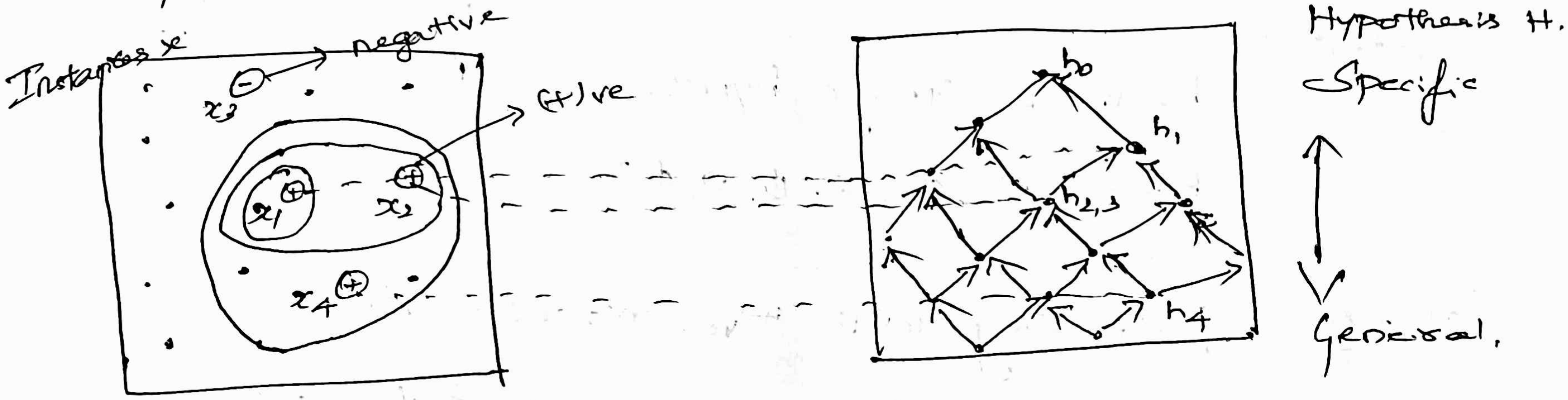
Step ④:- The algorithm makes No Change to h, because it is a (-)ve Example.

Step ⑤:- [Checking with Next (+)ve instances]

Since the value of 5^{th} attribute changes from "Warm" to "Cool" and 6^{th} attribute changes from "Same" to "Change" replace with ?

$$h_4 \leftarrow \langle Sunny, Warm, ?, Strong, ?, ? \rangle$$

Space Search:



