UNIT – III

* BAYES THEOREM :-

Bayes theorem gives probability of an event based on prior knowledge of Conditions (Previous knowledge)

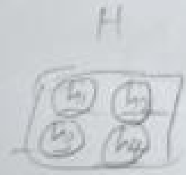
$$\boxed{P(A|B)} = \frac{P(B|A) \cdot P(A)}{P(B)}$$

P(A|B) ↓ hypothesis (posterior prob.)

P(B) ↳ marginal

$P(B|A) \rightarrow$ liklihood

$P(A) \rightarrow$ prior



Proof :- Let us take 2 events : A & B

$$P(A|B) = \frac{P(A\cap B)}{P(B)}$$

$$P(B|A) = \frac{P(B\cap A)}{P(A)}$$

$P(A|B)$ – Conditional prob.
↓
Means Prob. of A at given Prob. of B

$$P(A|B) \cdot P(B) = P(A\cap B) \quad \text{--- ①}$$

$$P(B|A) \cdot P(A) = P(B\cap A) \quad \text{--- ②}$$

$P(A\cap B) \rightarrow$ Common in A&B.

① & ② are equal, since $P(A\cap B) = P(B\cap A)$

$$\therefore \quad P(A|B) \cdot P(B) = P(B|A) \cdot P(A)$$

$$\boxed{P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}}$$

A - Hypothesis , B - Given data

$P(A|B)$ – Finding prob of hypothesis when Prob. of training examples given. da

$P(B|A)$ – Find prob. of given data provided with prob. of hypothesis that is true.

P(A) = Prob. of hypothesis before Considering the given data.

P(B) = Prob. of given data.

Example :- 1) P (King | face) → Prob. of King given that it is a face Card.

↓

J, K, Q

Heart, ACE, SPADE, DIAM

3 3 3 3

4×3 = 12 face Cards.

$$P(\text{King} \mid \text{face}) = \frac{P(\text{face} \mid \text{King}) \cdot P(\text{King})}{P(\text{face})}$$

P(face | King)

$\frac{4}{4} = 1$

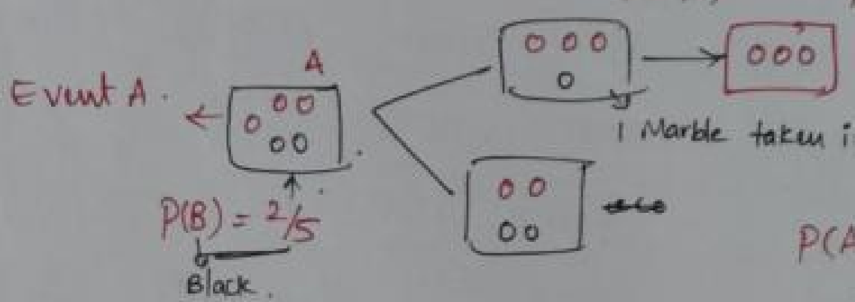
face Kings ← $\frac{4}{4}$

$$\frac{1 \times \frac{\cancel{4}\ 1}{\cancel{52}\ 13}}{\frac{\cancel{12}\ 3}{\cancel{52}\ 13}} = 1/3.$$

2)

Prob. of black marble at given event A.

P(B | A) 1/4



P(B | A) = 1/4

P(A ∩ B) = 2/5 × 1/4

= 1/10

↓

Combined prob. of events A & B.

* Bayes theorem AND Concept theorem Learning:

→ what is the relation between Bayes theorem and Concept learning?

→ Bayes theorem calculates the probability of each possible hypothesis and outputs the most probable one.

