Bayes rule minimizes avg prob error / Bayes rule optimum prove

→ probability of error for x, $P(\text{error}|x) = \begin{cases} P(\omega_1|x) & \text{if } \omega_2 \\ P(\omega_2|x) & \text{if } \omega_1 \end{cases}$

Bayes rules ensures $P(\text{error}|x)$ is as small as possible so that integral will be small,

$$P(\text{error}) = \int_{-\infty}^{\infty} P(\text{error}, x)\, dx$$

$$= \int_{-\infty}^{\infty} P(\text{error}|x) P(x)\, dx .$$

Thus, $P(\text{error}|x) = \min\left[P(\omega_1|x), P(\omega_2|x)\right]$

Select optimal decision (math).

Given, $P(\omega_1) = P(\omega_2) = 0.5$

$P(x|\omega_1) = N(0, 0.5)$

$P(x|\omega_2) = N(1, 0.5)$

$$\lambda = \begin{bmatrix} 0 & 0.5 \\ 1 & 0 \end{bmatrix}$$

Here, loss, λ has diagonals 0 but $\lambda_{12} \neq \lambda_{22}$

So, we will choose ω_1 if $\dfrac{P(x|\omega_1)}{P(x|\omega_2)} > \dfrac{(\lambda_{12} - \lambda_{22})\, P(\omega_2)}{(\lambda_{21} - \lambda_{11})\, P(\omega_1)}$

Calculate likelihood:

we know, $N = \dfrac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ and $N(\mu, \sigma^2)$

so for $p(x|\omega_1) = \dfrac{1}{\sqrt{2\pi(0.5)}} e^{-\frac{2(x-0)^2}{2}} = \dfrac{e^{-(x-0)^2}}{\sqrt{\pi}}$

for, $p(x|\omega_2) = \dfrac{\sqrt{2} \cdot e^{-\frac{2(x-1)^2}{2}}}{\sqrt{2\pi}} = \dfrac{e^{-(x-1)^2}}{\sqrt{\pi}}$

So, likelihood ration:

$$\frac{p(x|w_1)}{p(x|w_2)} = \frac{\cancel{\sqrt{\pi}}\, e^{-x^2+(x-1)^2}}{\cancel{\sqrt{\pi}}} = e^{-\cancel{x^2}+\cancel{x^2}-2x+1} = e^{-2x+1}.$$

Now,

$$e^{-2x+1} > \frac{(0.5-0)\times \cancel{0.5}}{(.1-0)\times \cancel{0.5}}$$

$$\Rightarrow e^{-2x+1} > \frac{1}{2}$$

$$\Rightarrow -2x+1 > \ln\left(\tfrac{1}{2}\right)$$

$$\Rightarrow -2x+1 > -0.69.$$

$$\Rightarrow -2x+1.69 > 0.$$

$$\Rightarrow 1.69 > 2x$$

$$\Rightarrow x < 0.845.$$

Thus, if $x < 0.845$ then w_1, else w_2.

proof 2D Gaussian/Normal Dfn are two 1D Gaussians if uncorrelated

Let, $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ $\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$ $\Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$

$$p(x:\mu,\Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} e^{\left[-\frac{1}{2}\begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix}^T \begin{bmatrix} \frac{1}{\sigma_1^2} & 0 \\ 0 & \frac{1}{\sigma_2^2} \end{bmatrix} \begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix}\right]}$$

Denominator : $(2\pi)^{2/2} \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}^{1/2} = 2\pi\sqrt{(\sigma_1^2\sigma_2^2)}$

$$= \sqrt{4\pi^2}\sqrt{\sigma_1^2\sigma_2^2}$$

$$= \sqrt{2\pi\sigma_1^2} \cdot \sqrt{2\pi\sigma_2^2}$$

Exponential index : $\begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix}^T \begin{bmatrix} \frac{1}{\sigma_1^2} & 0 \\ 0 & \frac{1}{\sigma_2^2} \end{bmatrix} \begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix}$

$$= \begin{bmatrix} \frac{x_1-\mu_1}{\sigma_1^2} & \frac{x_2-\mu_2}{\sigma_2^2} \end{bmatrix} \begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix}$$

$$= \frac{(x_1-\mu_1)^2}{\sigma_1^2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2}$$

Exponent : $e^{-\frac{1}{2}\left[\frac{(x_1-\mu_1)^2}{\sigma_1^2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2}\right]}$

$$= e^{-\frac{(x_1-\mu_1)^2}{2\sigma_1^2}} \cdot e^{-\frac{(x_2-\mu_2)^2}{2\sigma_2^2}}$$

So, $p(x:\mu,\Sigma) = \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(x_1-\mu_1)^2}{2\sigma_1^2}} \cdot \frac{1}{\sqrt{2\pi\sigma_2^2}} e^{-\frac{(x_2-\mu_2)^2}{2\sigma_2^2}}$

(*) Bayes decision rule interprets → w_1 if likelihood ratio exceeds a threshold value independent of observation of x.

Here

actions : α_1, α_2

states / class : w_1, w_2

loss, $\lambda_{ij} \Rightarrow \lambda(\alpha_i | w_j)$ so $\lambda(\alpha_1|w_2)$ → loss of w_1 when we decide w_2

$\lambda(\alpha_2|w_1)$ → " " w_2 " " " w_1

Conditional risk :

$$R(\alpha_i|x) = \sum_{i=1}^{c} \lambda(\alpha_i|w_j) P(w_j|x).$$

So, $R(\alpha_1|x) = \lambda_{11} P(w_1|x) + \lambda_{12} P(w_2|x)$.

$R(\alpha_2|x) = \lambda_{21} P(w_1|x) + \lambda_{22} P(w_2|x)$.