x_1 = <Sunny, Warm, Normal, strong, warm, same>, +

x_2 = < " > +

x_3 < " > −

x_4 = < " > +

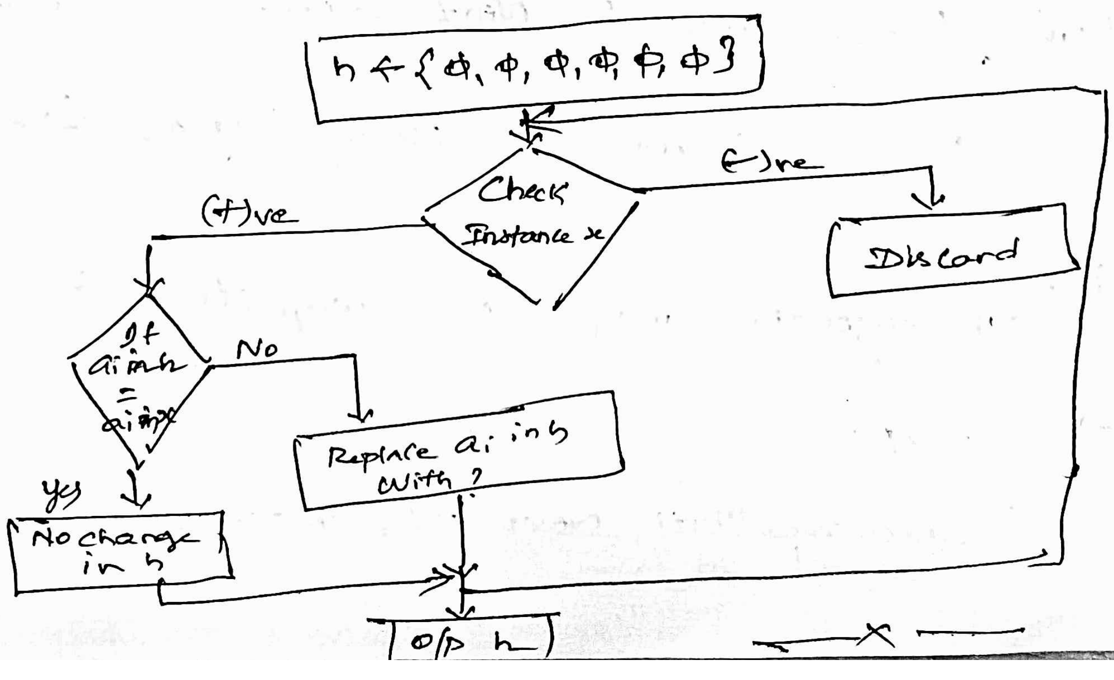
$h_0 = \langle \phi, \phi, \phi, \phi, \phi, \phi \rangle$

$h_1 = \langle \quad \rangle$

$h_2 = \langle \quad \rangle$

$h_3 = \langle$ same as h_2 $\rangle$

$h_4 = \langle \quad \rangle$



Candidate Elimination Algorithm

Study Hour

Time	Weather	Temp	Company	Humidity	Wind	Goes.
Morning	Sunny	Warm	yes	Mild	Strong	yes
Evening	Rainy	Cold	No	Mild	Normal	No
Morning	Sunny	Moderate	yes	Normal	Normal	yes
Evening	Sunny	Cold	yes	High	Strong	yes

yes – like to go walk   No – No walk.

- Extended form of find-S- Algorithm
- Consider both +ve and –ve instances (finally finding general and Specific hypothesis)
- finds all the hypothesis that match all the given training Examples.

"Which attribute matching for Training Example" using CEA".

How we solving using above data set.?

Attribute : Time, W, T, C, H, W, G

Task : Goes for a walk or Not
(+ve = yes ; (–)ve No)

finding specific (S) and general (G) hypothesis

$S_0 = < \phi, \phi, \phi, \phi, \phi, \phi >$ ⟸ Representation like this from our End
$G_0 = < ?, ?, ?, ?, ?, ? >$

Step ① x_0 = <Morning, Sunny, Warm, yes, mild, Strong>

+ve means update Specific hypothesis
- ve " " General "

Now it is (+)ve as per Tabulation So

find S_1 = <Morning, S, W, Y, M, S>

Compare (x_0, S_0)

G_1 = < ?, ?, ?, ?, ?, ? >

When we Compare with ϕ, we want to indicate Same attribute from x_0

so $x_0 = S_1$

Step ② :- x_1 = <Evening, Rainy, Cold, No, mild, Normal>

No is there in Tabulation So (-)ve, (i.e) General.
(-)ve So we updating only G_2, S_2 remains same

S_2 = <Morning, S, W, Y, M, S>

~~G_2 = <morning, ?, ?, ?, ?, ?>~~ ~~Sunny~~ <?, ~~Sunny~~, ? ? ? ?>

"Howe to find G_2 means, In Tabula... Compare 2nd Row and 1st Row (i.e) x_1 and x_0, if it differ, we have to mention previous data; if same leave it."

G_2 = <morning, ? ? ? ? ?> <?, Sunny, ? ? ? ?> <?, ?, warm, ?, ? ?>

<? ? ? yes ? ? ?> "Same mild leave it" <? ? ? ? ? ? Strong>

Step 3 :- (+)ve instances.

x_2 = <Morning, S, moderate, yes, Normal, ~~yes~~ Normal>

Note :- for (+)ve → S
same - write
diff - ?

