Lecture 1 23/10/23

- Supervised Learning

→ A set of patterns with class label information (i.e training pattern) is available

→ Make a classification model with the help of training pattern.

→ Assign the class label to new pattern to a predefined classes by classifier through collection of similarity / dissimilarity measures.

→ Probabilistic Model.
  ↳ Baye's classifier

→ Similarity Based
  ↳ MDC, KNN, LDC, QDC

→ Neural network.
  MLP, RBF, Hopefield.

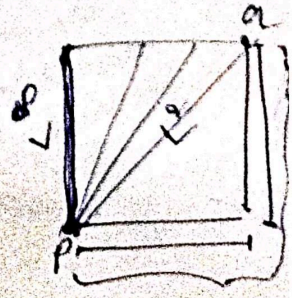
=> Graph based decision tree

→ SVM

→ Random Forest.

- a pattern is a point in n-dimensional space, where n is no. of features present in the data set.

L^p norm, $p=1$, City Block
$p=2$, Euclidean distance
$p=\infty$, L^∞ → supremum. Max $|x_i - y_i|$ over all i



$L_1 > L_2 > L_3 > L_\infty$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

$n \longrightarrow$ no. of features/dimensions.

$$Z = x - y = \begin{bmatrix} x_1 - y_1 \\ x_2 - y_2 \\ \vdots \\ x_n - y_n \end{bmatrix}$$

$$d_2^2 = \sum_{i=1}^{n} (x_i - y_i)^2$$

(d_2 → euclidean distance)

$$= \underset{1\times n}{\begin{bmatrix} x_1 - y_1 & x_2 - y_2 & \cdots & x_n - y_n \end{bmatrix}} \underset{n\times 1}{\begin{bmatrix} x_1 - y_1 \\ x_2 - y_2 \\ \vdots \\ x_n - y_n \end{bmatrix}}$$

$$= Z^T \cdot Z = Z^T \cdot I_{n\times n} \cdot Z$$

eg. • $d_2^2((150,4),(130,3)) = 400 + 1$

$\therefore 401$ → contributes more.

but what if second feature was more important?

∴ give diff. weights to diff features.

$$\therefore \quad d_{2w}^2 = \underset{1\times n}{\begin{bmatrix} x_1-y_1 & x_2-y_2 & \cdots & x_n-y_n \end{bmatrix}} \underset{n\times n}{\begin{bmatrix} \lambda_1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & & & 0 \\ \vdots & & & & & \\ 0 & 0 & 0 & 0 & \cdots & \lambda_n \end{bmatrix}} \underset{n\times 1}{\begin{bmatrix} x_1-y_1 \\ x_2-y_2 \\ x_3-y_3 \\ \vdots \\ x_n-y_n \end{bmatrix}}$$

$$d_{2w}^2 = z' A_{n\times n} z$$

$\downarrow$

A is symmetric positive definite Matrix.

↳ advantage of A

- all eigen values > 0
- $|A| > 0$
- A^{-1} also be positive definite Matrix.

- $\Sigma \Rightarrow$ varience / covarience matrix.

or

dispersion matrix

- x_0

x_0 has almost equal distance from means of both class1 and class2

$\therefore$ we need to use second order statistics.

↳ ie variences etc.

Lecture 2
25/10/23

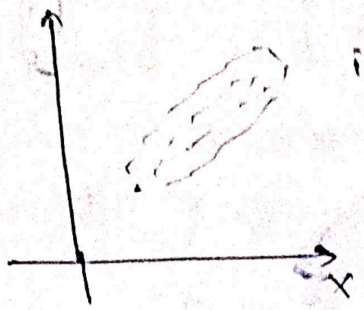
$$Var(x) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu_x)^2$$

↳ calculates spread wrt one variable/feature only.

$$Cov(x,y) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu_x)(y_i - \mu_y)$$

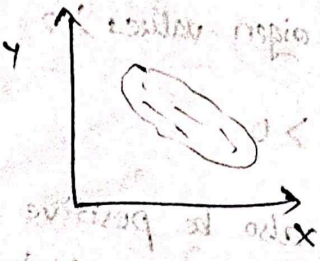
two features.

$Cov(x,y) > 0$

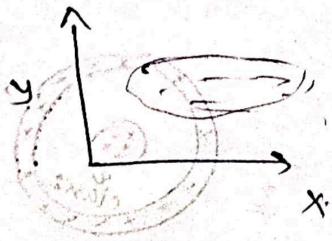


if $x \uparrow$ $y \uparrow$

$Cov(x,y) < 0$



if $x \uparrow$ $y \downarrow$



$Cov(x,y) = 0$

- Dispersion / Var-Cov Matrix

$$\Sigma = \begin{bmatrix} Cov(f_1,f_1) & \cdots & Cov(f_1,f_d) \\ \vdots & & \\ Cov(f_d,f_1) & \cdots & Cov(f_d,f_d) \end{bmatrix}_{D \times D}$$

$Cov(f_1,f_1) = Var(f_1)$

$Cov(f_1,f_2) = Cov(f_2,f_1)$

dispersion matrix is symmetric +ve definite matrix

Σ_1 ⇒ dispersion matrix for class 1.

How to calculate dispersion matrix?
easily

$$X = \begin{matrix} f_1 & f_2 \\ \end{matrix}\begin{bmatrix} 150 & 55 \\ 160 & 60 \\ 170 & 65 \end{bmatrix}$$

$$\begin{matrix} 160 & 60 \end{matrix}$$

means

$$\Sigma = \begin{matrix} f_1 \\ f_2 \end{matrix}\begin{bmatrix} 200/2 & 100/2 \\ 100/2 & 50/2 \end{bmatrix}\frac{1}{3}$$

2×2

Y = Mean subtraction of X_1

$$= \begin{bmatrix} -10 & -5 \\ 0 & 0 \\ 10 & 5 \end{bmatrix}$$

$$\Sigma = Y^T \cdot Y$$

$$= \begin{bmatrix} -10 & 0 & 10 \\ -5 & 0 & 5 \end{bmatrix}_{2\times3}\begin{bmatrix} -10 & -5 \\ 0 & 0 \\ 10 & 5 \end{bmatrix}_{3\times2}$$

$$= \begin{bmatrix} 200 & 100 \\ 100 & 50 \end{bmatrix}$$

(NOTE: y_3 not so important)

Classification

↳ supervised

- A training set (con)
- The prior prob. and probability density fns of classes are given.