$\underbrace{\text{most probable}}_{\text{Highest probability.}}$

* Bruteforce Bayes Concept learning:

1) For each hypothesis in H calculate posterior probability.

$$P(h/D) = \frac{P(D|h)\,P(h)}{P(D)} \quad \text{—— ①}$$

h – hypothesis
D – Set of training examples

2) Output the hypothesis

hmap (hmL) – maximum likelihood with highest probability

$$h_{MAP} = \underset{h \in H}{\operatorname{argmax}}\ P(h/D)$$

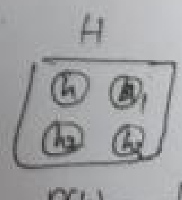
H – hypothesis space

To calculate, we need values of $\underline{P(h)}$ and $P(\underline{D|h})$

Some assumptions,

1) Training data D is noise free (No irrelavant data)
2) Target Concept C is present in hypothesis Space H
3) we have no prior reason to believe that any hypothesis is more probable than any other.

$$P(h) = \frac{1}{|H|} \quad \text{for all } h \text{ in } H$$



$$P(D|h) = \begin{cases} 1 & \text{if } d_i = h(x_i) \text{ for all } d_i \text{ in } D \\ 0 & \text{Otherwise} \end{cases}$$

$P(h) = \frac{1}{|H|}$

from ① $P(h/D) = \frac{P(D/h)\, P(h)}{P(D)}$

Case 1): Hypothesis h is inconsistent i.e $P(D/h) = 0$.

$$\therefore P(h/D) = \frac{0 \times P(h)}{P(D)} = 0$$ [$\because d_i \neq h(x_i)$ So $P(D/h)=0$]

Case 2): Hypothesis is Consistent i.e $P(D/h) = 1$

$$P(h/D) = \frac{1 \times P(h)}{P(D)} = \frac{\frac{1}{|H|}}{\frac{|VS_{H,D}|}{|H|}}$$ [$\because d_i = h(x_i)$ So $P(D/h) = 1$] Consistent new

$$= \frac{1}{|VS_{H,D}|}.$$ [$\because P(D) = \frac{|VS_{H,D}|}{|H|}$ Picking version Spaces from overall hypothesis H]

$$\therefore P(h/D) = \begin{cases} \frac{1}{|VS_{H,D}|} & \text{if } h \text{ is Consistent with } D \\ 0 & \text{otherwise.} \end{cases}$$

$\frac{V.S.}{}$ (H)

* Maximum Likelihood And Least Squared Error Hypothesis:

To find the maximum likelihood hypothesis in bayesian learning,

$$h_{MAP} = \underset{h \in H}{\operatorname{argmax}} \; p(h/D)$$

P - Prob. density fun.

→ Let us take training instances $(x_1, x_2 --- x_n)$ and consider target values $D = (d_1, d_2, --- d_m)$ we write $P(D/h)$ as product of $\underbrace{p(d_i/h)}_{\text{Prob. of each instance at given h.}}$

$$h_{MAP} = \operatorname{argmax} \prod_{i=1}^{m} P(d_i/h)$$

By assuming normal distribution,

$$f(x/\mu) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

x - instance
μ - Mean
σ - Standard deviation

$$h_{MAP} = \underset{h \in H}{\operatorname{argmax}} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(d_i-\mu)^2}$$

$$= \underset{h \in H}{\operatorname{argmax}} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(d_i-h(x_i))^2}$$

$[\because f(x/\mu) = f(d_i/h(x_i))]$

→ use ln function (log. fun) → Π becomes Σ.

$$= \underset{h \in H}{\operatorname{argmax}} \sum_{i=1}^{m} \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}(d_i - h(x_i))^2$$

↓ no 'i' term so const

$$= \underset{h \in H}{\operatorname{argmax}} \sum_{i=1}^{m} (1) - \frac{1}{2\sigma^2}(d_i - h(x_i))^2$$

↓ becoz of '-ve' symbol argmax becomes argmin.

$$= \underset{h \in H}{\arg\min} \sum_{i=1}^{m} \frac{1}{2\sigma^2} (d_i - h(x_i))^2$$

$$= \underset{h \in H}{\arg\min} \sum_{i=1}^{m} (d_i - h(x_i))^2 \quad \left[\because \frac{1}{2\sigma^2} \text{ is const}\right].$$

$$\boxed{\therefore h_{MAP} \text{ or } h_{ML} = \underset{h \in H}{\arg\min} \sum_{i=1}^{m} (d_i - h(x_i))^2}$$

$\therefore$ The maximum likelihood hypothesis is the one which has minimum squared error between the hypothesis. (Least squared error)

* Minimum Description Length Principle :-

Assumption:

Representing a concept in minimum possible way - Then it is said to be good one.