S_3 = < morning , sunny , ?, yes, ? , ? >

(x_2, S_2)
Compare   Only write (✓) for G_3.

G_3 = <morning, ? ? ? ? ?> < ? sunny ? ? ? ?> < ? ? ? yes ? ? >

Step ④ :-

x_3 = < Evening , S, C, Y, H, S >

(+)ve → ③

S_4 = < ?, sunny , ?, yes, ?, ? >

(x_3, S_3)   Only write (✓) for G_4

G_4 = < ?, sunny, ? ? ? ?> < ? ? ? yes ? ? >

<u>Specific</u> <u>hypothesis</u>.

< ?, sunny, ?, yes, ?, ? >

<u>General</u> <u>hypothesis</u>

< ?, sunny ? ? ? ? > < ? ? ? yes ? ? >

— x —

INDUCTIVE BIAS :-

→ Remarks on CE and VS algoritim:-

1) Will the CE algorithm gives us Correct hypothesis ?

2) what training Example Should the learner request next?

→ Inductive learning :

from Examples we derive Rules (Real Time Experience).

(ex):- House Building - Cement ratio we dont know
- So first water mixed more dilute
- so less " mixed becomes more

By the way of Experience we will come to know how much ratio have to mix.

" various Example - got Experience, and applied to New Ex".

- Deductive learning :

- Already Existing rules are applied to our Examples

(ex) known Civil Engineer knows how to build, Cement, water mix.

* Biased hypothesis Space:

Does not Consider all types of training Examples.

Bias means Showing Partiality, Difference.

" All types of Ex Not Consider. So what is the Solution means include all hypothesis."

Ex :

And ↓

Sunny ∧ warm ∧ normal ∧ strong ∧ cool ∧ change ⇒ yes.

Then only Player going the Enjoy spot.

" " " " " ∨ ⇒ NO.

minor changes in one attribute

also player can't Enjoy spot.

* Because of machine learned and habituated the word Change, Simple (∨ change) changes it will tell No.

* Thats why Biased Not Suitable.

Unbiased hypothesis Space:

Representing Set of Examples, from Tabulation

Possible instance $= 3 \times 2 \times 2 \times 2 \times 2 \times 2 = 96$.

Target Concepts : 2^{96} (huge)

(practically Not possible), to learn those many Ex, Thats why Not go for unbiased

* Idea of Inductive Bias :-

Making of addressing Capable of Inductive Bias

Again.

(Ex) We are CSE Engg, But I have to Construct Building, we don't know how to build, By the way we Searched Civil Engg Not available immediatly, What will do, we will Start to do Construct with ideas.

* learner generalizes beyond the observed training Ex to infer New Ex.

$\boxed{\text{'>' – inductively inferred from}}$

$x > y$ = y is inductively inferred from x

x is Predefined ~~is~~ Example. ~~Based~~ in the System, from the x you are defining y.

$\boxed{\text{x giving o/p for y.}}$

<u>Ex</u> :- Learning Alg = L

Training data $D_c = \{x_1, c(x)\}$

New instance $= x_i$ (Now my Task is to classify x_i)

↓

Represented as $L(x_i, D_c)$ ← How do you obtain Result of x_i

$(D_c \wedge x_i) > L(x_i, D_c)$.

already in Predefined System

(i.e) $L(x_i, D_c)$ inductively inferred in $(D_c \wedge x_i)$.

—x—

Trouble free

Decision Tree learning:-

* Mainly used in tree structured Classification and Regression

* Classification Consist of Many Algorithms. One of the Algorithm is decision tree (Tree Based).