- Prior probability ⇒ to be in a class.

suppose

		training data	
class 1	:	100	$P_1 = 1/6$
2	:	200	$P_2 = 1/3$
3	:	300	$P_3 = 1/2$
		600	

$P_i \mid P(\omega_i)$

∴ depending on size of class

$$\sum_i P_i = 1$$

$$1 > P_i > 0$$

- Probability density function.

Gaussian $G_D = \frac{1}{(2\pi)^{1/2} (\Sigma)^{1/2}} \exp -\frac{1}{2}\left\{(x-\mu)^T \Sigma^{-1} (x-\mu)\right\}$

follows 6σ-99.6% rule

$x_{D\times 1}$, $\mu_{D\times 1}$, $\Sigma_{D\times D}$ Matrix.

In one dimensional space:

$$G_1 = \frac{1}{2\pi\sigma} \exp -\frac{1}{2}\left\{\frac{(x-\mu)^2}{\sigma^2}\right\}$$

- pdf denoted by P_i , $P(x/w_i)$ for class i

↳ Baye's classification rule (Baye's classifier)

Let there be M classes ($M \geq 2$)

The prior probability be $P_1, P_2 \ldots P_M$ where

$$P_i > 0 \quad \text{and} \quad \sum_{i=1}^{M} P_i = 1$$

The prob. density functions.

$P_1(x)$, $P_2(x)$, $\ldots P_M(x)$ are given.

Assign new test pattern x_0 to class i

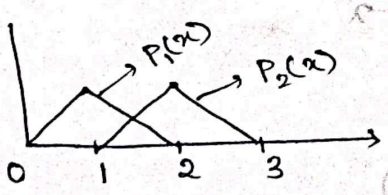
if $P_i P_i(x_0) \geq P_j P_j(x_0) \quad \forall i, j \quad i \neq j$

Resolve ties arbitarily

Example

$M = 2$

$P_1 = p \qquad P_2 = 1 - p$



$$P_1(x) = \begin{cases} x & 0 \leq x \leq 1 \\ 2-x & 1 \leq x \leq 2 \\ 0 & \text{otherwise} \end{cases}$$

$$P_2(x) = \begin{cases} x-1 & 1 < x \leq 2 \\ 3-x & 2 \leq x \leq 3 \\ 0 & \text{otherwise.} \end{cases}$$

$2p - px \quad x - 1 - px + p$

$p(2-x) = (1-p)(x-1) \quad p = x-1$

$x - p + 1$

Lecture 8
29/10/22

Threshold value of x : a value such that if $x < Th$ it belongs to class 1 and otherwise belongs to class 2.

Case I

$0 \leq x \leq 1$

$P_1 = p \quad P_2 = 1 - p$

$P_1(x) = x \quad P_2(x) = 0$

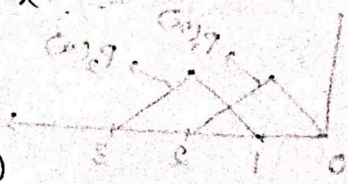
$P_1 P_1(x) > P_2 P_2(x)$ always here

$\therefore x$ be in class 1.

Case II

$2 \leq x \leq 3.$

$P_1(x) = 0 \quad P_2(x) = 3 - x$



$P_2 P_2(x) > P_1 P_1(x)$

$\therefore x$ is in class 2.

Case III $\quad 1 \leq x \leq 2.$

x be in class 1

$$P_1 P_1(x) \geq P_2 P_2(x)$$

$$\Rightarrow \quad p(2-x) \geq (1-p)(x-1)$$

$$\Rightarrow \quad 2p - px \geq x - 1 - px + p$$

$$\Rightarrow \quad 1+p \geq x \quad \Rightarrow \quad x \leq 1+p$$

$\therefore$ x be in class 1 if $x \leq 1+p$

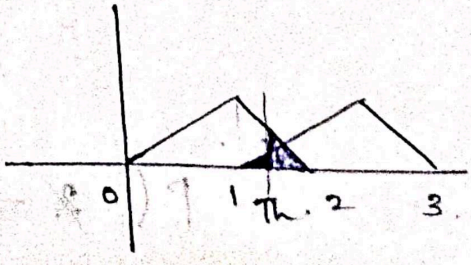
class 2 if $x \geq 1+p$

$\Omega = 0$ to 3.

↳ whole domain space

$\Omega_1 = 0$ to $1+p$

$\Omega_2 = 1+p$ to 3.



$$\int_{\Omega_1^c} P_1 P_1(x) + \int_{\Omega_2^c} P_2 P_2(x) \rightarrow \int_{1+p}^{2} (2-x)p\,dx + \int_{0}^{1+p} (x-1)(1-p)\,dx.$$

error made by this classifier.

generally

$$\text{Error} = \sum_{i=1}^{M} \int_{\Omega_i^c} P_i' P_i(x)$$

here $E = P \int_{1+p}^{2} P_1(x)\,dx + (1-p) \int_{0}^{1+p} P_2(x)\,dx$

$$E = P \int_{1+p}^{2} P_1(x)\,dx + (1-p) \int_{1}^{1+p} P_2(x)\,dx.$$

$$= P \int_{1+p}^{2} (2-x)\,dx + (1-p) \int_{1}^{1+p} (x-1)\,dx$$

$$= P\left(2(2-1-p) - \frac{1}{2}\left(4 - (1+p)^2\right)\right)$$

$$+ (1-p)\left(\frac{1}{2}\left[(1+p)^2 - 1\right] - (1+p-1)\right)$$

$$P\left(2 - 2p - 2 + \frac{(1+p)^2}{2}\right)$$

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

$E_B = \frac{p(1-p)}{2}$

Classifier 2.

$\Omega_1 = 0$ to 1.25

$\Omega_2 = 1.25$ to 3

$$E_2 = p\int_{1.25}^{2}(2-x)\,dx + (1-p)\int_{1}^{1.25}(x-1)\,dx$$

★ Bayes classifier gives minimum misclassification probability

- limitation of Bayes classification: pdf should be known

Implementing Bayes classifier on datasets

◉ Uci - repository

IRIS

⇒ 150 patterns

⇒ 4 attributes.

