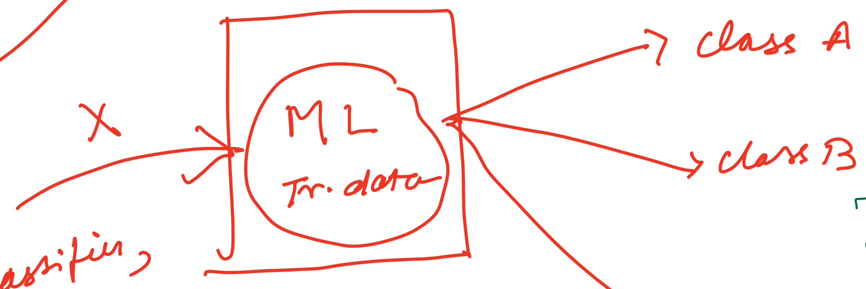
23|02|21 dec-20 - K-NN (K-nearest neighbor classifier)

→ concept of eager learners vs. lazy learners:

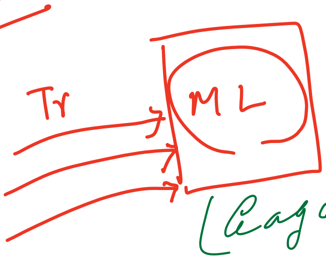
Eg: decision tree, Naive Bayes classifier, Neural Networks, SVM etc.



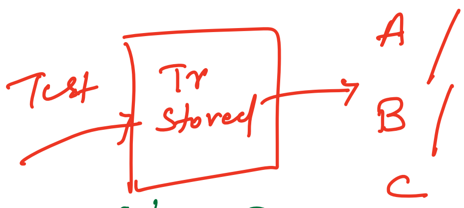
(classification process)

Eg. K-NN

PNN } Probabilistic NN



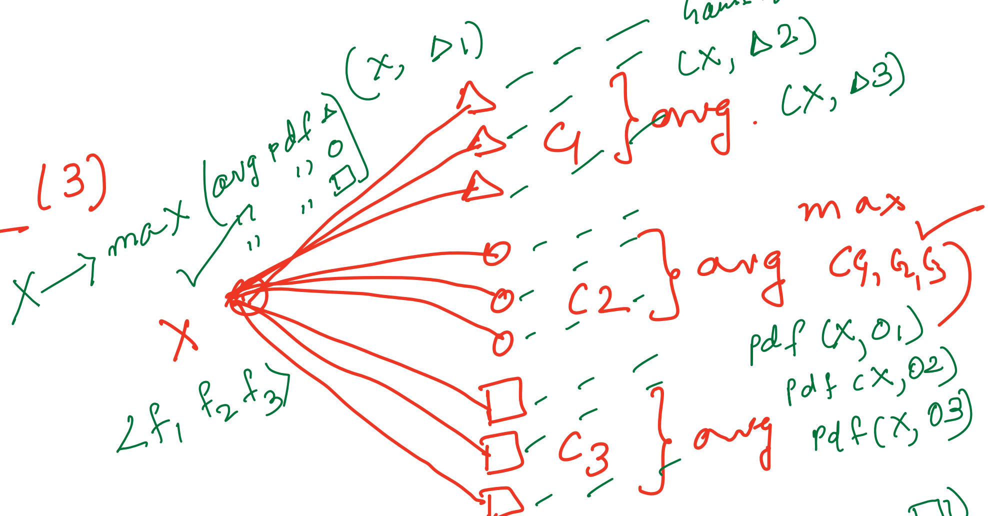
(eager)



(lazy)

→ As lazy learners store the training tuples or "instances", they are also referred as instance based learners.