Minimum risk decision :

Decide w_1 if $R(\alpha_1|x) < R(\alpha_2|x)$.

or, $\lambda_{11} P(w_1|x) + \lambda_{12} P(w_2|x) < \lambda_{21} P(w_1|x) + \lambda_{22} P(w_2|x)$

or, $(\lambda_{21} - \lambda_{11}) P(w_1|x) > (\lambda_{12} - \lambda_{22}) P(w_2|x)$

or $\frac{P(x|w_1)}{P(x|w_2)} > \frac{(\lambda_{12} - \lambda_{22})}{(\lambda_{21} - \lambda_{11})} \frac{P(w_2)}{P(w_1)}$.

Multivariate Dis fn :- (general)

$$g_i(x) = -\frac{(x-\mu_i)^T \Sigma^{-1} (x-\mu_i)}{2} - \frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma^{-1}| + \ln P(\omega_i)$$

Case 1 $\Sigma = \sigma^2 I = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$ $\quad \Sigma^{-1} = \begin{bmatrix} \frac{1}{\sigma^2} & 0 \\ 0 & \frac{1}{\sigma^2} \end{bmatrix} = \frac{1}{\sigma^2} \cdot I$

$$(x-\mu_i)^T \Sigma^{-1} (x-\mu_i) = (x_1-\mu_i)^2 + (x_2-\mu_i)^2 = \|x-\mu_i\|^2 .$$

$$g_i(x) = \frac{\|x-\mu_i\|^2}{-2} + \ln P(\omega_i)$$

$$= \frac{(x_1-\mu_i)^2 + (x_2-\mu_i)^2}{-2} + \ln P(\omega_i) .$$

Decision boundary : $w^T(x-x_0) = 0$.

$$w = (\mu_1 - \mu_2)$$

$$x_0 = \frac{(\mu_1+\mu_2)}{2} - \frac{\sigma^2}{\|\mu_1-\mu_2\|^2} \cdot \ln\left(\frac{P(\omega_1)}{P(\omega_2)}\right)(\mu_1-\mu_2) .$$

Case 2 $\Sigma_i = \Sigma \rightarrow \begin{bmatrix} a & cov \\ cov & b \end{bmatrix}$ (use mahalanobis) . . .

এ বার

$\rightarrow D_m = (x-\mu)^T \Sigma^{-1} (x-\mu)$.

Decision boundary : ~~$w^T(x-x_0)=0$~~ . $w^T(x-x_0) = 0$

$$w^T = \Sigma^{-1}(\mu_1-\mu_2) .$$

$$x_0 = \frac{(\mu_1+\mu_2)}{2} - \frac{1}{(\mu_1-\mu_2)^T \Sigma^{-1} (\mu_1-\mu_2)} \ln\left(\frac{P(\omega_1)}{P(\omega_2)}\right)(\mu_1-\mu_2) .$$

Case 3 Σ = Arbitary

দুইটা Σ_i থাকবে।

general equation for case 3 ⊛.

We know,

$$g_i(x) = \underbrace{(x-\mu_i)^T \Sigma^{-1} (x-\mu_i)}_{-2} - \frac{1}{2}\ln|\Sigma_i| + \ln P(\omega_i) .$$

Here, $(x-\mu_i)^T \Sigma^{-1} (x-\mu_i) = x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu_i - \mu_i^T\Sigma^{-1}x + \mu_i^T\Sigma^{-1}\mu_i$

$= x^T\Sigma^{-1}x - 2\mu_i^T\Sigma^{-1}x + \mu_i^T\Sigma^{-1}\mu_i$

Now, $g_i(x) = -\frac{x^T\Sigma^{-1}x}{2} + \mu_i^T\Sigma^{-1}x - \frac{\mu_i^T\Sigma^{-1}\mu_i}{2} - \frac{1}{2}\ln|\Sigma_i| + \ln P(\omega_i)$.

Let, $W_i = -\frac{\Sigma^{-1}}{2}$

$w_i = \mu_i^T\Sigma^{-1}$

$w_{i0} = -\frac{1}{2}\mu_i^T\Sigma^{-1}\mu_i - \frac{1}{2}\ln|\Sigma_i| + \ln P(\omega_i)$.

Thus, $g_i(x) = x^T W_i x + w_i x + w_{i0}$.

Decision boundary $\rightarrow g_1(x) = g_2(x)$ ✓

(*) Unknown mean μ:

assume $p(x|\omega_j) \longrightarrow N(\mu_j, \Sigma_j)$.

let $\theta_j = [\mu_j, \Sigma_j]^T$

Now. in Dataset D,

$$p(D|\theta) = p(x_1|\theta) * p(x_2|\theta) * \cdots * p(x_n|\theta).$$

$$= \prod_{k=1}^{n} p(x_k|\theta).$$

loglikelihood,

$$L(\theta) = \ln p(D|\theta) = \sum_{k=1}^{n} \ln p(x_k|\theta).$$

or, $\nabla_\theta L = \sum_{k=1}^{n} \nabla_\theta \ln p(x_k|\theta)$.

$$-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu).$$

we know, $p(x) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} e$

for a sample point,

$$p(x_k|\mu) = \underbrace{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}_{\text{unknown } \mu} \underbrace{- \frac{1}{2} \ln\left[(2\pi)^d |\Sigma|\right]}_{\Sigma \text{ known}}$$

Here, $\nabla_\mu \ln p(x_k|\mu) = \frac{\partial}{\partial \mu}\left[-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right]$

$$= \frac{\partial}{\partial \mu}\left[-\frac{1}{2}\left(x^T \Sigma^{-1} x - 2\mu^T \Sigma^{-1} x + \mu^T \Sigma^{-1} \mu\right)\right]$$

$$= -\frac{1}{2}\left(-2\Sigma^{-1} x + \Sigma^{-1} \mu\right)$$

$$= \Sigma^{-1} x - \Sigma^{-1} \mu.$$

$$= \Sigma^{-1}(x-\mu)$$

Thus, $\sum_{k=1}^{n} \Sigma^{-1}(x_k - \mu) = 0$.

