* we actually have 4 steps to design a learning system.

* performing all those 4 steps. we arrive a final design.

Final design.

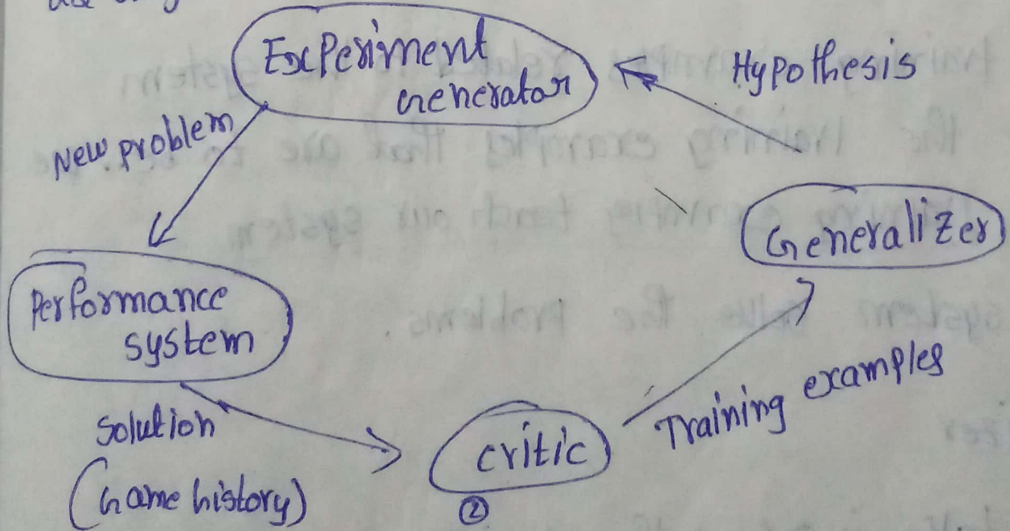
in final design we have 4 different type of modules.

1) Performance system
2) critic
3) Generalizer
4) experiment Generator.

* after perform 4 steps we arrive at this final design

* Final design we have 4 module. Every module have input and output.

* Each module you give something as input something as output.



* The first ① is performance system. in performance system what happend is.

* Any New problem is taken as the input by performance system.

* output you getting is solution tree
* solution tree is Ntg but history
* for this previous what happent. which solution we taken to the problem
* are any solution (or) not. wheather the solⁿ is worked are not.
* whatever the previous history related to New Problem that will be given output as the performance system

→ Nxt is critic module

* The critic module will do is. it will take input as Previous history. which is given by the performance system
* The previous history what is there that will be taken as input by the critic
* And the critic module generate a ~~module that is~~ output regarding training example
* So, the training examples related to the system
* what are the training examples that are to be. we know the training exampleg teach our system
* so the system solve the problems.

→ Generalizer

* It will take input as the training example and it will generate a hypothesis.
* Hypothesis is Ntg but estimation
* it can be true (or) false this was not gaurante every hypothesis is correct.

* that hypothesis will be taken by the experiment Generator as the input

* and it again generates New problem as output
    * if any other issue (or) any other problem that will be generate output.

This is How designing a learning system will go

Experiment generator

in Experiment Generator what happen is it will pic New practic problem

* that New problem will maximize the learning rate
* so, that system will understand more and more critical situation. The system will habitat it.

* we can simpley say that. Role of E Generator
* will take input as hypothesis. based on the hypothesis
* ~~b~~ it will be true (or) false. it is Just our imagination
* based on hypothesis it will again generate some more New Problems
* ~~it~~ that means the hypothesis these Kind of New Problems not get adressed
* What will do the ~~system~~ machine ~~is~~ again the New Problems encountered ?
* it will ~~be~~ be thinking on that way and Keep Generating New problems / New examples.
* So that can teach (or) train system in more better way it becomes more perfect.