PNN (3)



KNN:

early [1950s]

	SL SW PL PW
1–50	C1 Iris setosa
51–100	C2 virginica
101–150	C3 versicolor

150 × 4

Avg Pdf (C1)
Avg Pdf (C2)
Avg Pdf (C3)

X → C3

Pdf (X, □1)
Pdf (X, □2)
Pdf (X, □3)

(UCI - ML repository)

// normalize // 0 → 1

$$v' = \frac{v - \min A}{\max A - \min A}$$

$X \longrightarrow \langle f_1, f_2, f_3, f_4 \rangle \quad n = 4$

$y \quad f_1 \; f_2 \; f_3$

$v' = \dfrac{15 - 5}{20 - 5} = \dfrac{10}{15}$

closeness : is measured by Euclidean distance

$x_1 = (x_{11} \quad x_{12} \quad x_{13} \ldots x_{1n})$

$x_2 = (x_{21} \quad x_{22} \quad x_{23} \ldots x_{2n})$

$$\text{Euclidean dis}(x_1, x_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}$$

Working

Tr

1 5.2 C1
2 2.8
⋮
10 4.9
1 2.9 C2
4.6
10 3.1
1 5.8 C3
10 4.7

Test

1 C1
5
1 C2
5
1 C3
5

1 0 0
1 0 0
1 0 0
1 0 0
0 1 0
0 1 0
0 1 0
0 1 0
0 0 1
0 0 1
0 0 1
0 0 1

Test (C1, 1) $\xrightarrow[\text{Euclidean}]{\text{compute}}$ Train (1 : 30, C1, C2, C3)

K = 5

C1	C2	C3	C2	C2	C3	C1	C1	C3
2.8	2.9	4.7	3.1	4.6	4.8	4.9	5.2	5.8

3.1 4.6 4.7

2.8 → C1
2.9 → C2
3.1 → C2
4.6 → C2
4.7 → C3
4.7 → C3

C1 = 01
C3 = 01
C2 = 03

~~X~~ Majority voting fails.
06
C1 - 02
C2 = 02
C3 = 02

Test (C1, 1) → C2

C1 → 01 misclassification

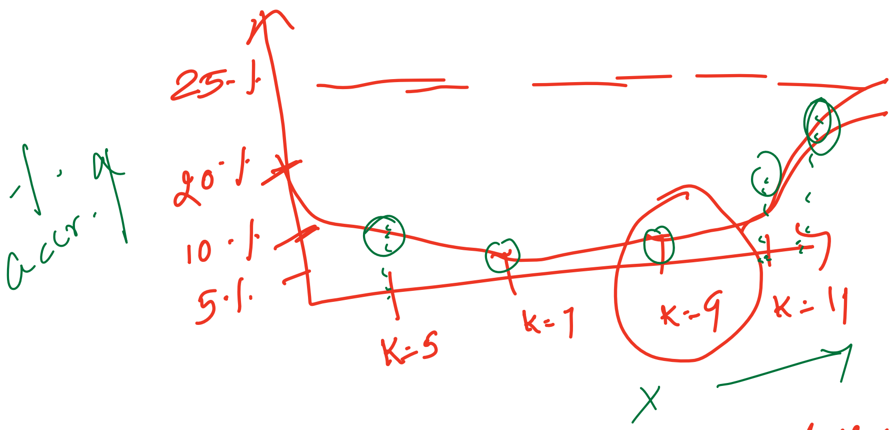
C1 → $\frac{3}{5}$ ✓ $\quad \frac{2}{5}$ ✗

C2 → $\frac{4}{5}$ ✓ $\quad \frac{1}{5}$ ✗

C3 → $5/5$ ✓ $\quad 0/5$ ✗

$\frac{3}{15} \times 100$ = 20

20 %



$\% \text{ acc} = \left(\frac{12}{15}\right) \times 100\%$

K=6 → C1 #02, C2 #02, C3 #02

K → optimal value of K is found by GA

[GA (A) KNN]

Issues :

1. How will you compute Euclidean if attributes are not numerical rather categorical ?
2. How will you handle missing values ?
3. How to determine the optimal value of K, where K denotes the no. of nearest neighbors ?

Steps of KNN

for every point in the data set :

i) calculate the distance between test tuple X and all training tuples (Euclidean distance)

ii) sort the distances in increasing order.