$$\Rightarrow \sum_{k=1}^{n} \Sigma^{-1} x_k - \sum_{k=1}^{n} \Sigma^{-1} \mu = 0.$$

$$\Rightarrow \sum_{k=1}^{n} x_k = \sum_{k=1}^{n} \mu.$$

$$\Rightarrow \sum_{k=1}^{n} x_k = \mu \cdot n$$

$$\Rightarrow \mu = \frac{1}{n} \sum_{k=1}^{n} x_k.$$

Unknown mean μ and covarian Σ :-

Let $\theta_1 = \mu$ and $\theta_2 = \Sigma$

~~We know, $p(x) = \frac{1}{(2\pi)^{d} |\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}$.~~

$$p(x) = \frac{1}{\sqrt{2\pi|\Sigma|}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

~~$p(x_k|\theta) = -\frac{1}{2}(x-\theta_1)^T$ ...~~

For univariate, $\ln p(x_k|\theta) = -\frac{1}{2\Sigma}(x_k - \mu)^2 - \frac{1}{2}\ln(2\pi|\Sigma|)$.

$$= -\frac{1}{2\theta_2}(x_k - \theta_1)^2 - \frac{1}{2}\ln(2\pi\,\theta_2).$$

So, $\nabla_\theta \ln p(x_k|\theta) = \begin{bmatrix} \frac{1}{\theta_2}(x_k - \theta_1) \\ \frac{1}{2\theta_2^2}(x_k - \theta_1)^2 - \frac{1}{2\theta_2} \end{bmatrix}$.

KKT wise,

$$\sum_{k=1}^{n} \frac{1}{\theta_2}(x_k - \theta_1) = 0 \quad \text{——— (1)}$$

$$\sum_{k=1}^{n} \left[\frac{1}{2\theta_2^2}(x_k - \theta_1)^2 - \frac{1}{2\theta_2}\right] = 0.$$

For ①,

$$\sum_{k=1}^{n} (x_k - \theta_1) = 0$$

$$\Rightarrow \sum_{k=1}^{n} x_k = \sum_{k=1}^{n} \theta_1$$

$$\Rightarrow \sum_{k=1}^{n} x_k = n\theta_1$$

$$\Rightarrow \theta_1 = \frac{1}{n} \sum_{k=1}^{n} x_k$$

$$\Rightarrow \hat{\mu} = \frac{1}{n} \sum_{k=1}^{n} x_k$$

For ②,

$$\sum_{k=1}^{n} \left[\frac{(x-\theta_1)^2}{2\theta_2^2} - \frac{1}{2\theta_2^2} \right] = 0$$

$$\Rightarrow \sum_{k=1}^{n} \left[\frac{(x-\theta_1)^2}{\theta_2^2} - \frac{1}{\theta_2} \right] = 0$$

$$\Rightarrow \sum_{k=1}^{n} \frac{(x-\theta_1)^2}{\theta_2^2} = \sum_{k=1}^{n} \frac{1}{\theta_2}$$

$$\Rightarrow \sum_{k=1}^{n} (x-\theta_1)^2 = \sum_{k=1}^{n} \theta_2$$

$$\Rightarrow \sum_{k=1}^{n} (x-\theta_1)^2 = n\theta_2$$

$$\Rightarrow \sum_{k=1}^{n} (x-\hat{\mu})^2 = n\hat{\theta}_2$$

$$\Rightarrow \hat{\theta}_2 = \frac{1}{n} \sum_{k=1}^{n} (x-\hat{\mu})^2$$

** Perceptron convergence theorem:

Assumptions:

1. Dataset is linearly seperable with γ margin
2. Dataset D where for all x belongs to D, $\forall x \in D$ and $\|x_i\| \leq R$. when $R > 0$.
3. If w is the soln vector that seperates data with γ margin then each data can be rescaled $\rightarrow w^*$ and $\|w^*\| = 1$.
   $\|x_i\| \leq 1, x \in D$.
4. $x = 1$
5. $x \rightarrow$ features
   $y \rightarrow$ Labels
6. $w_0 \leftarrow 0$.

Proof: w^L is weight vector wrt (x, y). for which mistake occur.

again if $y(w^{L^T} \cdot x) <= 0$.

updat: $w^{L+1} = w^L + xy$.

Now, upperlimit:

$$\|w^{L+1}\|^2 = \|w^L + xy\|^2$$

$$= (w^L + xy)^T (w^L + xy)$$

$$= \|w^L\|^2 + w^{L^T} xy + x^T w^L y + x^T x y^2.$$

$$= \|w^L\|^2 + \underbrace{2((w^L)^T x) y + \|x\|^2 y^2}_{<R}.$$

Now, $\|w^{L+1}\|^2 <= \|w^L\|^2 + R^2$

$$<= (\|w^{L-1}\|^2 + R) + R^2.$$

$$\vdots$$

$$<= w^0 + LR^2$$

So, $\|w^{L+1}\|^2 <= LR^2$ ——— (1).