* So This is all about the design what we obtain

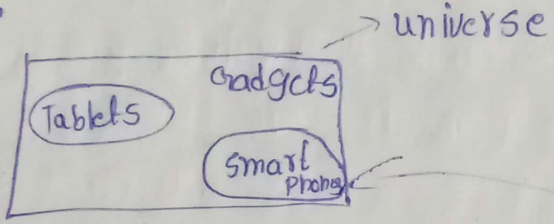
* after we all done the 4 steps in designing the learning system.

concept learning — II

concept learning is Ntg but it help us in finding all the consistant hypothesis (or) the concepts

* so among we have different hypothesis and we have different concepts

ex:



* in universe many are there. we are Just know about Tablets and smartphones

* Each and every gadget ~~object~~ each and every thing will have features

* Those features are defined as binary valued attributes

Features (Binary valued attributes)

The Feature are size, color

*suppose size of ~~smartp~~ tablet, smartphone is small

Size : Large/small — This difened as x1 → X_1

Color : black (Tab) blue (→ for mobile) → X_2

Screentype : Flat, Folded → X_3

shape : square, Rectangle → X_4

* Then How do you define concepts. it is represented as
→ concept = $<x_1, x_2, x_3, x_4>$

* Particulary tablet, Particulary smartphones represented in belows

$<x_1, x_2, x_3, x_4>$ → This is generalized

→ Tablet = < large, black, Flat, square > → but individual is this

→ Smartphone = < small, Blue, Folded, Rectangle >

* This is how represented a concept for tablet and smartphone.

No. of Possible instances = 2^d

d = no. of features

$<x_1\ x_2\ x_3\ x_4> \to 4$

2^4

$\boxed{2^4 = 16}$ → These are possible ins

$\boxed{2^d = 16}$

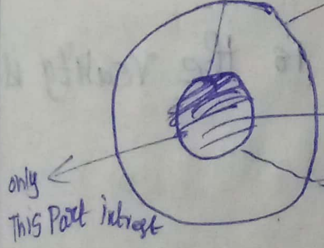
$\boxed{\text{Total possible concepts} = 2^{2^d}}$

2^{16}

* 2^{16} is the no. of possible concepts which are possible for this example which we have taken

whatever conclusion from this → it is apply to entire system.

→ in this concept we are not intrested to take all in this



→ target concept / hypothesis space

only this part intrest

→ we are not consentrate all the space. we just concentrate some space. that is target concept

~~< ϕ, ϕ, ϕ, ϕ > → Reject all~~

* suppose taking tablet Features
< L, B, F, S > → based on this we have to reject it (or) accept it.

* These are 2 hypothesis are there that is common to al concepts.
< ϕ, ϕ, ϕ, ϕ > → Reject all
(most specif hypothesis)

$<?,?,?,?>$ → accept all
(most general hypothesis)

* suppose all the features represented as null (ϕ). Then in that case we have to Reject all thoes.

* It may be best, it may be worst anything represented with null. Then all are Rejected

* it will called as most specific hypothesis.

→ if everything is represented with $<?,?>$. it means anything whatever is there we accept it, wheather is good for you are bad for you. like irrespective of consequence you accept it.

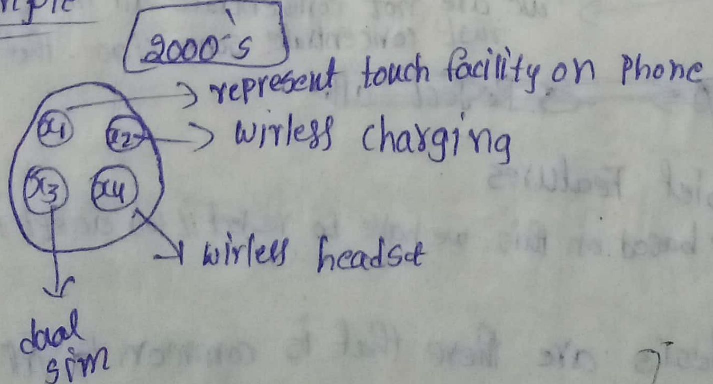
* it is called most general hypothesis.

Main Goal

→ The main goal of concept learning is
* finds all the concepts/hypothesis that are consistent.