⇒ 3 classes.

after taking say 50% of data as training data

calculate

P_1 $P_1 \Rightarrow \mu_1, \Sigma_1$

P_2 $P_2 \Rightarrow \mu_2, \Sigma_2$ } assume Gaussian

P_3 $P_3 \Rightarrow \mu_3, \Sigma_3$

take x_0 from test set

$x_0 = \{ -, -, -, - \}$

↳ pass this to classifier

if it give highest value of $P_3 p_3(x_0)$

then 3 is the estimated class label.

to verify if it is correct compare it with actual class later.

- To measure goodness of classifier we have to follow some quantitative measure called evolution measure.

Lecture 4
30/10/23.

Minimum Distance Classifier.

~~Conditi~~ Condition

1) $P_1 = P_2 = \ldots = P_n = 1/M$

All classes are equally probable.

2) Prob. density function follows Gausian distribution

$$P_i = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} \exp\{ -1/2 (x-\mu_i)' \Sigma_i^{-1} (x-\mu_i) \}$$

3) $\Sigma_1 = \Sigma_2 = \ldots \Sigma_M = I$ ↳ identity Matrix.

x will be class i

$$P_i P_i(x) \geq P_j P_j(x) \quad \forall \; i,j \quad i \neq j$$

$\because P_1 = P_2 = \ldots 1/M$

$$P_i(x) \geq P_j(x)$$

$$\frac{1}{(2\pi)^{D/2} |I|^{1/2}} \exp\left\{ \frac{-1}{2} (x-\mu_i)' I^{-1} (x-\mu_i) \right\} \geq \frac{1}{(2\pi)^{D/2} |I|^{1/2}} \exp\{ -1/2 (x-\mu_j)' I^{-1} (x-\mu_j) \}$$

$\log_e$ in both side

$$-\frac{1}{2}(x-\mu_i)' I^{-1}(x-\mu_i) \geq -\frac{1}{2}(x-\mu_j)' I^{-1}(x-\mu_j)$$

$$(x-\mu_i)' I^{-1}(x-\mu_i) \leq (x-\mu_j)' I^{-1}(x-\mu_j)$$

$$(I^{-1} = I)$$

$$\underbrace{(x-\mu_i)'(x-\mu_i)}_{d^2(x,\mu_i)} \leq (x-\mu_j)'(x-\mu_j)$$

$d^2(x,\mu_i)$ → euclidean distance of x from μ_i

assign x in class i because distance $d(x,\mu_i)$ is minimum.

MDC says that

1) Calculate the μ_i of each class $i=1$ to M based on training data
2) Calculate the distance of test pattern to each of the class mean (i.e $d(x,\mu_i)\ \forall i$)
3) Assign x in that class where distance $d(x,\mu_i)$ is minimum.

Distance

- (1) Euclidean distance
- (2) weighted euclidian distance
- (3) L^p norm
- (4) Mahalanobis distance:

$$d_M(x,\mu_i) = (x-\mu_i)' \Sigma^{-1} (x-\mu_i)$$

→ Mean of classes may be physical or virtual Pattern.
↳ may not be a data point

→ Simple and less computational complex method.

→ If above three conditions satisfy then MDC is equivalent to Bayes classifier as well as best one to apply.

- **Confusion Matrix**

it is a matrix which represents actual and predicted class label with their count.

Actual \ Predicted	+ve	-ve
+ve	True Positive = 600	False Negative = 50
-ve	False positive = 100	True Negative = 250

Overall Accuracy = $\frac{\text{No. of Pattern connectly classified}}{\text{Total no. of Patterns.}}$

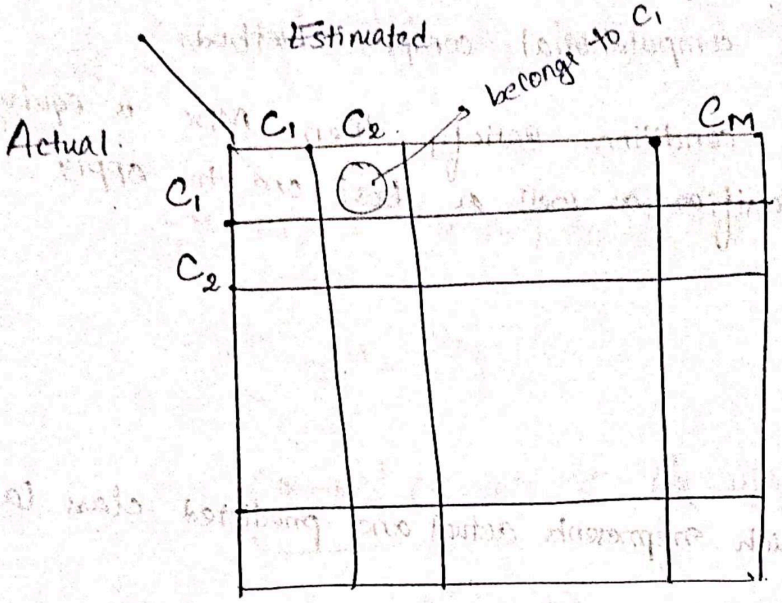
↳ as with all test patterns

$$= \frac{TP + TN}{TP + TN + FP + FN} \times 100 = \frac{850}{1000} \times 100 = 85\%$$

class wise accuracy : for +ve class $= \frac{TP}{TP + FN} = \frac{600}{650} \times 100\%$

for -ve class $= \frac{TN}{TN + FP} = \frac{250}{350} \times 100\%$

For more than 2 classes



Overall Accuracy = $\frac{\text{sum of diagonal element}}{\text{sum of all element}} \times 100\%$

Class wise accuracy : $C_i = \frac{\text{value of cell } i,i}{\text{sum of elements in } C_i \text{ row}} \times 100\%$

Lecture 5
4/11/23

K- Nearest Neighbour.

1. Training Set

$$S = \{ x_1, x_2, \ldots x_n \} \in R^D$$

and 'K' +ve integer given to you.

Task ⇒ Assign the test pattern x_0.

2. Calculate.

$$d(x_0, x_i), \quad \forall\ i = 1 \text{ to } n$$

$d(\cdot)$ ⇒ Euclidean Distance.

3. Sort $d(x_0, x_i)$ in non-decreasing order.

Choose K patterns corresponding to these distances and treat them as k nearest neighbours.