Again lowerlimit:

$$(w^{L+1})^T \cdot w^* = (w^L + xy)^T \cdot w^*$$

$$= w^L w^* + ((w^*)^T x) y.$$

So, $(w^{L+1})^T w^* > = (w^L)^T w^* + \gamma$

$$> = ((w^{L-1})^T w^* + \gamma) + \gamma$$

$$\vdots$$

$$> = (w^0)^T w^* + L\gamma$$ ——— (2)

In (2),

$$L^2\gamma^2 \cdot <= \|w^{L+1}\|^2 \|w^*\|^2$$

or, $L^2\gamma^2 <= \|w^{L+1}\|^2 \quad [\because \|w^*\|^2 = 1]$ ——— (3)

In (1) and (3),

$$L^2\gamma^2 <= \|w^{L+1}\|^2 <= LR^2$$

or, $L^2\gamma^2 <= LR^2$

or, $L^2 <= R^2 <= \frac{1}{\gamma^2}$

or $L <= \frac{1}{\gamma^2}$ (Proved)

Perceptron: (Math)

Batch : Itr → $a^T x = [\ \ ][\ \ ]$ → sign $(a^T x)$ → classify করব

& if misclassified → update

$$a = a + \eta \sum y.x.$$

Fixed Incr :

Itr → $a^T x = [\ \ ][\ \ ]$ → sign $(a^T x)$ → mis

$a = a + \eta\, y.x.$

$= [\ \ ] + (\ )\times(\ )\times[\ \ ]$

(update হলে আবার x_1 – থেকে start).

ROC - AUC Math :

- Score < threshold → not spam
  Score ≥ " → spam.

Truth	Prediction
N	0.1
N	0.2
P	0.3
N	0.4
P	0.5
P	0.6
N	0.7
P	0.8
P	0.9
N	1

For Thres = 1.0,

	+	−
+	0	5
−	1	4

$FPR = \frac{FP}{TN+FP} = \frac{1}{4+1} = 0.2$

$TPR = \frac{TP}{TP+FN} = \frac{0}{0+1} = 0$.

(0.2, 0)

For Thres = 0.9

	+	−
+	1	4
−	1	4

$FPR = \frac{1}{1+4} = 0.2$.

$TPR = \frac{1}{1+4} = 0.2$

(0.2, 0.2)

For Thres = 0.8

	+	−
+	2	3
−	1	4

$FPR = \frac{1}{1+4} = 0.2$

$TPR = \frac{2}{2+3} = \frac{2}{5} = 0.4$

(0.2, 0.4).

For Thres = 0.7

	+	−
+	2	3
−	2	3

$FPR = \frac{2}{5} = 0.4$

TPR = 0.4

(0.4, 0.4)

For Thres = 0.6

	+	−
+	3	2
−	2	3

$FPR = \frac{2}{5} = 0.4$

$TPR = \frac{3}{5} = 0.6$

(0.4, 0.6)

Thres = 0.5.

	+	−
+	4	1
−	2	3

$FPR = \frac{2}{2+3} = 0.4$

$TPR = \frac{4}{5} = 0.8$

(0.4, 0.8)

Thres = 0.4

	+	−
+	4	1
−	3	2

$FPR = \frac{3}{5} = 0.6$

$TPR = \frac{4}{5}$

0.8

(0.6, 0.8)

Thres = 0.3

	+	−
+	5	0
−	3	2

$FPR = \frac{3}{5} = 0.6$

$TPR = \frac{5}{5} = 1$

(0.6, 1)

Thres = 0.2

	+	−
+	5	0
−	4	1

$FPR = \frac{4}{5} = 0.8$

$TPR = \frac{5}{5} = 1$

(0.8, 1)

Thres = 0.1

	+	−
+	5	0
−	5	0

FPR = 1

TPR = 1

(1, 1)

AUC (area under curve) $\frac{(F_2 - F_1)(T_1 + T_2)}{2}$

For (0.2, 0) (0.2, 0.2) → $\frac{(0.2-0.2)\cdot(0+0.2)}{2} = \frac{0}{2} = 0$.

For (0.2, 0.2) (0.2, 0.4) → [illegible] = 0.

For (0.2, 0.4) (0.4, 0.4) → $\frac{(0.4-0.2)(0.4+0.4)}{2} = 0.08$

For (0.4, 0.4) (0.4, 0.6) = 0

For (0.4, 0.6) (0.4, 0.8) = 0

For (0.4, 0.8) (0.6, 0.8) = $\frac{0.2 \times 1.6}{2} = 0.16$

For (0.6, 0.8) (0.6, 1) = 0

For (0.6, 1) (0.8, 1) = $\frac{0.2 \times 2}{2} = 0.2$

For (0.8, 1) (1, 1) = $\frac{0.2 \times 2}{2} = 0.2$

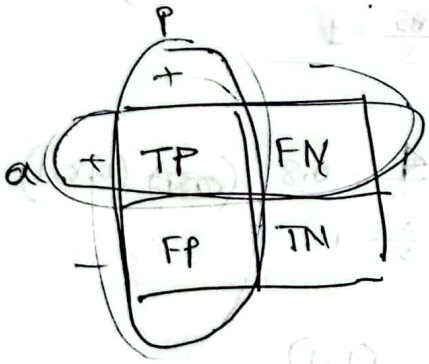
Σ Area = 0 + 0 + 0.08 + 0 + 0 + 0.16 + 0.2 + 0.2

= 0.64

good but not perfect

as AUC $\geq$ 0.5.

but ~~not~~ AUC < 1.



$FPR = \frac{FP}{FP+TN}$

$recall = \frac{TP}{TP+FN}$

$acc = \frac{TP}{TP+FN+FP+TN}$

$prec = \frac{TP}{TP+FP}$

$F1\ score = \frac{2TP}{2TP+FN+FP}$.

* K Means Cluster :-

① choose k
② initialize C.
③ C - ला distance of all points. + assign to nearest
④ update C → $\frac{\text{all points}}{N}\left(\frac{\Sigma x}{N}, \frac{\Sigma y}{N}\right)$
⑤ repeat 3 and 4,
(convergence → no data points chng cluster)

Given, $P_1(2,10)$. $P_2(2,5)$ $P_3=(8,4)$ $P_4(5,8)$ $P_5(7,5)$
$P_6(6,4)$ $P_7(1,2)$ $P_8(4,9)$ $P_9(3,6)$ $P_{10}(6,3)$.