→ for example that is hypothesis and what is the reality is belows

Example

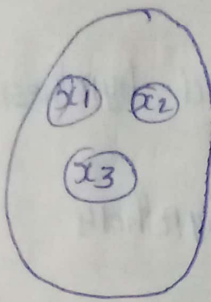


* what have all these sets in 2000's. 20 years back all these things are not possible. it is just assumptions.

* what if touch facility in phn. what if we have wirless charging
what if we have dual sim.

* All These are assumptions at that time

→ in case 2020's



* if thing became possible like touch facility.
* atleast few, out of 4, 3 possibilities are there.
* so, that is hypothesis. it 2020's is reality.
* which things are really implemented. the things which are really consistant.
* Not all the features

→ concept learning as SEARCH :

* Main goal of this search is to find the hypothesis that best fits the training examples.

Examples : EnJoysport learning task

* EnJoysport is Ntg but we have a table that 6 attributes
* 6 attributes

(sky①, Air temperature②, humidity③, wind④, water⑤ and forecast⑥)

↓ only sky has 3 values

→ 3 values — Rainy, cloudy and sunny.

→ Remaining 5 attributes have → only 2 values.

* We have to calculate Different instance possible we have
→ only sky 3, remaining 2

∴ Different instance possible $= 3 \times 2 \times 2 \times 2 \times 2 \times 2$
$= 96$

Note : instances are different, attributes are different, hypothesis is different.

* Now we calculate syntactical dist hypo. we (Additionally add 2 more values - ? and ϕ)

→ syntactically distinct hypothesis = $5 \times 4 \times 4 \times 4 \times 4 \times 4 = 5120$ < ϕ, Rainy, cloudy, sunny, ? >
(Additionally, 2 more values - ? and ϕ)

* for simoilary we to add these two ? ϕ values, so remaining attributes have 4 values

* so we have 5120 syntactically distinct hypothesis possible.

→ Now we calculate semantically distinct hypothesis

semantically distinct hypothesis = $1 + (4 \times 3 \times 3 \times 3 \times 3 \times 3)$ = 973
(Null - taken as common)
(ϕ). only ① more value)

→ after finding all the syntactically & semantically distinct hypothesis.

* we search the best match from all these. (i.e much closer to our learning problem)

Find S - ALGORITHM

* with the help of find-s algorithm we finding a maximally specific hypothesis)

* In Find S ⇝ means most specific hypothesis = ϕ

* in this algorithm considers only positive examples.

* it mean a table the end of attribute have a class → yes/No in that yes/No we have to take only positive values.

→ Maximum yes ⇒ +ve
No ⇒ -ve

Representations

most specific hypothesis = ϕ
most general hypothesis => ?

Algorithm :

step 1 : initialise with most specific hypothesis (ϕ)

→ weare having 4 attributes

$h_0 = < \phi, \phi, \phi, \phi, \phi >$ → 5 attributes. apply 4 ϕ

6 values 6(ϕ)

step 2 : for each and every +ve sample, if the they are -ve samples Ignore them.

* after select +ve sample we have to each and every attribute.
* The New value which we are going to take is equal to hypothesis vale ignore it

if (value = hypothesis value) => Ignore
else
Replace with the most general hypothesis (?)

origin	manufacturer	color	Year	Type	class
JP	Ho	Blue	1980	eco	+ (Yes)
JP	To	Green	1970	Sports	- (No)
JP	To	Blue	1990	eco	+
USA	AU	Red	1980	eco	-
JP	Ho	white	1980	eco	+
JP	To	Green	1980	eco	+
JP	Ho	Red	1980	eco	-

* $h_0 = < \phi, \phi, \phi, \phi, \phi >$

$h_1 = <$ JP, Ho, Blue, 1980, eco $>$ → first +ve attributes values

h_2 = -negative value, so ignore it hypothesis is h_1 itself.

$h_2 = h_1$.

$h_3 = <$ JP, ?, Blue, ?, eco $>$

$h_4 = h_3$

$h_5 = <$ JP, ?, ?, ?, eco $>$

$\boxed{h_6 = < \text{JP}, ?, ?, ?, \text{eco} >}$ → Most generalized

~~h_7~~ = h_6 → This is most general hypothesis

* what is the draw back here is.