4. let k_i = be the no. of nearest neighbours belongs to class i

$$\sum_{i=1}^{c} k_i = K.$$

x_0 be in class i

if $\boxed{k_i \geq k_j}$ $\forall\ i, j \quad i \neq j$

Resolve ties arbitrarily.

Observations

1. k is +ve integer

$$k = 1, 2, 3, 4, 5 \ldots$$

Generally k is choosen as odd number.

$$k = 3, 5, 7, \ldots$$ (to avoid ties)

2. if $k=1$, call it as Nearest Neighbour.

3. KNN follows weak learning. (there is no learning, only assigning is happen)

4. Different k values give different decision.

5. It is computationally expensive

6. If a tie situation arise, resolve it arbitrarily. But if it happens frequently increase value of k.

- **Bayes' Decision to KNN Decision.**

x_0 be in class i

$$P_i P_i(x_0) \geq P_j P_j(x)$$

P_i = Prior prob. = $\frac{n_i}{n}$

$S = \{x_1, x_2, \ldots x_n\}$. $\sum_{i=1}^{c} n_i = n$

$p_i(x)$ - prob. density function of class i

Mixen density function. $p(x) = \sum_{i=1}^{c} P_i p_i(x)$

$$\text{density} = \frac{\text{Mass}}{\text{volume}}$$

Let r be the distance between x_0 and its k^{th} neighbour.

let A_n be the volume with radius r in D-dimensional domain

$$p_i(x) = \frac{k_i/n_i}{A_n}$$

- $k_i \Rightarrow$ no. of patterns in i near to x
- $n_j \Rightarrow$ no. of patterns in i

Baiye's Rule

$$P_i p_i(x) \geq P_j p_j(x)$$

$$\Rightarrow \frac{n_i}{n} \times \frac{k_i}{n_i A_n} \geq \frac{n_j}{n} \frac{k_j}{n_j A_n}$$

$$\Rightarrow \boxed{k_i \geq k_j}$$ x be in class i.

$\therefore$ KNN is a very good classifier.

Lecture 8b
8/11/23

Linear Discriminant Analysis

$M = 2$ $\quad P_1 P_1(x) \geq P_2 P_2(x)$.

(1) $p(x)$ follows Gaussian Distribution.

(2) $\Sigma_1 = \Sigma_2 = \Sigma$.

$$P_1 \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left\{\frac{-1}{2}(x-\mu_1)'\Sigma^{-1}(x-\mu_1)\right\} \geq P_2 \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left\{\frac{-1}{2}(x_2-\mu_2)' \Sigma^{-1}(x-\mu_2)\right\}$$

Log both side

$$\log P_1 - \frac{1}{2}\left\{(x-\mu_1)'\Sigma^{-1}(x-\mu_1)\right\} \geq \log P_2 - \frac{1}{2}\left\{(x-\mu_2)' \Sigma^{-1}(x-\mu_2)\right\}$$

$$\log P_1 - \frac{1}{2}\left[x'\Sigma^{-1}x - x'\Sigma^{-1}\mu_1 - \mu_1'\Sigma^{-1}x' + \mu_1'\Sigma^{-1}\mu_1\right]$$

$$\geq \log P_2 - \frac{1}{2}\left[x'\Sigma^{-1}x - x'\Sigma^{-1}\mu_2 - \mu_2'\Sigma^{-1}x' + \mu_2'\Sigma^{-1}\mu_2\right]$$

$\therefore$

$$\left(\begin{array}{l} x'\mu = \mu' x \\ x' I \mu = \mu' I x \\ x'\Sigma\mu = \mu'\Sigma x \end{array}\right)$$

$$\Rightarrow \log \frac{P_1}{P_2} \geq \frac{1}{2}\left[2x'\Sigma^{-1}(\mu_2-\mu_1) + \left(\mu_1'\Sigma^{-1}\mu_1 - \mu_2'\Sigma^{-1}\mu_2\right)\right]$$

$$\Rightarrow \underbrace{x'\underbrace{\Sigma^{-1}(\mu_2-\mu_1)}_{D\times D \quad D\times 1}}_{B} + \underbrace{\frac{1}{2}\underbrace{\left(\mu_1'\Sigma^{-1}\mu_1 - \mu_2'\Sigma^{-1}\mu_2\right)}_{\downarrow\downarrow\downarrow \quad 1\times D \quad D\times D \quad D\times 1 \;\Rightarrow\; 1\times 1} + \log\frac{P_2}{P_1}}_{C} \leq 0$$

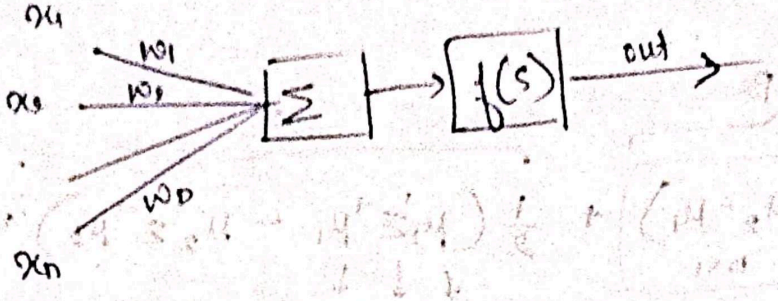
$$\Rightarrow x'B + C \leq 0$$

$$x = \underset{D\times 1}{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix}} \qquad B = \underset{D\times 1}{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_D \end{bmatrix}}$$

$$b_1x_1 + b_2x_2 + \dots + b_Dx_D + C \leq 0$$

Lecture
10/11/23

Perceptron



$$S = \sum_{i=1}^{D} w_i x_i$$

$f(\cdot)$ ⇒ let it be threshold function

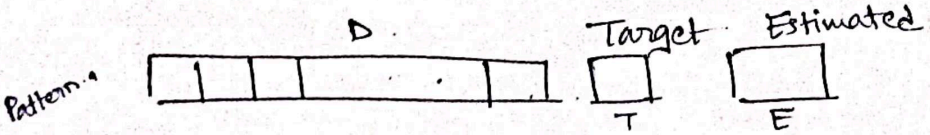
$$f\left(\Sigma w_i x_i\right) \begin{cases} > \theta & \text{output} = 1 \\ \leq \theta & \text{output } 0. \end{cases}$$

$$\Sigma w_i x_i + \theta = 0.$$

Learning ⇒ Modification of weight

initially take random weight

Training set.