Clusters :- $C_1(2,10)$ $C_2=(5,8)$ $C_3(1,2)$.

① K = 3.
② Clusters
③ Distance:

$C_1(2,10)$	$C_2(5,8)$	$C_3(1,2)$	
P_1 (2,10) $= \sqrt{(2-2)^2+(10-10)^2}=0$.	3.6055	8.06.	C_1
P(2,5) = 5.	4.24	3.17.	C_3.

④ $S_1 = \{P_1\}$

$S_2 = \{P_3\ P_4\ P_5\ P_6\ P_8\ P_9\ P_{10}\}$

$S_3 = \{P_2\ P_7\}$

update centroids:

$C_1 = (2, 10)$

$C_2 = \left(\frac{8+5+7+6+4+3+9}{7}, \frac{4+8+5+4+9+6+3}{7}\right)$

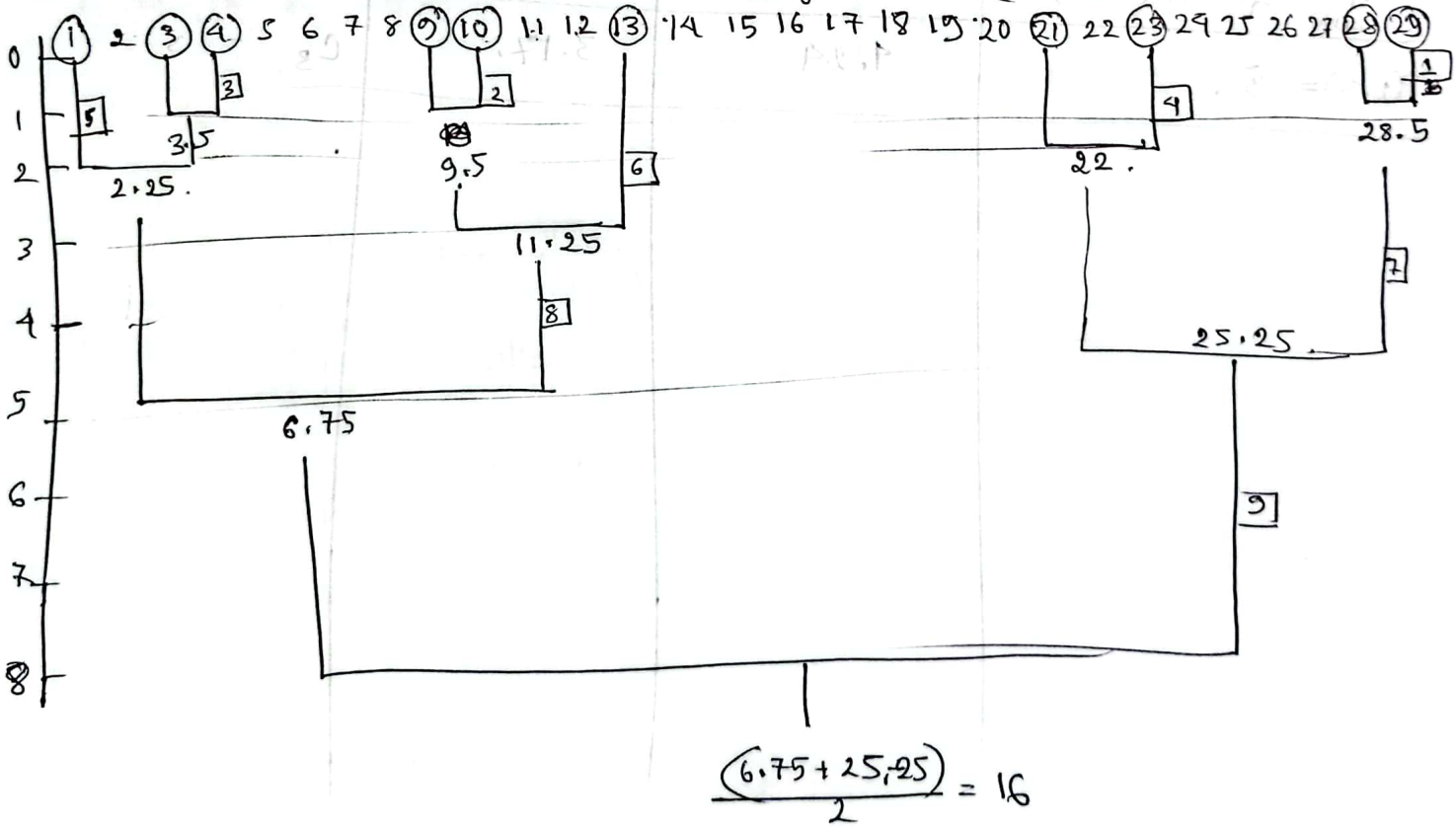
$= (6, 5.57)$

$C_3 = \left(\frac{2+1}{2}, \frac{5+2}{2}\right) = (1.5, 3.5)$.

⑤ repeat with new C.

Agglomerative Hierchical Cluster : Given, (1, 3, 4, 9, 10, 13, 21, 23, 28, 29.

largest near $(21-13) = 8 \rightsquigarrow$ Dist = 8



$\frac{(6.75 + 25.25)}{2} = 16$

PCA → ~~max~~ CA that maximizes variance
LDA → maximize CA for class seperation.