Mathematically,

$$h_{MAP} = \underset{h \in H}{\arg\max}\ P(D|h)\, P(h)$$

Applying logarithm,

$$\underset{h \in H}{\arg\max} \log P(D|h) + \log P(h) \quad [\because \log(ab) = \log a + \log b]$$

$$\underset{h \in H}{\arg\min} [-\log P(D|h) - \log P(h)].$$

(minimum length | Short hypothesis is preferred)

Example:

1) Let us consider a probability of designing a code to transmit messages drawn at random from a set D where probability of drawing an i^{th} message $= P_i$

2) While transmitting, we want a code that minimizes the expected no. of bits.

To do this, we should assign shorter codes to the most probable.

We represent the length of message i with respect to 'c' as $\underset{(\text{Length})}{L_C(i)}$ i - msg. , $L_C(1)$ - first msg. $L_C(2)$ - Second msg.

$$\therefore h_{MAP} = \operatorname{argmin} L_{C_H}(h) + L_{C_{D|h}}(D|h)$$

$C_H \rightarrow$ optimal encoding for H

$C_{D|h} \rightarrow$ optimal encoding for D given h.

$$\therefore h_{MDL} = h_{MAP}.$$

* BAYES OPTIMAL CLASSIFIER :

Bayes optimal classifier is a probabilistic model that makes the most probable prediction for a new example.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

for a dataset,

$$x = \{x_1, x_2, x_3 \text{ - - - } x_n\} \{\underset{\rightarrow \text{yes/NO.}}{y}\}$$

$$P(y|x_1 x_2 \text{ - - - } x_n) = [P(x_1|y) \cdot P(x_2|y) \text{ - - - } P(x_n|y)] \times P(y)$$

For a dataset,

$$x = \{x_1\ x_2\ x_3\ ---\ x_n\}\{y\}$$

$$P(y \mid x_1, x_2, ---\ x_n) = \frac{[P(x_1 \mid y) \cdot P(x_2 \mid y) \cdots P(x_n \mid y)] \times P(y)}{P(x_1) \cdot P(x_2)\ ---\ P(x_n)}$$

$$= \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}{P(x_1) P(x_2) --- P(x_n)}$$

$$\Rightarrow P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

Suppose 10 samples.
$x_1 \rightarrow 2$, $x_2 \rightarrow 4$
$P(x_1) \rightarrow \frac{2}{10}$ $P(x_2) \rightarrow \frac{4}{10}$
we are getting optimal sol ~~for the~~ value, so can eliminate const [deno]

Example:

* outlook ~~& * temperature~~.

	Yes	NO	P(Y)	P(N)
Sunny	2	3	2/9	3/5
Outcast	4	0	4/9	0/5
rain	3	2	3/9	2/5
Total	9	5	100%	100%

* temperature.

	Yes	No	P(Y)	P(N)
Hot	2	2	2/9	2/5
mild	4	2	4/9	2/5
Cold	3	1	3/9	1/5
Total.	9	5	100%	100%

* play

Yes	9	9/14
NO	5	5/14
Total	14	100%

III 5

Total (Sunny, hot)

$$P(\text{Yes} \mid \text{sunny, hot}) = P(\text{Sunny} \mid \text{yes}) \times P(\overset{\text{Hot}}{\cancel{\text{Sunny}}} \mid \text{yes}) \times P(\text{Yes})$$

$$= \frac{2}{9} \times \frac{2}{9} \times \frac{9}{14} = 0.031$$

$$P(\text{No} \mid \text{sunny, hot}) = P(\text{Sunny} \mid \text{NO}) \times P(\text{hot} / \text{NO}) \times P(\text{NO})$$

$$= \frac{3}{5} \times \frac{2}{5} \times \frac{5}{14} = 0.08571$$

$$\text{Total} = 0.031 + 0.08571 = 0.27$$

$$P(\text{Yes}) = \frac{0.031}{0.27} = 0.114$$

$$P(\text{No}) = \frac{0.08571}{0.27} = 0.317$$

probability of No is more, therefore player will not enjoysport.