Disadvantages:

i) considers only +ve values

* it is not gurante to match all the data. it will not over all the data.

⇒ h_6 may not be sole hypothesis that fits the complete data.

Version Spaces:

* version space is Ntg but x hypothesis H subset of consistant with the training examples.

* Main hypothesis (H) → take some hypothesis. we are taking subset of this based on which condition we deriving a subset?

* There should be consistent with the training example.

* whatever you picking a hypothesis that should be consistent with training examples.

* Training exa

VS H,D =

H =

D =

Consistent

h(x) = c(

hypothesis

* we will le

the version

* So let us

* Algorithm

Clis

D. * we t

1. versio

2. from

hypothe

* for each

that is

* condition

remove

3. output

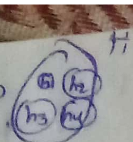
Checking

* So on

output

* output

* Training examples are Denoted with D.



$VS_{H,D} = \{ h \in H \mid consistent\ (h, D) \}$

H = hypothesis

D = training examples

* How do you check whe-ather the given hypothesis is consistant with the training example (or) not
* that check it by $h(x) = c(x)$

Consistent

$h(x) = c(x) \rightarrow$ target function

$\downarrow$ hypothesis

* we will learn the algorithm it will helps to finding the version space.
* so let us see the algorithm

* **Algorithm to obtain version space :**

(List then eliminate algorithm)

1. * we total have 3 steps

→ every hypothesis we assign to version space.

1. version space ← list containing every hypothesis in H.

2. from this step, we should keep on removing inconsistant hypothesis from version space.

* for each training example, $\langle x, c(x) \rangle$ remove an hypothesis that is $h(x) \neq c(x)$.

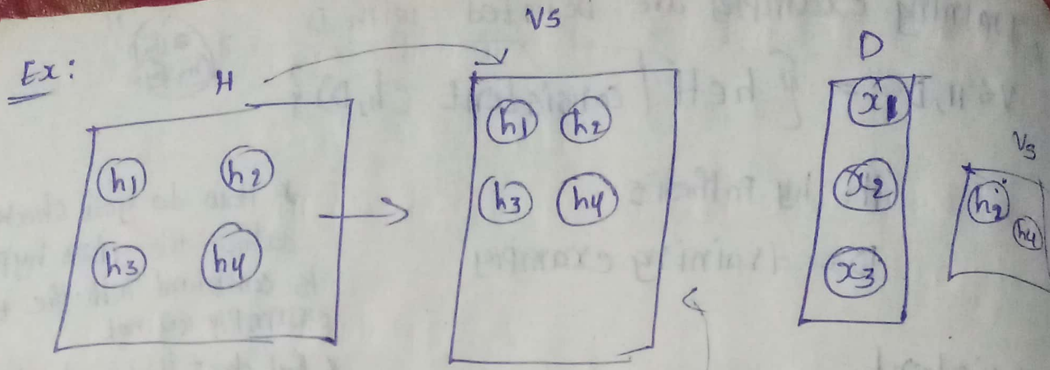
(representing → target function; x ↓ variable/instance)

* condition for consistency $h(x) = C(x)$ but here $h(x) \neq c(x) \rightarrow$ so we remove inconsistang.

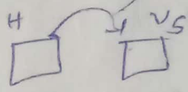
3. output the list of hypthosis into version space after checking for all training examples.

* so once you checking all the training examples x then you need to output the list.
* output the hypthosis what is finalised.

Ex:



* This hypothesis initially what we have to do is all the hypothesis into version space



* once you copied, once you assigned all these into version space.

* No you start checking for the consistency.

$$h(x) = c(x)$$

* No let us defin the 'd'

* d is ntg but training example.

* in d (in set) we have x_1, x_2, x_3

* we are having 4 hypothesis and 3 examples

* Now we have to start from h_1

$h_1(x_1) = c(x_1)$ → h_1 is consistant with first instance x_1

$h_1(x_2) \neq c(x_2)$ → we remove h_1 from vs

Suppose $h_2(x_1) = c(x_1)$ → h_2 is consistant with all so no need to remove it.