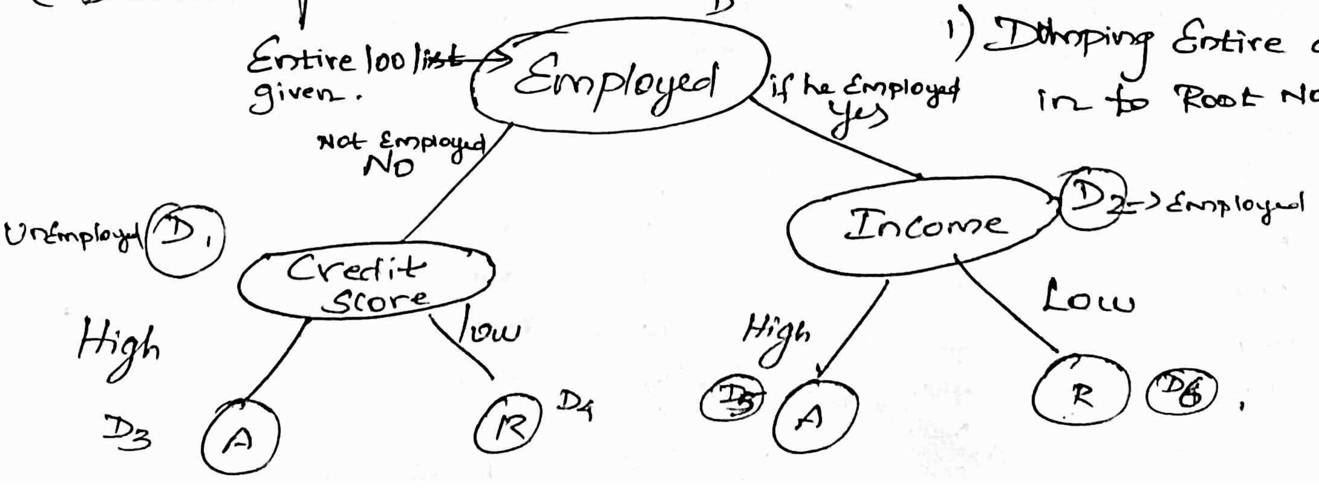
* Dataset $\xrightarrow{\text{given}}$ Algorithm $\longrightarrow$ Classifies the data (By using decision tree Alg).

* When we given a data to classifier, it will Say which Class the data belongs to (yes class, No class, (+)ve, (−)ve) Based on Ex.

* 2 Types of Nodes:-
  1) Decision Node (root node - where the Branch is begin)
  2) Leaf Node (you cannot have further Branches (i.e) last Row)

Ex. Loan System
(Decides if the loan should be approved / Rejected)

1) Dumping Entire data in to Root Node



* 4 data sets we got i.e (D_3, D_4, D_5, D_6), whom you have to approved / Rejected.

Algorithm :

1) In the given data set, choose a target attribute (age of Employee, Income status)

2) Calculate information gain of target attribute

$$\text{Information gain} = \frac{-P}{P+N}\log_2\left(\frac{P}{P+N}\right) - \frac{N}{P+N}\log_2\left(\frac{N}{P+N}\right)$$

3. for remaining attributes, find Entropy

Entropy = IG × Probability
How many Times occur.

$$E(A) = \sum \frac{P_i + N_i}{P+N} I(P_i N_i)$$
Entropy

4. Calculate Gain = IG − E(A)

Based on the gain construct decision tree

—x—

Example & Algorithm of Decision Tree learning:-

Age	Competetion	Type	Profit.
old	yes	S/W	Down
old	NO	S/W	Down
old	NO	H/W	Down
mid	yes	S/W	Down
mid	yes	H/W	Down
mid	NO	H/W	UP
mid	NO	S/W	UP

new	yes	S/W	UP
new	NO	H/W	UP
new	No	S/W	UP

Step 1 :-

Target attribute = Profit.