Error = T − E

if $\delta \neq 0$, we need to modify weight

$$w_i(t+1) = w_i(t) + \Delta w_i(t+1)$$

$$\Delta w_i(t+1) \propto \delta$$

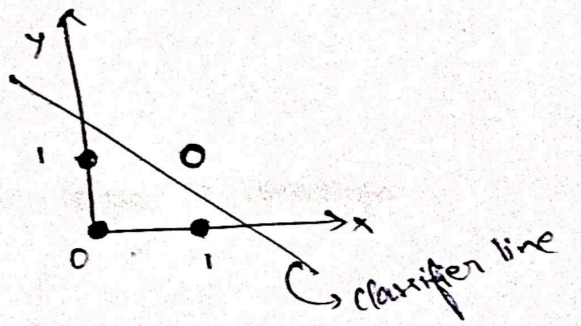
$$\propto x_i$$

$$\Delta w(t+1) = \eta \, \delta \, x_i$$

↳ learning rate, constant

Example:

Input x	Input y	Output
0	0	0
0	1	0
1	0	0
1	1	1



let $\theta = 1.5$ $\eta = 0.5$

Randomly initilize

$w_1 = 0$ $w_2 = 1$

time/ t = 1.

$x = 0, y = 0.$

I_1: $s = 0 \times 0 + 1 \times 0 = 0 < 1.5$ $\therefore$ class = 0.

NO CHANGE $\delta = 0 - T - E = 0.$

I_2 : $x=0$ $y=1$ $s = 0\times0 + 1\times1 = 1 \leq 1.5$

Class 0.

NO CHANGE, $\delta = T - E = 0 - 0 = 0$

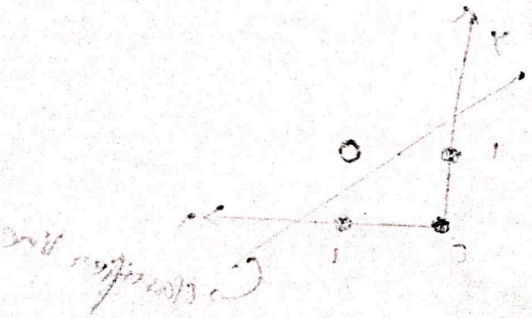
I_3 : $x=1$ $y=0$ $s = 0\times1 + 1\times0 = 0 \leq 1.5$

Class 0

NO CHANGE, $\delta = 0$

I_4 : $x=1$ $y=1$ $s = 0\times1 + 1\times1 = 1 \leq 1.5$

Class 0



0	0	$\delta = T - E$
0	1	$= 1 - 0 = 1$
0	0	1
1	1	1

$w_1 = 0 + 0.5 \times 1 \times 1 = 0.5$

$w_2 = 1 + 0.5 \times 1 \times 1 = 1.5$

t=2 $w_1 = 0.5$ $w_2 = 1.5$

& $\eta = 0.5$ $\theta = 1.5$

I_1 : —— no change ——.

I_2 : $x=0$ $y=1$ $s = 1.5 \leq 1.5$

Class = 0.

No change $\delta = 0$.

I_3: No change

I_4: $x=1$ $y=1$

$S = 0.5 \times 1 + 1.5 \times 1 = 2 > 1.5$

Class 1

No change $\delta = 1 - 1$

$= 0$

Stop. (as no change for all iterations)

$\therefore$ $\boxed{0.5x + 1.5y = 1.5}$ → can be the classification line.

- Learning Rate η

if η is small, time will be large to converge

if η is large, oscillates between original values and may not converge.

- For multidass problem ($M > 2$) need multiple perceptrons.

↳ (not multi layer perceptron)

H.W

(• logical OR and logical NAND)

Lecture
15/11/22

Linear Classifier : $w^T x + b = 0$

best classifier line as in middle of C_1 and C_2 so that least prob. of miss classification

by distance of each pattern from classifier line.

can be perpendicular to line

Support vectors : these some patterns which are on boundary of Clases., whose bar distance from classifier line, helps deciding line

Support vector Machines

Margin between two classes : distance of one ~~bounda~~ clas's boundary point with that of other.

↓

Maximize Margin.

$[x_i\ y_i]$ dataset
↓ ↓
pattern class label $(-1, 1)$

x_i : $w^T x_i + b > 0$ then x_i be in +1, in class 1

< 0 then x_i be in -1, in class 2

$= 0$ x_i is on the line.

should be

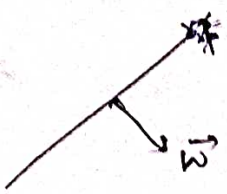
$$\boxed{y_i (w^T x_i + b) > 0}$$

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_D \end{bmatrix} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix}$$

∴ can represent them as vectors.

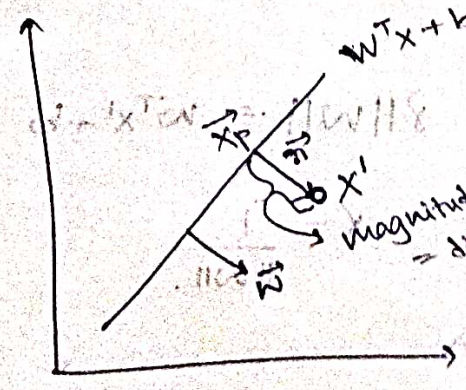
$\vec{w}$, $\vec{x}$

direction of $\vec{w}$ will be perpendicular to the line



also $\vec{w} = \|w\| \hat{w}$

$$\hat{w} = \frac{\vec{w}}{\|\vec{w}\|}$$



$\vec{x}_p + \vec{r} = \vec{x}'$

$\therefore \vec{x}_p = \vec{x}' - \vec{r}$.