$h_2(x_2) = c(x_2)$

$h_2(x_3) = c(x_3)$

Suppose $h_4(x_1) = c(x_1)$

$h_4(x_2) = c(x_2)$

$h_4(x_3) - c(x_3)$

* for h_2 and h_4 they consistant with all examples.
* so the version space only have h_2, h_4 $\boxed{h_2 \ h_4}$

candidate elimination

-> in order to understand the candidate algorithm, we must know Find-s algorithm.

-> in Find(s) we only consider +ve values

-> in candidate elimination both +ve and -ve values (yes/no)

-> it is concern with both specific and general hypothesis.

* at the end we get both specific and general hypothesis

-> For positive samples, Move from most specific hyp (ϕ) to most general (?) hypothesis

-> For Negative samples, Move from general to specefic.

$S = \{\phi \ \phi \ \phi \ \phi \ \phi\}$ + ↓ -> in case +ve sample, it will be moving from most specific hypothesis to the most general hypo

$G = \{? \ ? \ ? \ ? \ ?\}$ − ↑ -> in case -ve sample, it will be moving from most general hyp to the most specific hyp

Algorithm

1. initialize both general and specific hypothesis (S and G)

$S = \{\phi, \phi, \phi, \phi, \dots \phi\}$

$G = \{?, ?, ? \dots ?\}$

depends on no of attributes
suppose 6 attribut so 6(?) 6(ϕ)

2. for each example, you need to first check with example is +ve (or) -ve

if example is positive

you have to change (make specific to general hyp

else example is -ve

make general to specific

Example: Enjoy sport

$S_0 = \{\phi\ \phi\ \phi\ \phi\ \phi\ \phi\}$ $G_0 = \{?\ ?\ ?\ ?\ ?\ ?\}$

6 (ϕ) 6 (?)

Dataset (EnjoySport)

Sky	Temperature	Humidity	wind	water	Forecast	Enjoy
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

1) +ve

for positive we change most specif. to General hyp

→ but it is first row, no predefined hyp so initially we have to write as it is.

$S_1 = \{$'sunny', 'warm', 'normal', 'strong', 'warm', 'same'$\}$

$G_1 = \{?\ ?\ ?\ ?\ ?\ ?\}$ → it initially like there.

2) +ve

(specific to general)

S2 = { 'sunny', 'warm', ?, 'strong', 'warm', 'same' }

G2 = { ? ? ? ? ? ? ? } → in +ve no change general hyp

→ in -ve specific hypothesis is same

3) S3 = { 'sunny', 'warm', ? 'strong', 'warm', 'same' }

we change only G3

← G3 = { < 'suny' ? ? ? ? ? >, < ? warm ? ? ? >

< ? ? ? ? ? 'same' > }

∴ so this general hyp

sunny become rainy so { 's' ? ? ? ? }
warm become cold so { ? 'warm' ? ? ? ? }
same become change so { ? ? ? ? ? 'same' }.

same matches are not written

compare to S3 → same attributes write it otherwise make ?

4) S4 = { sunny' 'warm', ? 'strong' ? ? >

G4 = { < sunny ? ? ? ? ? > < ? 'warm' ? ? ? ? > }

as it is G3

Here same attribute removed becoz S4 have No 'same'.

S4 and G4 ⟹ Final hypothesis.

* so, S4 is Most specif one and G4 is Most General one.

* based on this we teach the machine.

INDUctive BIAS

→ what concept of inductive BIAS?

The Remarks that were obtain on candidate elimination version space Algorithms.

Remarks on CE and VS Algorithms:

1) will the CE algorithm gives us correct hypothesis?

out of CE algorithm we got one specific hypothesis and one General hypothesis. so specific hyp and General hypo which we have obtained is correct or not.

* How do you decide wheather it is correct or not.
* The CE algorithm gives the correct results there is ~~some~~ no errors there some hypothesis 'h' that will correctly exactly describe the target function.