Step 2 :- Information Gain

$$I_G = \frac{-P}{P+N} \log_2 \left(\frac{P}{P+N}\right) - \frac{N}{P+N} \log_2 \left(\frac{N}{P+N}\right)$$

$\Rightarrow$ P = Count (down) = 5

N = Count (up) = 5

$$= \frac{-5}{10} \log_2 \left(\frac{5}{10}\right) - \frac{5}{10} \log \left(\frac{5}{10}\right)$$

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

$$= -\left(\frac{1}{2} \times -1 \log_2^2 + \frac{1}{2} \times -1 \log_2^2\right) \qquad \therefore \log_n a^m = m \log a$$

$$\therefore \log_2^2 = 1$$

$$= -\left(\frac{1}{2} \times -1 + \frac{1}{2} \times -1\right)$$

$$= -(-1) = 1$$

$$\boxed{IG = 1}$$

Step 3 :- Calculate entropy for Remaining attributes

$$E(A) = \sum \frac{P_i + N_i}{P+N} I(P_i N_i) \qquad [IG \times \text{probability}]$$

To find the Entropy for Non-Target attributes.

Remaining attributes are Age, Competition, Type.

(Age) :-

1) Prepare a table for Each Attribute

rows - Values of undertaken attribute (old, mid, New)

Columns - Values of target Attribute (down, UP)

	down	UP
old	3	0
mid	2	2
new	0	3

Entropy = IG x Probability

P = down Count

N = UP Count

$$\text{Information Gain (Old)} = -\left(\frac{3}{3}\log\left(\frac{3}{3}\right) + \frac{0}{3}\log\left(\frac{0}{3}\right)\right) = 0$$

P = 3

N = 0

$\because \log(1) = 0$

Probability $= \frac{3}{10}$ [In (Ex) Tabulation, How many Age Count 10, from that How many Age (Old) = 3]

$$\boxed{\text{Entropy (Old)} = 0 \times \frac{3}{10} = 0}$$

$$\text{IG (mid)} = -\left(\frac{2}{4}\log\left(\frac{2}{4}\right) + \frac{2}{4}\log\left(\frac{2}{4}\right)\right) = 1$$

Probability $= 4/10$

$$\boxed{\text{Entropy (mid)} = 1 \times \frac{4}{10} = 0.4}$$

~~IG (mid)~~

$$\text{IG (new)} = -\left(\frac{0}{3}\log\left(\frac{0}{3}\right) + \frac{3}{3}\log\left(\frac{3}{3}\right)\right] = 0$$

Probability $= 3/10$

$$\boxed{\text{Entropy (new)} = 0 \times 3/10 = 0}$$

Entropy (Age) = E(o) + E(m) + E(N) = 0 + 0.4 + 0 = 0.4

Step 4:- Entropy = IG − E(A) = 1 − 0.4 = 0.6

Like that for Competition & Type.

—×—

In the same way, Calculate Gain for Other attributes.

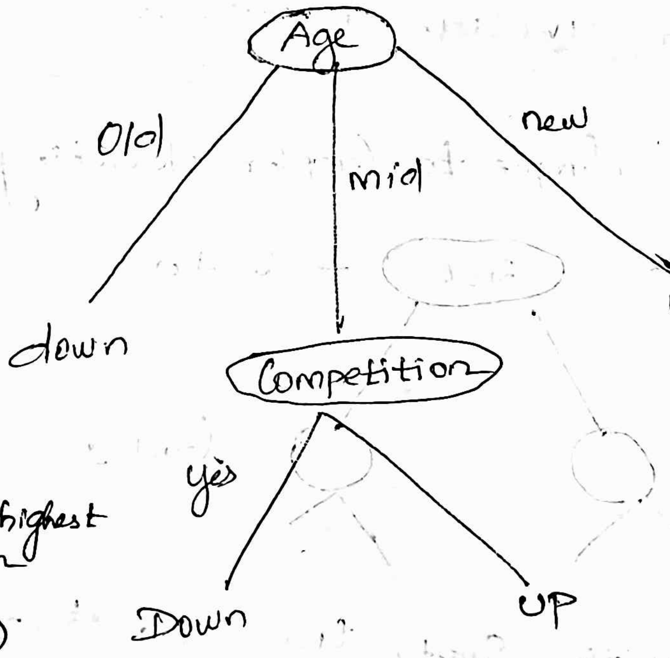
Gain (Competetion) = 0.124

Gain (Type) = 0

Gain (Age) = 0.6

Highest Gain → root Node
(Age) = 0.6.