$\because \vec{r} \parallel \vec{w}$

$\therefore \vec{r} = r \hat{w}$. magnitude

$$\vec{x_p} - \gamma \hat{w} = \gamma \frac{\vec{w}}{\|w\|}$$

γ = Margin = distance of x_p and x'

$$w^T x_p + b = 0$$

$$\Rightarrow \quad w^T \left[\vec{x}' - \vec{x_p} \right] + b = 0$$

$$\Rightarrow \quad w^T \left[x' - \gamma \frac{\vec{w}}{\|w\|} \right] + b = 0$$

$$\Rightarrow \quad \left[w^T x' - \gamma \frac{w^T \vec{w}}{\|w\|} \right] + b = 0$$

$$\because \quad w^T w = \|w\|^2$$

$$\Rightarrow \quad w^T x' - \gamma \|w\| + b = 0$$

$$\Rightarrow \quad w^T x' + b = \gamma \|w\|$$

time is independent with scaling operation.

$$w^T x' + b = 1.$$

$$\therefore \quad \gamma \|w\| = w^T x' + b = 1$$

$$\gamma = \frac{1}{\|w\|}.$$

similarly for x' belonging to -1 i.e class 2.

$\therefore$ ~~total~~ $w^T x' + b = -1$

$$\therefore \gamma = \frac{-1}{\|w\|}$$

$\therefore$ Total Margin $= 2\gamma = \frac{2}{\|w\|}$

Our task is to Maximize ~~total~~ Margin $= \frac{2}{\|w\|}$

$\therefore$ Minimize $f(w,b) = \frac{\|w\|}{2}$.

with

$$y_i (w^T x_i + b) > 0$$

constraint optimization problem

Lecture
17/11/23

Minimization.

$$\phi(w) = \frac{1}{2} w.w$$

constraint

$$y_i (w_i x_i + b) - 1 \geq 0$$

$\forall$ $i = 1$ to n

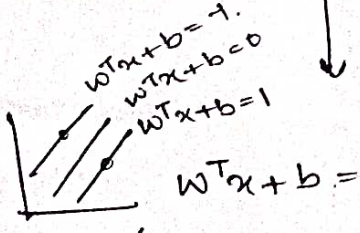
★ optimise $f(x)$ constraint $g_i(x)$.

↳ unconstraint $h(x) = f(x) - \sum \alpha_i g_i(x)$.

α_i = Lagrange Multiplier

$$L(w,b) = \frac{1}{2} w.w - \sum \alpha_i [y_i (w_i x_i + b) - 1]$$

α = Lagrange Multiplier.



$w^T x + b = \underbrace{1/-1}_{\text{for boundary support vectors}}$

$\therefore$ $y_i (w_i x_i + b) \geq 1$

↳ for x_i to not be within the margin.

$$L(w,b) = \frac{1}{2} w \cdot w - \sum \alpha_i y_i w x_i - \sum \alpha_i y_i b + \sum \alpha_i$$

$$\frac{\partial L}{\partial w} = 0 \qquad \frac{\partial L}{\partial b} = 0$$

$$\frac{\partial L}{\partial b} = -\sum \alpha_i y_i = 0$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\frac{\partial L}{\partial w} = w - \sum \alpha_i y_i x_i = 0$$

$$\boxed{w = \sum \alpha_i y_i x_i}$$

$$L = \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

$$- \sum_i \sum_j \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

$$+ \sum \alpha_i$$

$$L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

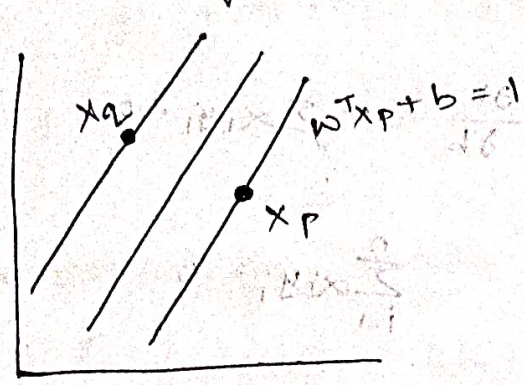
y_i, y_j, x_i, x_j are known from training data

$\therefore$ L is optimization function of α_i

Solving it we get α_i $i = 1$ to n.

$$\alpha_i \geq 0$$

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \qquad \text{①}$$



$$\frac{1}{2}\left(w^T x_p + w^T x_q\right) = b \qquad \text{②}$$

$$b = \frac{1}{2}\left[\underset{j}{\text{Max}}\left(\sum_{i|y_i=-1} \alpha_i y_i (x_i \cdot x_j)\right) + \underset{j}{\text{Min}}\left(\sum_{i|y_i=+1} \alpha_i y_i (x_i \cdot x_j)\right)\right]$$

For any test pattern X_p.

$w^T x_p + b \quad > 0 \Rightarrow$ class +1

$\quad\quad\quad\quad < 0 \Rightarrow$ class −1.

In SVM we say only support vectors are required but in eqn. of w all α_i are contributing.

$\alpha_i = 0$; corresponding patterns is non Support vector

$\alpha_i > 0$; " " is support vector.

Extraordinary cases

α_i very very large value.

$\alpha_i \gg 0$ → (noise patterns)

∴ discard them. i.e don't use them in calc. of w and b.

- Multiclass $M > 2$.

SVM 1 → separate class 1 and Rest

SVM 2 → " class 2 and Rest

.

.

.

SVM M → " class M and Rest.

- in real world. rarely linear classification occurs.

but non-linear can be transformed to linear.

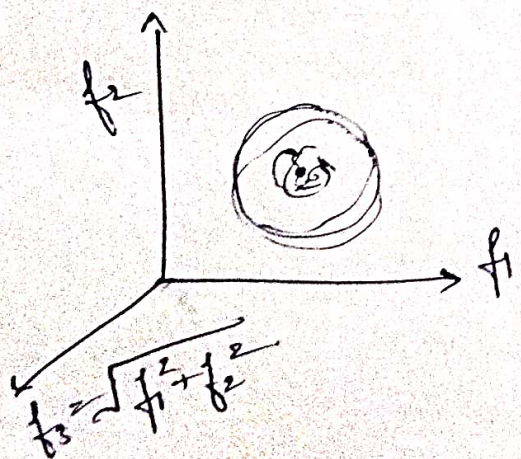
in some higher dimension.

↳ using kernel function. → very famous.

Kernel SVM

kernal function

↳ Transform the data set from lower dimensional space to higher dimensional space.



→ linear

→ polynomial

→ Radial basis function (RBF)

→ Exponential ... ...

Lecture
22/11/23

Decision Tree (based Classification)

↳ follow tree structure.

↳ Each node contain attribute (Test of attribute)

↳ No. of branches depends on the no. of discrete values of attribute.