2) what training example should the learner request next?

* so one training example the machine/learner learn. what kind of example the machine has to request next
* that will depend on the type of the problem. type of the question the Machine is faced.
* And also the Machine can request on it's own.

* so, Now what happend if new situation comes, new problem comes
* Then How the Machine will reactaining. in the end of topic we get a full clarity about inductive bias.

before learning we know the Inductive learning, Deductive learning

→ Inductive learning

* From examples we derive rules

* From real life examples are seeng from real life situations we will be deriving the results.

* for example we don't know anything and you started doing some work

* suppose constructing the house, we don't know the how to buy cement, we don't know How much sand has to be taken. we don't anything

* but we construct the house.

* as you do The amount you mix lot of water then it becomes very loose

some

* Then you should understand. wate should not add largely.

* you took large amount of cement and low amount of water it will No become Mixture (creamy) it Just be Powder.

* so you understand ok, i may not take low amount of water

* before we don't know anything. but when you started getting error then you got understand that this will not correct.

* Then you become a generalized version

* That is inductive

→ from examples we will be learning from experience you will be learning the general procedure.

Deductive learning

→ Already existing rules are applied to our examples.

so, in case of Deductive what happend is

* initially you are a civil Engineer. you would know How much amount of cement is added. How much amount water, sand

* everything you know.

Scanned with OKEN Scanner

* you will go to the site. you will mix correctly and apply that
* so, this is deductive
* already existing rules you will be applying, ~~that~~ already existing what we have learned and you will be applying that and constructing that.

* giving the word inductive bias. because you will be learning from examples
* see the Machine will already have some basic knowledge The Machine alredy have some basic examples. based on that examples
* it will handle the New examples.
* so when you teching the you are not give the theory. human beings can understand theory machines can not understand
* based on the example you have given the Machine will learn and that general rules she learn't on itself it will applying to the New comming examples.
* so simply that is what inductive bias.

→ we have 2 types of ~~bias~~ hypothesis
1) Biased hypothesis space.
2) unbiased hypothesis space.

Biased hypothesis space

* biased means showing partiality / differences
* in here biased means all types of examples. all type of training examples.

all situation
→ it will no
Then what
* it includ
Ex: summ
* sky
* Then
* instead
* change
then go
* There i
* but Mo
→ so, This
* unbiase
* in unb
* Now let
* Actually
→ in unbia
Set of
Possi
* it

all situations are not taken into consideration.

→ it will not consider all types of training examples
Then what is solution is this?

* it include all hypothesis

Ex: sunny ∧ warm ∧ normal ∧ strong ∧ cool ∧ change => yes
(→ conjunction)

* sky is sunny ∧ and temp is warm ∧ are these are there the only yes.

* Then only The player is enjoying the sport.

* instead of change they are same it will become => No

* change is luxury first we need basic requirements then go to luxury that is Change

* There is change (or) same the player will njoy sport

* but Machine habitat to must be change only. otherwise No.

→ So, This is why the biased hypothesis is not possible.

* unbiased hypothesis space:

* in unbiased case you are discussing each and every example

* Now let us see How many examples we have

* Actually get involved to everything.

→ in unbiase providing a hypothesis capable of representing set of all examples.

Possible Instances : $3 \times 2 \times 2 \times 2 \times 2 \times 2 = 96$.
(→ sky attribute 3 values)

Target concepts : $2^d = 2^{96}$ (huge).

(pratically not possible)

* it is practically not possible to learn those many examples.

also

* No biased hypothesis, No unbiased hypothesis then what will do

* but solution for unbiased you need to go to biased one

* but in biased one also you should make biase one capable of adressing all the type of examples

* that means you need make it ~~capable~~ general it will adressing all types of examples.

* <u>Idea of Inductive Bias</u> :

* you are a normal ordinary pearson, computer engineer electrical engineer you want to construct house. then what will do you go to civil engineer.

* you are not able to find any civil engineer in the market high demand for civil engineer so not find in the market

* but you want to construct house immediately. you will defintely

* Just for example pratically not impossible.