* GIBS Algorithm :-

1) chooses one hypothesis at random, according to $P(h|D)$
2) Use this to classify new instance.

$$\boxed{E[\text{error}_{gibs}] \leq 2E[\text{error}_{Bayesoptimal}]}$$

→ **Why GIBS :-**

Bayesian optimal classifier will give best results, but need more hypothesis So more expensive.
So we go for GIBS algorithm with the same procer and also the error we get in GIBS algorithm will be $\leq \underline{2}$ (error in Bayesian optimal classifier)

* NAIVE BAYES CLASSIFIER :-

- classification technique based on Bayes theorem with an assumption of independence among features.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}.$$

Example :- Prb: fruit → { Yellow, sweet, long }

Fruit	Yellow	sweet	long	total	
orange	350	450	0	650	{ ∵ repeated fruit)
Banana	400	300	350	400	
others	50	100	50	150	
total	800	850	400	1200	

→ we need to find/pick more Yellow, more Sweet & more long/lengthy one from the table. IV (6)

Sol: $$P(\text{Yellow}/\text{orange}) = \frac{P(\text{orange}/\text{Yellow})\, P(\text{Yellow})}{P(\text{orange})}$$

$$= \frac{\frac{350}{800} \times \frac{800}{1200}}{\frac{650}{1200}} = 0.53$$

$$P(\text{sweet}/\text{orange}) = \frac{P(\text{orange}/\text{sweet})\, P(\text{Sweet})}{P(\text{orange})}$$

$$= \frac{\frac{450}{850} \times \frac{850}{1200}}{\frac{650}{1200}} = 0.69$$

$$P(\text{Long}/\text{orange}) = \frac{P(\text{orange}/\text{long})\, P(\text{long})}{P(\text{orange})}$$

$$= \frac{0 \times 400/1200}{650/1200} = 0$$

$$P(\text{Fruit}/\text{orange}) = P(\text{Yellow}/\text{orange}) \times P(\text{Sweet}/\text{orange}) \times P(\text{long}/\text{orange})$$

$$= 0.53 \times 0.69 \times 0 = 0$$

$$P(\text{Fruit}/\text{Banana}) = P(\text{Yellow}/\text{Banana}) \times P(\text{Sweet}/\text{Banana}) \times P(\text{long}/\text{Banana})$$

$$= 1 \times 0.75 \times 0.89 = 0.65$$ → More value

$$P(\text{Fruit}/\text{others}) = P(\text{Yellow}/\text{others}) \times P(\text{Sweet}/\text{others}) \times P(\text{long}/\text{others})$$

$$= 0.33 \times 0.66 \times 0.33 = 0.072$$

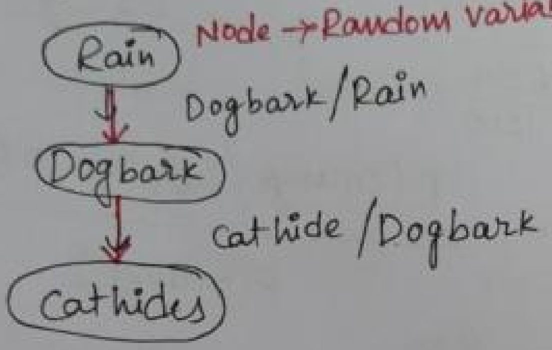
∴ Banana is having more probability, so we can pick Banana which is satisfying the given problem conditions

* BAYSIAN BELIEF NETWORKS :-

- 2 important Concepts.

(1) Directed Acyclic Graph (DAG)
(2) Conditional Probability table (CPT)

* Directed acyclic Graph.