↳ The leaf node represents the decision

CGPA	Communication	Aptitude	Prog. skill	Job. Appl
H	G	H	G	Yes
M	G	H	G	Yes
L	B	L	G	No
L	G	L	B	No
H	G	H	B	Yes
M	B	H	G	Yes
M	B	L	B	No
H	G	L	G	No
M	B	H	G	Yes
L	B	H	G	Yes
L	G	H	B	No
M	G	H	B	No
L	B	H	B	Yes
H	B	L	G	No
M	B	L	B	No
H	B	H	G	No
M	G	L	B	No
		H	B	Yes

Observations..

→ Works on discrete valued attribute set. (not continuous)

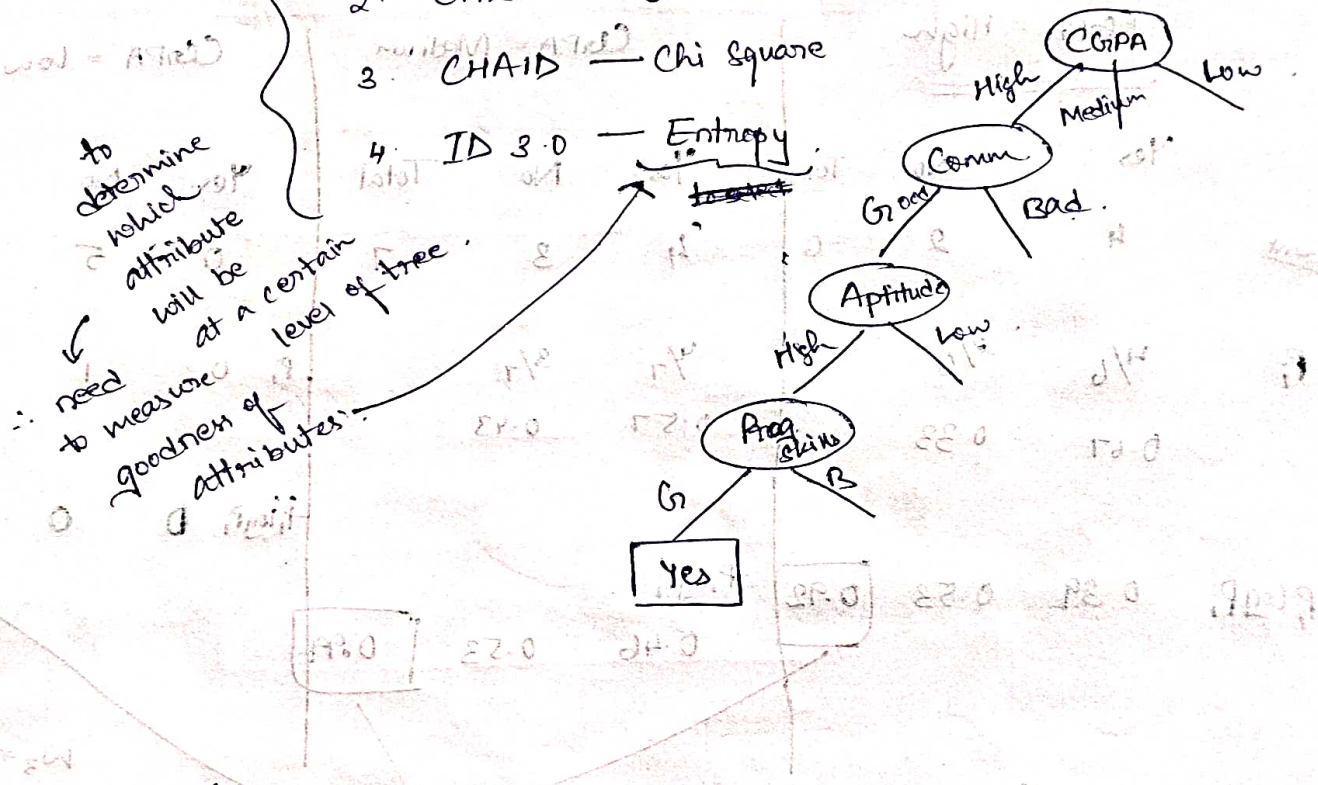
→ works on ~~non~~ numerical / non-numerical ~~se~~ attribut set

Examples of some classification techniques which follows decision tree based technique.

1. C 5.0 — Entropy
2. CART — Gini Index
3. CHAID — Chi Square
4. ID 3.0 — Entropy

to determine which attribute will be at a certain level of tree.

∴ need to measure goodness of attributes.



◉ Entropy based

↳ measures randomness present on the dataset

$$E_S = \sum_{i=1}^{c} -p_i \log p_i$$

p_i = prior probability. $= \frac{n_i}{N}$

$$E_{S,A} = \sum_{i \in \text{partition}} w_i E_i$$

↳ w.r.t attribute A

Information Gain. $(A) = E_S - E_{SA}$

Eg.

S	Yes	No	Total
count	8	10	18
p_i	$8/18$	$10/18$	
	$= 0.44$	0.56	

	Yes	No	Total
$-P_i \log P_i$	$-0.44 \log 0.44$	0.47	0.99

$= 0.52$

$E_S = 0.99$.

A = CGPA

CGPA = High

	Yes	No	Total
Count	4	2	6
P_i	4/6	2/6	
	0.67	0.33	
$-P_i \log P_i$	0.39	0.53	0.92

CGPA = Medium

	Yes	No	Total
	4	3	7
	4/7	3/7	
	0.57	0.43	
$P_i \log P_i$	0.46	0.53	0.99

CGPA = Low

	Yes	No	Total
	0	5	5
P_i	0	1	
$P_i \log P_i$	0	0	0

$W_1 = 6/18$ $\quad W_2 = 7/18$ $\quad W_3 = 5/18$

E_i i.e Position ie High, Median, Low

$$E_{SA} = W_1 E_1 + W_2 E_2 + W_3 E_3$$

$$= \frac{6}{18} \times 0.92 + \frac{7}{18} \times 0.99 + \frac{5}{18} \times 0$$

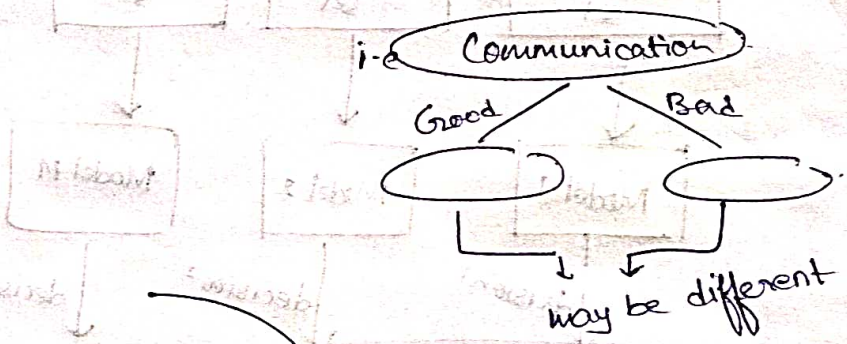
$$= 0.69$$

IG(CGPA) = 0.99 − 0.69 = 0.30.