LDA → ~~5~~ steps :-

1. Class μ_0, μ_1
2. Overall μ
3. within Class scatter matrix (S_W) $\quad S_i = \sum (x-\mu_i)^T (x-\mu_i)$; $S_W = \sum S_i$
4. Between " " " (S_B) $\quad S_{B_i} = \sum (\mu_i - \mu)^T (\mu_i - \mu)$; $S_B = \sum S_{B_i}$
5. $A = S_W^{-1} S_B$.
6. $|A - \lambda I| = 0$; $\lambda = ?$ Eigenvectors, $w = ?$
7. project → XW.

Example :

x_1	x_2	Class
1	2	0
2	3	0
3	1	0
6	5	1
7	7	1
8	6	1

① Class μ_i :-

$$\mu_0 = \left(\frac{1+2+3}{3}, \frac{2+3+1}{3}\right) = (2, 2)$$

$$\mu_1 = \left(\frac{6+7+8}{3}, \frac{5+7+6}{3}\right) = (7, 6)$$

② Overall μ :

$$\mu = (4.5, 4)$$

③ S_W :

$$x_0 - \mu_0 = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 1 \end{bmatrix} - [2, 2] = \begin{bmatrix} -1 & 0 \\ 0 & 1 \\ 1 & -1 \end{bmatrix} ;$$

S_{W_0} =

$$S_{W_0} = \begin{bmatrix} -1 & 0 & 1 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$$

$$x_1 - \mu_1 = \begin{bmatrix} 6 & 5 \\ 7 & 7 \\ 8 & 6 \end{bmatrix} - [7,6] = \begin{bmatrix} -1 & -1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} ; \quad S_{w1} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} -1 & -1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

$$S_w = S_{w0} + S_{w1} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} + \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$$

④ S_B:

$$\mu_0 - \mu = \begin{bmatrix} 2 & 2 \end{bmatrix} - \begin{bmatrix} 4.5 & 4 \end{bmatrix} = \begin{bmatrix} -2.5 & -2 \end{bmatrix}$$

$n_0 = 3$

$$S_{B0} = \begin{bmatrix} -2.5 & -2 \end{bmatrix} \begin{bmatrix} -2.5 \\ -2 \end{bmatrix} = 2.25$$

$$S_{B0} = 3 \times \begin{bmatrix} -2.5 \\ -2 \end{bmatrix} \begin{bmatrix} -2.5 & -2 \end{bmatrix} = \begin{bmatrix} 6.25 & 5 \\ 5 & 4 \end{bmatrix} \times 3 = \begin{bmatrix} 18.75 & 15 \\ 15 & 12 \end{bmatrix}$$

$$\mu_1 - \mu = \begin{bmatrix} 7 \\ 6 \end{bmatrix} - \begin{bmatrix} 4.5 \\ 4 \end{bmatrix} = \begin{bmatrix} 2.5 \\ 2 \end{bmatrix}$$

$$S_{B1} = 3 \times \begin{bmatrix} 2.5 \\ 2 \end{bmatrix} \begin{bmatrix} 2.5 & 2 \end{bmatrix} = \begin{bmatrix} 6.25 & 5 \\ 5 & 4 \end{bmatrix} \times 3 = \begin{bmatrix} 18.75 & 15 \\ 15 & 12 \end{bmatrix}$$

$$S_{B_{\text{tot}}} = S_{B0} + S_{B1} = \begin{bmatrix} 37.5 & 30 \\ 30 & 24 \end{bmatrix}$$

⑤ $S_w^{-1} S_B = A$

$$S_w^{-1} = \frac{\text{adj}(S_w)}{|S_w|} = \frac{1}{16} \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix}$$

So, $S_w^{-1} S_B = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix} \begin{bmatrix} 37.5 & 30 \\ 30 & 24 \end{bmatrix} = \begin{bmatrix} 9.375 & 7.5 \\ 7.5 & 6 \end{bmatrix}$

⑥ $|A-\lambda I| = \begin{bmatrix} 9.375-\lambda & 7.5 \\ 7.5 & 6-\lambda \end{bmatrix} = 0$

$\Rightarrow (9.375-\lambda)(6-\lambda) - 56.25 = 0$

$\Rightarrow 56.25 - 15.375\lambda + \lambda^2 - 56.25 = 0$

$\Rightarrow \lambda^2 = 15.375\lambda$

$\Rightarrow \lambda = 15.375, 0$

Eigenvector, $(A-\lambda I)v = 0$

$$\begin{bmatrix} 9.375-15.375 & 7.5 \\ 7.5 & 6-15.375 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0$$

$\Rightarrow -6v_1 + 7.5v_2 = 0$ $\Rightarrow 7.5v_1 - 9.375v_2 = 0$

$\Rightarrow v_2 = 0.8v_1$ $\Rightarrow v_2 = 0.8v_1$

So, $w = \begin{bmatrix} 1 \\ 0.8 \end{bmatrix}$

⑦ $Xw = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 1 \\ 6 & 5 \\ 7 & 7 \\ 8 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ 0.8 \end{bmatrix} = \begin{bmatrix} 2.6 \\ 2 \\ 3.8 \\ 10 \\ 12.6 \\ 12.8 \end{bmatrix}$ LD1 scores.

LDA classifier:

① $x_{new} = (5, 4)$.

$z_{new} = (1 \times 5) + (0.8 \times 4) = (5 + 3.2) = 8.2$.

~~② mean, class 0: $\begin{bmatrix} 5 & 4 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ =~~

② mean, class 0: $\frac{2.6+2+3.8}{3} = 3.6$

class 1: $\frac{10+12.6+12.8}{3} = 11.8$

③ Dist:

(0) $|8.2 - 3.6| = 4.6$

(1) $|8.2 - 11.8| = 3.6$

class 1 closer ✓

PCA :-

Steps → ① Standardize → $s = \sqrt{\frac{\sum (x-\bar{x})^2}{N-1}}$

② Cov Matrix → $\frac{1}{N-1}(X^T X)$

③ Eigenvalue, $\lambda \rightsquigarrow |A - \lambda I| = 0$

Eigenvector, $(A-\lambda_1 I)(V_1) = 0 \rightsquigarrow v_1$

$(A-\lambda_2 I)(V_2) = 0 \rightsquigarrow v_2$

$\begin{bmatrix} v_1 & v_2 \\ | & | \\ | & | \end{bmatrix}$.