→ when it rains, the dog will bark, when dog barks, the cat hides (May / May not happen)

→ Acyclic (no loop)

* Conditional probability table :-

	Rain (R)	Not Rain (~R)
Bark (B)	9/48	18/48
Not Bark (~B)	3/48	18/48

(B=T & R=T) = 9/48 = 0.19

(B=T & R=F) = 18/48 = 0.375

(B=F & R=T) = 3/48 = 0.06

(B=F & R=F) = 18/48 = 0.375

→ when we calculate the conditional probability, we need to calculate w.r.t the parent node (Rain) But not with child node, that's why we are not considering cat hide (child node) here.

Rain → Bark → cat hide.

Parent to Bark   Parent to C.H.

* Bayesian Belief N/ws :-

→ Bayesian belief N/w is a probabilistic graphical model (PGM) that represents conditional dependencies between random variables through DAG. (Direct Acyclic Graph)

→ Also suitable for representing probabilistic relation between multiple events (more than 2 events)

Ex :- 2)

Alarm detection System

Given probabilities are,

$P(B = T) = 0.001$

$P(B = F) = 0.999$

$P(E = T) = 0.002$

$P(E = F) = 0.998$

Probability of Alarm. (Parents of Alarm → B & E)

Burglary (B)	Earthquake (E)	P (A = T)	P (A = F)
T	T	0.95	0.05
T	F	0.99	0.06
F	T	0.29	0.71
F	F	0.001	0.999.

Probability of P1 (for P_1 & P_2) parent → Alarm.

Alarm (A)	$P(P_1 = T)$	$P(P_1 = F)$
T	0.90	0.10
F	0.05	0.95.

Probability of P2.

A	$P(P_2 = T)$	$P(P_2 = F)$
T	0.70	0.30
F	0.01	0.99.

Sol → Find the probability of P_1 is T, P_2 is T, A is T, B is F and E is F.

i.e $P(P_1, P_2, A, \sim B, \sim E)$

root nodes no parents so not taking Condi. Prob.

$$= P(P_1/A)\, P(P_2/A)\, P(A/\sim B, \sim E) \cdot P(\sim B)\, P(\sim E)$$

$$= 0.90 \times 0.70 \times 0.001 \times 0.999 \times 0.998. = 0.00062$$ //.

* EM Algorithm (Expectation - Maximisation)

→ Used to find latent variable (Not directly observed variables)

→ Basic for many unsupervised clustering Algorithm.

* Steps involved in EM algorithm.

1. Initially, a set of initial values are Considered.
   A set of incomplete data is given to system.

2. Next Step - expectation Step → E Step.
   Here, we use observed data to $\underbrace{\text{estimate}}_{\text{By previous data.}}$ or guess the values (incomplete data) of missing / incomplete data.

3. Maximisation Step or M - Step.
   Here, we use the Complete data generated in preceding e-step to update the values.

4. we check if values are converging / not.
   If Converging - Stop.
   Otherwise, repeat Step 2 & 3 (E & M) till the Convergence occurs

* Usage.

1) Used to fill missing data.
2) Used for unsupervised clustering.
3) Used to discover Values of latent Variables

* Advantages :

1) With each iteration, likelihood increases.
2) E-Step & M-Step are easy to implement.

* Disadvantages:

1) Slow Convergence.
2) Make Convergence to local optimal only.

* INSTANCE BASED LEARNING.

Memorise and then apply.

- Instead of performing explicit generalisation, it compares new problems with instances in training, which are stored in memory.

  Example : spam mails.

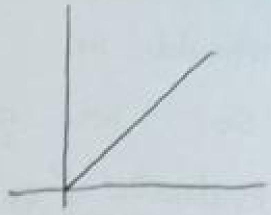
→ Also called as memory based learning / lazy learning.

→ - done with 3 different approaches.

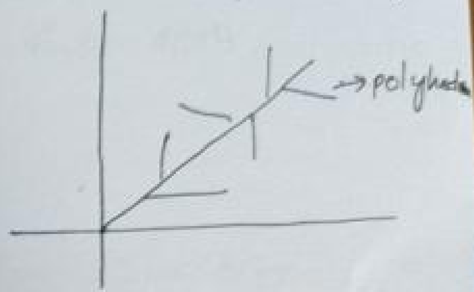
Instance Based learning

- Lazy learners
  - Ex (KNN)
- Radial based functions
  - Radial Base funs (RBF)
- Case based Reasoning (CBR)

Ex: Initially, we have $f(y) = 2x+5$.