Age → High value (0.6)



Old → all down

mid → Some down & Some up.

new → all UP

Competition = 0.124

Why Competition → next highest gain

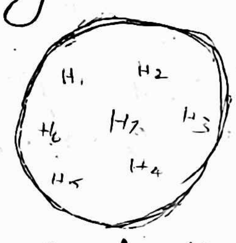
Why not Other (Type)

Type gain value is 0, No Requirement, ignore

—×—

Hypothesis Space Search in Decision Tree Learning:

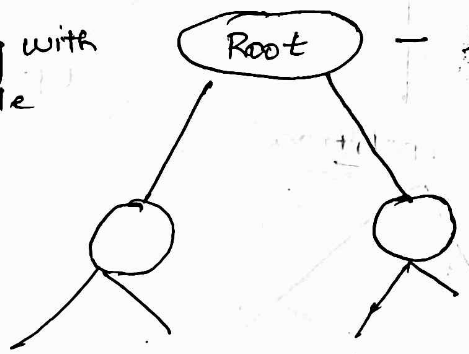
* How to pickup Decision Tree
* $ID_3 \Rightarrow$ Decision Tree
* ID_3 Can be Characterized as Searching a Space of hypothesis for One that fits the training Example
* $ID_3 \rightarrow$ will Search set of possible Decision tree from available hypothesis.



Many hypothesis are here

* ID_3 Performs Simple to Complex Searching.

first starting with root node



– level 0

level 1.

* first Start with Empty tree and keep on adding (Root) node.
* Every Discrete Valued (finite) fn Can be described by Some Decision tree.

Value Not depending on Each other

* Avoid major Risk of Searching incomplete hypothesis. Ex. 2^{96}.
* Has only Single Current hypothesis $\rightarrow$ (Ex) Time duration 1 $\xleftrightarrow{\text{only one}}$ 2 hypothesis will hold

- Cannot determine alternative decision trees.
- Backtracking is Not possible (Level 0, Level-1 can't Replace).
- Can be Extended Easily to noisy data also

—x—

Trouble free

INDUCTIVE BIAS IN DECISION TREE ALGORITHM:-
DT

* first we want to know about Inductive Bias.
* Inductive Bias means Set of assumptions. Based on What criteria
* Inductive Bias of ID_3 Consists of describing the basis selecting DT by which ID_3 Choose One Content decision tree over all the Possible Decision Trees.

ID_3 Search Strategy:- (Among 100 tree, 90 tree long 10 tree Small It will first give a favour to ↑)

1) Selects in favour of Shorter tree over longer ones
2) Selects element with highest IG as Root attribute Over lowest IG ones.

Element IG value depends → Root attribute

Types of Inductive Bias:-

1) Restrictive Bias - Based on Conditions (+ve, -Ve)
2) Preference Bias - Based on Priorities.

ID_3 → Preference. → why means (It will give priority Shorter tree and IG value).

Version Space & Canditate Elimination ⇒ Restrictive

Why Short hypothesis?- According to Occums Razor Prefer Simplest hypothesis that fits the data "all data Should fit in to the tree."

Issues in Decision Tree learning :-

① Reduced error pruning

1) Overfitting the data. Can over come by ↑.

(will take only Discrete)

2) Incorporating Continuous valued attributes

No continuous

3) Determining depth of tree for correct Classification

(Determine Level 0, 1, 2, 3.
When Constructing

4) Handling attributes with different cost

5) Alternative Measures for Selecting attributes.

— x —