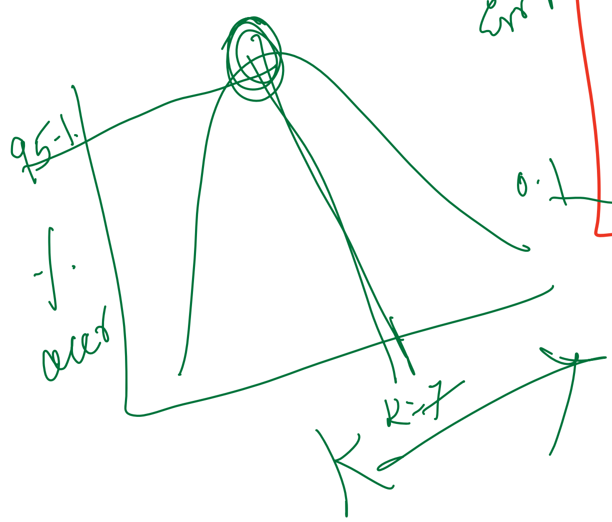
iii) consider K items as the first K nearest neighbors.

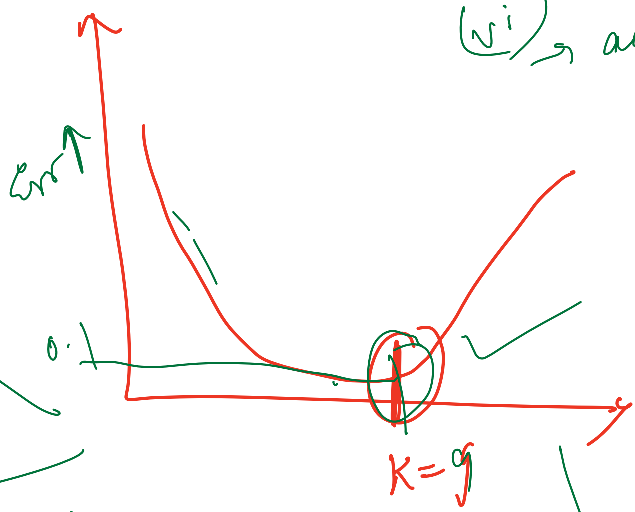
iv) use majority voting among the first K items.

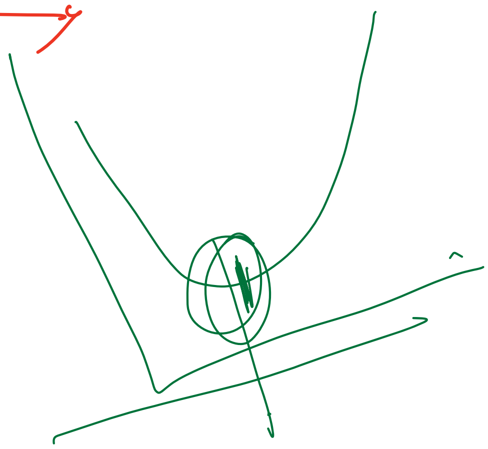
v) return the majority class as prediction for the test tuple X.

Exit.

(vi) → add classification accuracy computation in algo.