④ Sort Eigenvectors → $\begin{bmatrix} V_1 & V_2 \\ \cdot & \cdot \\ \cdot & \cdot \end{bmatrix}$

⑤ Projection → $X_{\text{normalized}} V = \begin{bmatrix} \quad \end{bmatrix} \times \begin{bmatrix} \quad \end{bmatrix} = \begin{bmatrix} PC1 & PC2 \\ | & | \end{bmatrix}$

ans) $PC1 = \frac{\sigma^2_{PC1}}{(\sigma^2_{PC1} + \sigma^2_{PC2})} \times 100$ ✓

$PC2 =$ ↳ same — ✓

① **Standardized:**

Given,	x_1	x_2
1	126	78
2	128	80
3	128	82
4	130	82
5	130	84
6	132	86
mean, $\bar{x} = \frac{\sum x}{N}$	129	82

$N = 1$, $\bar{x} = \frac{\sum x}{N}$

Formula : $\sigma = \sqrt{\frac{\sum (x-\bar{x})^2}{N-1}}$

$x - \bar{x}$ →

x_1 centered	x_2 centered
−3	−4
−1	−2
−1	0
1	0
1	2
3	4
$\sqrt{\frac{\sum (x-\bar{x})^2}{(N-1)}} =$	2.1	2.8

② Covariance Matrix, A :-

$$\frac{\sum X_{centered}^{T} \times centered}{(N-1)} = \frac{1}{N-1}\begin{bmatrix} -3 & -1 & +1 & 1 & 3 \\ -4 & -2 & 0 & 0 & 2 & 4 \end{bmatrix}\begin{bmatrix} -3 & -4 \\ -1 & -2 \\ -1 & 0 \\ 1 & 0 \\ 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$= \frac{1}{5}\begin{bmatrix} 22 & 28 \\ 28 & 40 \end{bmatrix} = \begin{bmatrix} 4.4 & 5.6 \\ 5.6 & 8 \end{bmatrix}$$

③ Eigen value, λ :- $|A-\lambda I| = 0$.

so, $|A - \lambda I| = \begin{bmatrix} 4.4 & 5.6 \\ 5.6 & 8 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = \begin{bmatrix} 4.4-\lambda & 5.6 \\ 5.6 & 8-\lambda \end{bmatrix}$

so, $\begin{bmatrix} 4.4-\lambda & 5.6 \\ 5.6 & 8-\lambda \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$\Rightarrow (4.4-\lambda)(8-\lambda) - (5.6)^2 = 0$

$\Rightarrow 35.2 - 4.4\lambda - 8\lambda + \lambda^2 - 31.36 = 0$.

$\Rightarrow \lambda^2 - 12.4\lambda + 3.84 = 0$.

$\Rightarrow \lambda_1 = 12.08 \qquad \lambda_2 = 0.32$.

Eigen vectors v_1 v_2 :- $(A-\lambda I)v = 0$.

For λ_1, $A - \lambda_1 I = \begin{bmatrix} 4.4-12.08 & 5.6 \\ 5.6 & 8-12.08 \end{bmatrix} = \begin{bmatrix} -7.68 & 5.6 \\ 5.6 & -4.08 \end{bmatrix}$

So, $(A - \lambda_1 I) V_1 = 0$

or, $\begin{bmatrix} -7.68 & 5.6 \\ 5.6 & -4.08 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

Here, $-7.68 v_1 + 5.6 v_2 = 0$ $\Rightarrow v_1 = \frac{5.6}{7.68} v_2$

$5.6 v_1 - 4.08 v_2 = 0$ $\Rightarrow v_1 = \frac{4.08}{5.6} v_2$

$\Rightarrow v_2 = \frac{7.68}{5.6} v_1 \approx 1.37 v_1$

$\Rightarrow v_2 = \frac{5.6}{4.08} v_1 \approx 1.37 v_1$

So, if $v_1 = 1$; $v_2 = 1.37$. Thus $v_1 = \begin{bmatrix} 1 \\ 1.37 \end{bmatrix}$

normalized, $v_1 = \begin{bmatrix} \frac{1}{\sqrt{1^2 + 1.37^2}} \\ \frac{1.37}{\sqrt{1^2 + 1.37^2}} \end{bmatrix}$

or, $v_1 = \begin{bmatrix} 0.59 \\ 0.807 \end{bmatrix}$

For, λ_2, $(A - \lambda_2 I) V_2 = 0$.

$\begin{bmatrix} 4.08 & 5.6 \\ 5.6 & 7.68 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0$

Here, $4.08 v_1 + 5.6 v_2 = 0$

$5.6 v_1 + 7.68 v_2 = 0$

So, $v_2 = -\frac{4.08}{5.6} v_1 \simeq -0.72 v_1$

$v_2 = \frac{-5.6}{7.68} v_1 \approx -0.72 v_1$

So, if $v_1 = 1$, $v_2 = -0.72$

normalize, $v_2 \begin{bmatrix} 0.81 \\ -0.59 \end{bmatrix}$

④ Sort Eigenvectors by value.