If we have $f(y) = 2x^2 + 3x^{\infty}$ and, divides into pockets/segments



* K-NEREST NEIGHBOUR ALGORITHM (KNN):

- example for lazy learning.

Ex: Given data Query ⇒ $x =$ (Maths = 6, Comp.sc = 8)

and $K = 3$. - nearest neighbours.

classification - pass/fail.

	Maths	CS	Result
1)	4	3	F
2)	6	7	P ✓
3)	7	8	P ✓
4)	5	5	F.
5)	8	8	P ✓

Euclidean distance (d)

$$d = \sqrt{|x_{O_1} - x_{A_1}|^2 + |x_{O_2} - x_{A_2}|^2}$$

O — Observed value

A — Actual value.

1 — Maths

2 — CS.

(1) calculate $d_1 = \sqrt{(6-4)^2+(8-3)^2} = 5.38$ //

✓(2) $d_2 = \sqrt{(6-6)^2+(8-7)^2} = 1$ //

✓(3) $d_3 = \sqrt{(7-6)^2+(8-8)^2} = 1$ //

(4) $d_4 = \sqrt{(6-5)^2+(8-5)^2} = 3.16$ //

(5) $d_5 = \sqrt{(6-8)^2+(8-8)^2}$

$= 2$ //

→ we have to choose 3 neighbours, the distance should be as min as possible.

→ So, we get 3-pass, so given query is evaluated / calculated on pass category based on the KNN algorithm.

Here, in 3 neighbours $\underset{\text{Majority}}{\underline{3P}}$ & 0F.

* Regression:-

Satistical tool used to understand and quantify the relation between 2 or more variables.

* linear Regression:

$$y = \beta_0 + \beta_1 x + \epsilon$$

y - dependent variable

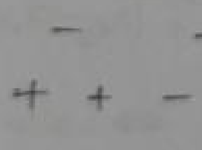
x - Independendent Variable

β_0 - Constant / Intercept

β_1 - x - slope / Co-efficient

ϵ - Error.

→ best suited for linearly seperable data only.



Non-linearly seperable.

→ for non linearly separable data we use locally weighted regression. III (10)

* locally weighted Regression

— To overcome the problem of non linearly Separable data.

— LWR algorithm assigns weights to data to overcome the prob.

— Computationally more expensive.

Finding weights – ? By Kernel Smoothing.

$$D = a\, e^{-\frac{\| x - x_0 \|}{2c^2}}$$

x → each training i/p

x_0 → Value we are predicting

→ If the i/p is more closure to the predicting value then the weight at that feature. (data item)

C – Constant

— we Construct a weight matrix (ω), for each training i/p (x) and for the value we are trying to predict (x_0) [If 10-data set is there, 11 – weight matrices – 1 for x_0, 10 for x]

weight matrix – diagonal matrix

$$\boxed{\begin{array}{l} \beta = (x^T \omega x)^{-1} x^{-1} \omega y \\ \beta \text{ – model parameter} \end{array}}$$

then, Prediction Can be defined as

$y = \beta x_0$, $y \Rightarrow$ Prediction

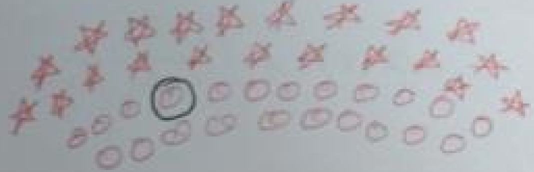
* Drawbacks:-

1) Need to evaluate whole dataset everytime.
2) Computation cost is more
3) Memory requirement is more.