* you will learn building construction

* you will learn constructing house and you will start the construction of house in your own.

* Here also biased not enough, unbiased is more than enough, and it is pratically impossible to implement the unbalsed one.

* what we are doing, you are comming back to biased one

type of
* so that
Idea o
-> Th
trainin
what
-> base
genre
the r
it is
x ≤
x is alrea
in the sy
Exam
we
How the
we have to do
Predefined
How d
-> (
-> from

and you are making bias capable of adressing all type of examples

* so that is what Inductive bias means.

Idea of Inductive

→ The learner will generalize beyond the observed training examples to infer new examples.

what we have given — what Machine will learn previously.

→ based on the example we have given the learner will genrelize in order to answer (or) inorder to adress the newly coming examples.

$\succ$ symbol means → Inductively inferred from

$x \succ y$ ⇒ y is inductively infered from x.

x is alredy predifined in the system. → it is classification for y (or) output for y.

Example

we having learning alg

learning algorithm = L → from learning alg you get training data

training data $D_c = \{x, c(x)\}$ → we got new inst

New instance = x_i → x_i is predefined in system we have to goal is calculate x_i

Represented as $L(x_i, D_c)$

→ How do you classify this

How do you obtain this x_i

How the result of xi?

we have to do inductively infered from the predefined examples in system.

How do you do that?

→ $(D_c \wedge x_i) \succ L(x_i, D_c)$

$L(x_i, D_c)$ inductively infered from $(D_c \wedge x_i)$

→ from the existing system from the predefined examples you are going to inductivly infered a new example.

* And if we are not able to remind it. write the concept
• you are able to remind write it. your answer become lengthy.

5.

Decision Tree Learning

where you will use in decision tree?

→ This decision tree is mainly used in classification and Regression also.

* in tree structured classification we have some algorithm
* we have Navie bayes some are there. algorithm
* in classification decision tree is one of ~~classifier~~. it is
• tree based classifier

* we will be classifying the data in tree structure.

* we see. How it is implemented in classification?

→ actually you are having a dataset. you have a raw data

Alg means decision tree alg

Dataset → Algorithm → classifies the data

* you are giving raw datased to a Algorithm. that algorithm will classify the data.

* → what is the classification?

when you are giveing the data to the classifier. it can any type of classifier (Navies bias, decison clasifier) are any other.

* it will which class the data belongs to.

* The data belong to yes class/ No class, +ve class/-ve class..

• based on the example it will tell us to which class the particular data belongs to. it will divide the data into sub categories, into 2 categories 3 catg 4 catg also.

* ~~we are~~ so, This about the decision tree Learning
* we are classifying the data ~~into~~ using decision tree algorithm

2 types of Nodes

1. Decision Node
2. Leaf node

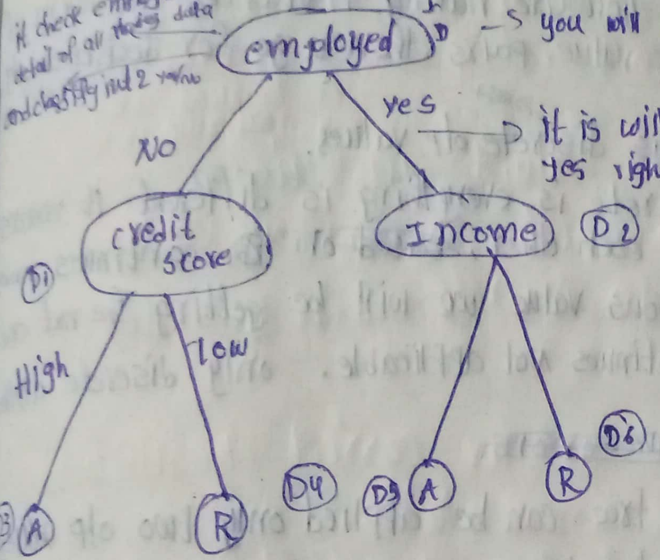
example: loan system

(decides if the Loan should be approved/Rejected)

it check employment detail of all the data and classify into 2 values

→ you will be dumping the entire data in decision node.