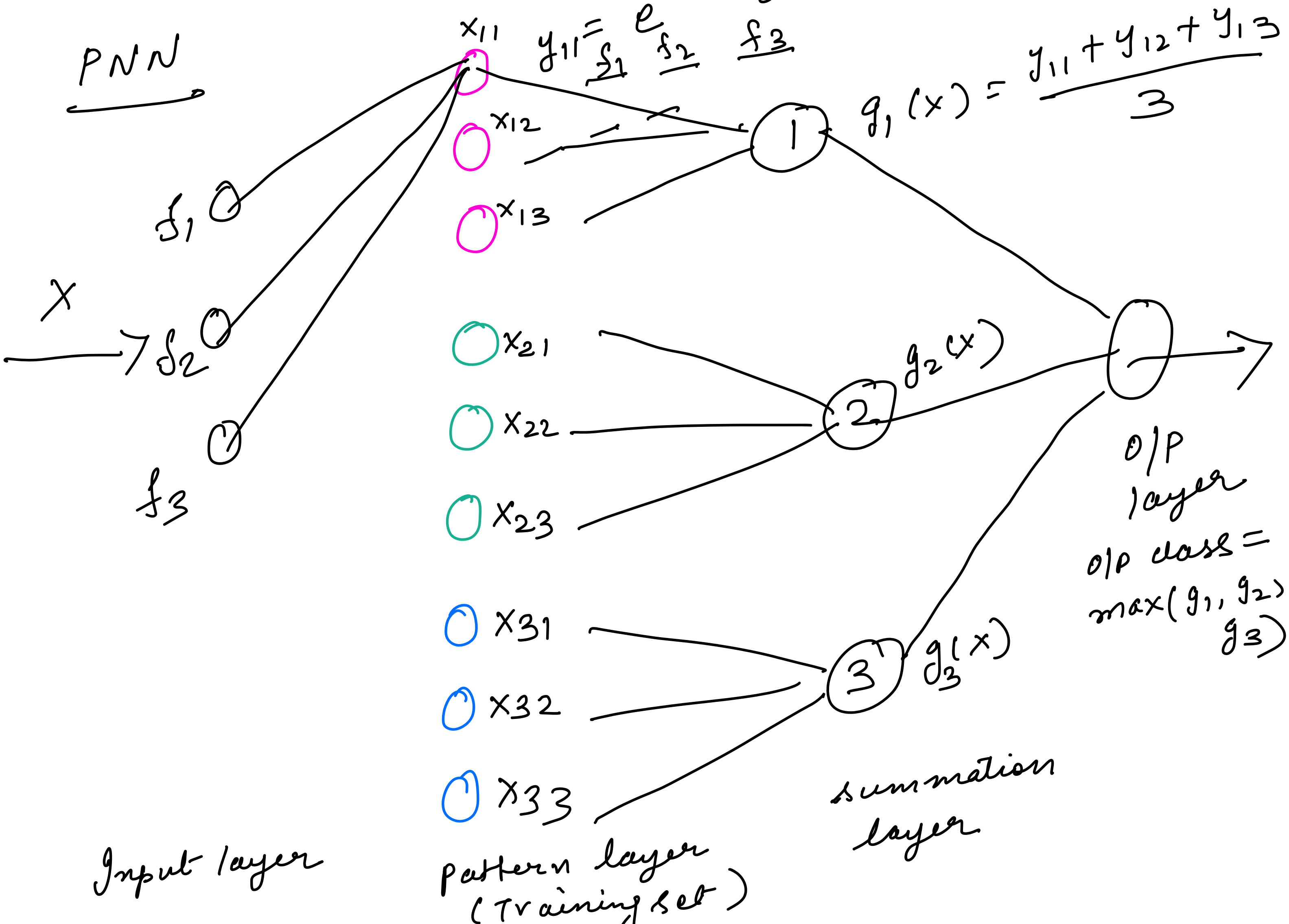


PNN

$$y_{11} = e^{-\frac{\|x - x_{11}\|^2}{\sigma^2}}$$

$$g_1(x) = \frac{y_{11} + y_{12} + y_{13}}{3}$$

o/p class = $\max(g_1, g_2, g_3)$



Test

$[0.2 \quad 0.3 \quad 0.1]$

$0 \rightarrow [0.5 \quad 0.6 \quad 0.2]$

Gaussian pdf : $e^{-\frac{\|x - x_i\|^2}{\sigma^2}}$

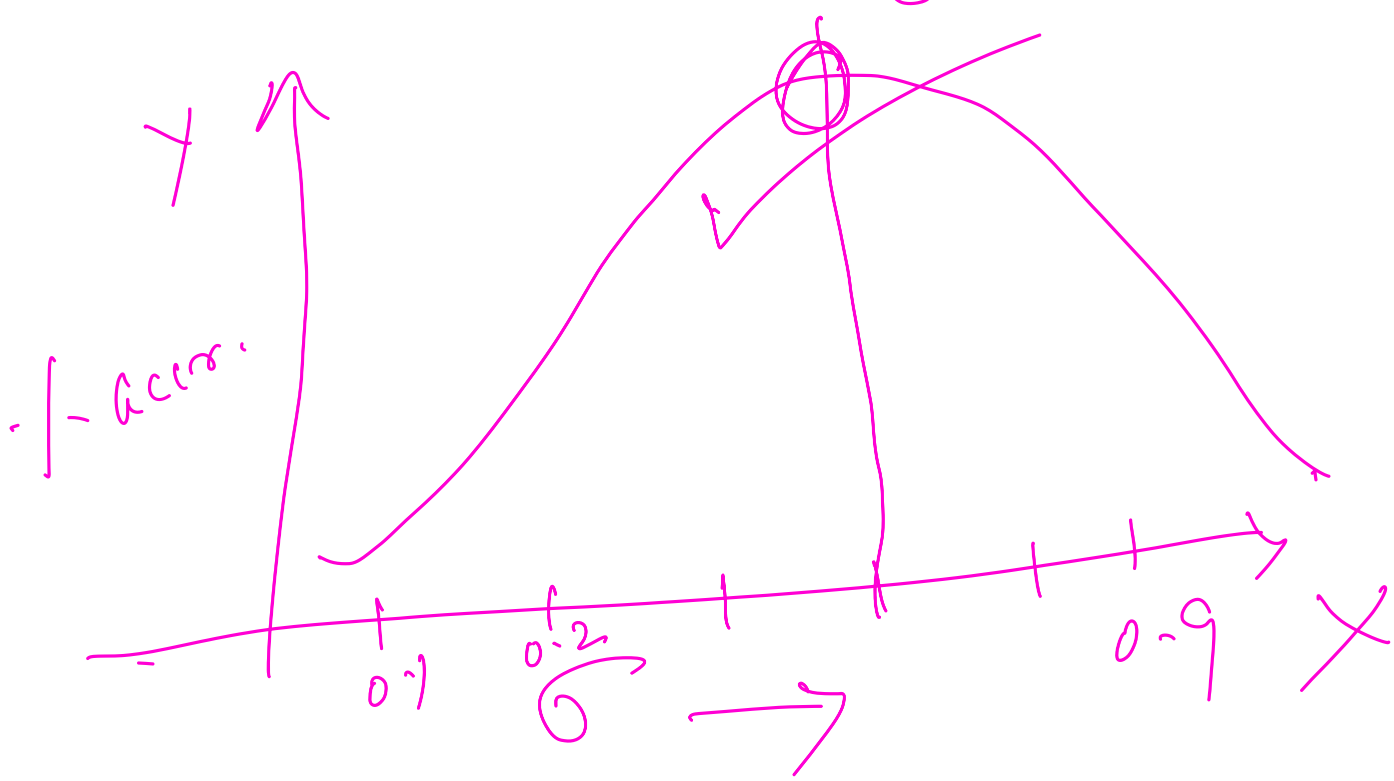
$\| \|^2 = \sqrt{(0.2 - 0.5)^2 + (0.3 - 0.6)^2 + (0.1 - 0.2)^2}$

L2 norm

Euclidean norm

$(0.2 - 0.5)^2 + (0.3 - 0.6)^2 + (0.1 - 0.2)^2$

σ^2 (σ = smoothing parameter)



$$\begin{array}{c} 1 \\ 2 \\ \vdots \\ 20 \end{array}\begin{bmatrix} \\ \\ \\ \end{bmatrix} \qquad \cancel{\frac{16}{20}} \times 100 \qquad \frac{y}{20}$$

80%

KNN

Test

Training

$$\begin{array}{c} C1 \\ \\ \\ C2 \end{array}\begin{bmatrix} 1 & f_1 & f_2 & f_3 \\ 20 & & & \\ 21 & & & \\ 40 & & & \end{bmatrix} \qquad \begin{bmatrix} c1 & & c1 \\ c30 & & \\ 31 & & c2 \\ 60 & & \end{bmatrix}$$

d1	d2	...	d30	c31	...	...	d60
C1	C2	...	c30	1	40	45	0

25

0.1	0.2	0.25	0.29	0.31	0.36	0.38	...
1	0	0	1	1	0	1	
1	0					48	
40	25	8	45	47	20		

03 0 02

7

04 1 05