* RADIAL BASIS FUNCTIONS :-

- used in ANN
- has only one hidden nodes

Example :-



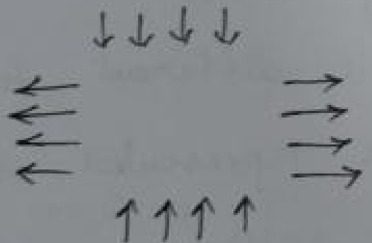
→ 2 classes
(Stars & circle)

→ data is not linearly separable.

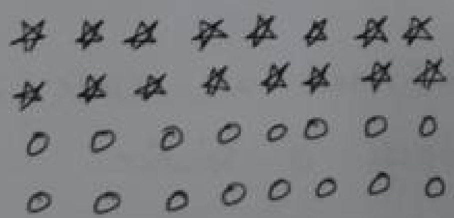
- 2 steps.

1. Increase the dimensionality (2D - 3D)
   (But this step is not mandatory, only based on requirement)

* 2. Expand the direction (Horizontal)
   Compress the direction (Vertical)



Resultant dataset is



How RBF works?

- consider one center randomly.
- draw concentric circles
  (Same center)

To expand / compress, we use ③ functions

1) multiquadric :

$$\phi(r) = (r^2+c^2)^{1/2}$$

$c > 0 \Rightarrow$ Constant

2) Inverse multiquadric :

$$\phi(r) = \frac{1}{(r^2+c^2)^{1/2}}$$

** 3) Gaussian function :

$$\phi(r) = \exp\left[\frac{-r^2}{2\sigma^2}\right]$$

σ - const

* Case Based Reasoning :-

All instance based learners have 3 properties

*1) They are lazy learners

*2) classification is different for each instance.

3) Instances are represented with n dimensional Euclidean space.

In CBR,

Everything is considered as case and based on previous cases - we propose a solution.

- Instances are represented as Symbols (not values)

CBR has 3 Components :

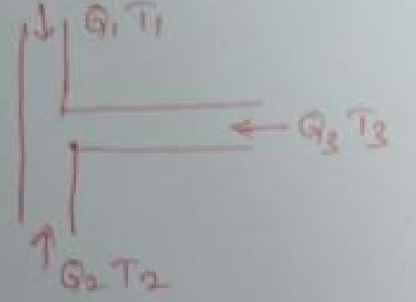
1. Similarity functions or distance measure
2. Approximation / Adjustment of instances
3. Symbolic representation of instances.

For modelling CBR, we use CADET system (Case based design tool)

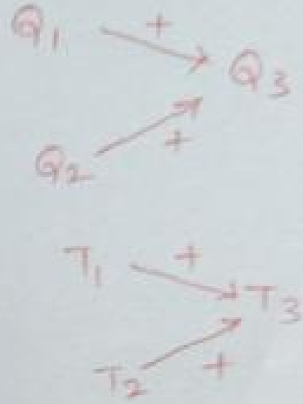
has 75 predefined libraries.

Example: Modern water Taps.

T- Junction Pipe:



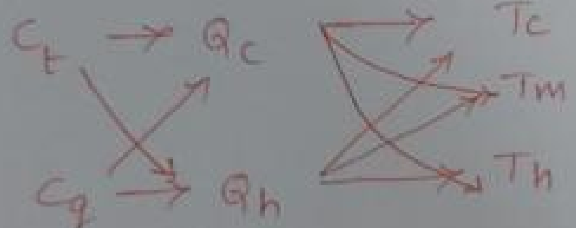
Functions.



Q- water flow

T - temp

Another tap: Control temp and waterflow.



$C_T \rightarrow$ Control of temp

$C_Q \rightarrow$ Control water flow

→ Based on previous predefined cases, giving sol to the new system.

Remarks on Lazy and Eager Algorithms:-

* Lazy learning:-

1. Simply stores training data and waits until it gets a test tuple
2. Less training time, more prediction time.
3. Ex: All instance based learning algorithms

* Eager learning:

1. when we give a training Set, it Constructs a model for classification before getting new example.
2. More training time, less prediction time.
3. Ex: Decision tree, Naive Bayes, ANN etc.