→ it is will check first is employed or not. yes right side No left side



6 classification are there.

* first we check employment details
* Then we check credit score (or) Income. based on Employment details
* classify again them in credit score and Income you got high, low, high, low.
  D3 D4 D5 D6
* if the credit score is high it is approved. if income is high so approved loan. if income low Reject it.
* from income will low How will pay the Loan.
* This is why the decision tree is.

The decision tree explanation

Appropriate Problems for decision tree learning

Decision tree is best suited to problems with the following characteristics.

(Age -> old, mid, new) — attribute, values, pair

* ~~first~~ we have instances

1. Instances are represented by attribute value pairs.

   * when you are having instances in form of attribute value pairs. then u go for decision tree algo

2. The target function has discrete o/p values.

   * discrete is everything is different it means independent on each other, ~~based on~~ continues means based on previous value we will be getting the next one. in case of continues Not applicable. only discrete applicable

~~3. Training data can have errors.~~

* decision tree can be applied only two o/p values it can expand if having 3 o/p values. suppose in previous example we have learnt only 2 o/p values (1) is up (2) is down on Profit attribute. even if have something else, apart from up, down, study also there not up not down some are in study. you use the decision tree algorithm.

3. Training data can have errors

   * when though the training data have errors also in that case also you can use it decision tree algorithm. so supports for errors also

4. may contain missing attribute values also.

   * even though the training data is having some missing values also. in that case also you can apply decision tree

* for example. in our Previous example
* when we are constructing the decision tree but did not consider the type.

* so we were not worried about type attribute.

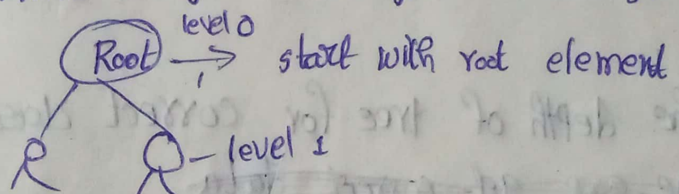
Hypothesis space search in decision tree learning

ID3 → is nothing but decision tree

ID3 can be characterised as searching a space of hypothesis (many concepts are there) for one that fits the training examples.

ID3 → will search set of possible decision trees from available hypothesis

ID3 Performs simple to complex searching.



* first, start with empty tree and keep on adding

* - every discret valued (finite) function can be described by some decision tree

* - avoids major risk of searching incomplete hypothesis.
    * The required hypothesis according to our data is to be present in the hypothesis

* - has only single current hypothesis
* - cannot determine alternative decision trees.
* - Backtracking is not possible.
* - it can be extended easily to noisy data also.

Issues in Decision tree learning

1. overfitting the data → * it mean. if we are depending too much on the ~~data~~ training data. based training data only we calculate Information Gain (IG) we are calculating Entropy (E) we also calculate overal Gain. based on that only we constructing the decision tree. so we are completely reling on the training examples. we depend too much on training data the situation will happen in overfitting for testing data it will not good. To overcome overfitting
   ① Reduced error pruning
   ② Post rule pruning.

2. Incorporating continous valued attributes.
   * it cannot acomodate for continous values. so convert continous to discrete. then you apply this

3. Determining the depth of tree for correct classification
   * ~~we are get correct data~~
   we need to do the classification correctly. How much depth we need to go. it will go the level 0, level 1, level 2 at what depth we are get exact classification of data

4. Handling attributes with different costs (importance (or) weight)
   * We are going to priority of one attribute. but that attribute is not so much influencing. that attribute not covers variety of examples. Then obviously if we are giving more importance it. and less importance to a attribute that covers variety of examples.

5. Alternati...
   * you selecti... Inform...
   * we d... calcul... forge...

5. Alternative measure for selecting attributes.

* you don't have any alternate measures for selecting attributes. for example you calculate the information gain, we calculate entropy like that

* we don't have any alternate ~~calculation~~, apart from doing calculations. Here also you giving importance to the target attribute. it is again go under bias.