Why IG(Comm) = ? IG(Aptitude) = ? IG(Prog. Skill) = ?

Select the attribute with Least IG as root attribute.

Now, for next layer/level of tree, again calculate for next attribute as it depends on each partition data of previous level attribute.

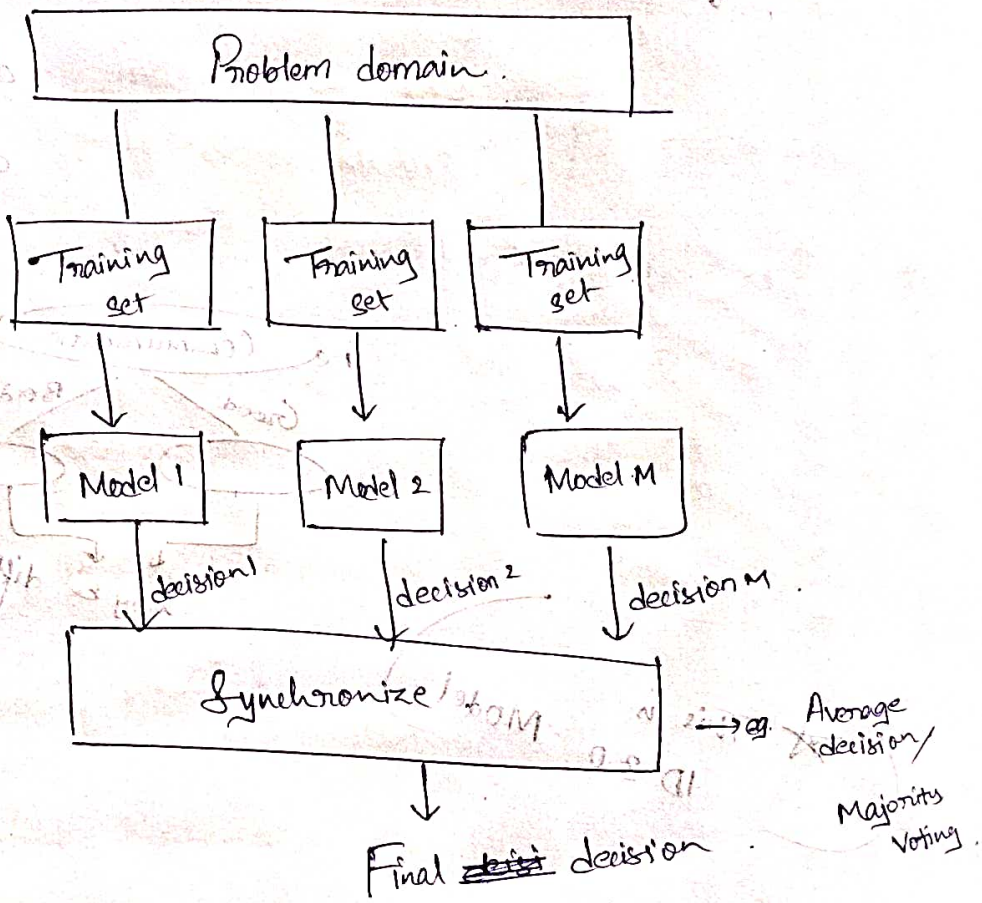


★ This is ID 3.0 Model.

Lecture
24/11/23

• Ensemble of classifier.

↳ It allows us to incorporate more than one similar and different model and then synchronise their decision to take the final decision.



Similar → means working principle is same

different → may give diff result for same pattern.

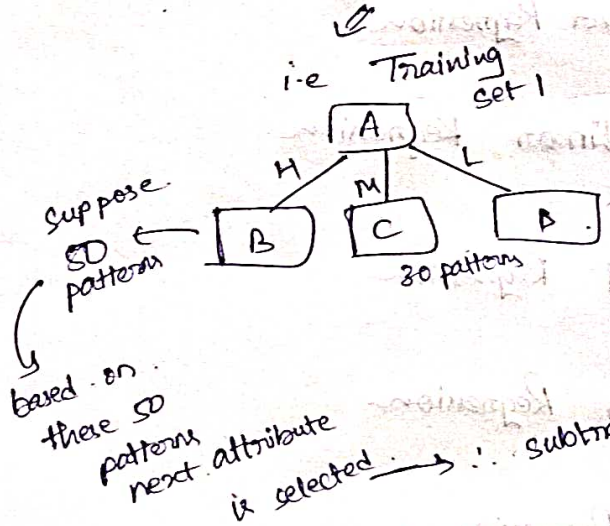
• Random Forest Classification

↳ based on Ensemble of classifier

⇓

It allows us to inconporate more than one decision tree model and then synchronise their decision to take the final decision.

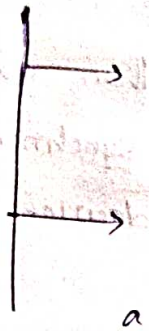
we get different decision tree not because of diff. quantifiers used but because, different training patterns are chosen randomly for each model hence each set may have diff decision tree (subtrees) ↳ which are formed base on training set patterns.



∴ subtree depends on Training set

∴ diff trees for diff training set

- **Regression**



→ Supervised.

→ Estimating / Approximating

a dependent variable (predicted ~~value~~ variable) from one or more independent variable (predictor variable) with the help of prior information.

$$y = f(x_1, x_2, x_3 \ldots x_n)$$

(1) Simple Linear Regression

(2) Multiple Linear Regression.

(3) Polynomial Regression

(4) Non-Linear Regression

(5) Logistic Regression.
↳ for classification