So, $v_1 = \begin{bmatrix} 0.59 \\ 0.807 \end{bmatrix}$

$v_2 = \begin{bmatrix} 0.81 \\ -0.59 \end{bmatrix}$

together, $\begin{matrix} \lambda_1 & \lambda_2 \end{matrix}$

$\begin{bmatrix} 0.59 & 0.81 \\ 0.807 & -0.59 \end{bmatrix}$

$$\sigma^2 = \frac{\sum (x-\bar{x})^2}{N-1}$$

⑤ **Projection:** $X_{centered} \cdot V = \begin{bmatrix} -3 & -4 \\ -1 & -2 \\ -1 & 0 \\ 1 & 0 \\ 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 0.59 & 0.81 \\ 0.807 & -0.59 \end{bmatrix}$

$$= \begin{bmatrix} -5 & -0.07 \\ -2.2 & 0.4 \\ -0.6 & -0.81 \\ 0.6 & 0.81 \\ 2.2 & -0.37 \\ 5 & 0.07 \end{bmatrix}$$

(PC1) (PC2)

$\sigma^2 PC1 = 12.4$.

$\sigma^2 PC2 = 0.26$.

PC1 contains $= \frac{12.4}{(0.26+12.4)} \times 100 = 97.87\%$ of variation

PCA " $= \frac{0.26}{(0.26+12.4)} \times 100 = 2.05\%$ " "

Hard Margin SVM

Minimize : $\min_{w,b} \frac{1}{2}\|w\|^2 = \frac{1}{2} w^T w$.

Constraint : $y_i(w^T x_i + b) \geq 1$

Here, $g_i(w,b) = -(y_i(w^T x_i + b) - 1) \leq 0$.

Lagrangian func :

$$\mathcal{L}(w,b,\lambda) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \lambda_i (y_i(w^T x_i + b) - 1)$$

So, according to KKT,

$$\frac{\partial \mathcal{L}}{\partial w} = \frac{2}{2} w - \sum_{i=1}^{N} y_i x_i \lambda_i = w - \sum_{i=1}^{N} \lambda_i y_i x_i$$

$$\frac{\partial \mathcal{L}}{\partial b} = -\sum_{i=1}^{N} \lambda_i y_i$$

So, $w - \sum_{i=1}^{N} \lambda_i y_i x_i = 0$

$$\Rightarrow w = \sum_{i=1}^{N} \lambda_i y_i x_i$$

$b = y_i - w^T x_i$

and $\sum_{i=1}^{N} \lambda_i y_i = 0$.

Substitute value :

$$\mathcal{L}(w,b,\lambda) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \lambda_i y_i w^T x_i - \sum_{i=1}^{N} \lambda_i y_i b + \sum_{i=1}^{N} \lambda_i$$

$$= \frac{1}{2} w^T w - w^T w + \sum_{i=1}^{N} \lambda_i$$

$$= -\frac{1}{2} w^T w + \sum_{i=1}^{N} \lambda_i$$

$$= \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j y_i y_j x_i x_j$$

Soft Margin

Minimize : $\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$
w, b

Constraint : $y_i(w^T x_i + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$.

So, $L(w, b, \xi, \lambda, \mu) = \frac{1}{2}w^T w + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\mu_i\xi_i - \sum_{i=1}^{n}\lambda_i(y_i(w^T x_i + b) - 1 + \xi_i)$

$= \frac{1}{2}w^T w + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\mu_i\xi_i - \sum_{i=1}^{n}\lambda_i y_i w^T x_i - \sum_{i=1}^{n}\lambda_i y_i b$.

$+ \sum_{i=1}^{n}\lambda_i - \sum_{i=1}^{n}\xi_i\lambda_i$

Partial derivatives:

$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{n}\lambda_i y_i x_i$.

$\frac{\partial L}{\partial b} = -\sum_{i=1}^{n}\lambda_i y_i$

$\frac{\partial L}{\partial \xi_i} = C - \sum_{i=1}^{n}\mu_i - \sum_{i=1}^{n}\lambda_i = C - \mu_i - \lambda_i = 0$.

KKT,

$\sum_{i=1}^{n}\lambda_i[y_i(w^T x_i + b) - 1 + \xi_i] = 0$.

$\mu_i\xi_i = 0$

So, $\mu_i \geq 0$ and $\lambda_i \geq 0$.

Here, $C - \mu_i - \lambda_i = 0$.

or, $\mu_i = C - \lambda_i \geq 0$.

If $\xi_i \geq 0$, $\mu_i = 0$.

or, $C - \lambda_i \neq 0$.

or, $C = \lambda_i$

If $\xi_i = 0$; $\mu_i \geq 0$.

or, $C - \lambda_i \geq 0$.

or, $C \geq \lambda_i$

Eliminate ξ_i,

we get lagrangian fn

$L(w, b, \lambda)$

তারপর আগের মতই করতে হবে।

* Kernel Method :

To avoid ~~HD~~ projecting data directly to HD space — compute dot-prod

$$\phi : x \rightarrow \phi(x) .$$

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j) .$$

Find w:

We know,

$$\max_{\lambda} \left[\sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i=1}^{N} \lambda_i \lambda_j y_i y_j x_i x_j \right] , \quad w = \sum_{i=1}^{n} \lambda_i y_i x_i$$

$$b = y_i - w^T x_i$$

Find w:

$$g(x_{new}) = \left(\sum_{i=1}^{n} \lambda_i y_i \phi(x_i) \right) \phi(x_{new}) + b .$$

$$w = \sum_{i=1}^{n} \lambda_i y_i K(x_i, x_{new}) + b .$$

For b,

$$b = y_i - w^T x_i$$

$$= y_i - \sum_{i=1}^{n} \lambda_i y_i K(x_i, x_{new}) .$$

(*) Euclidean distance → A=(3,4) B=(0,0) . in 2D space

↳ $\sqrt{(3-0)^2+(4-0)^2} = \sqrt{9+16} = 5$

(*) Normalized Euclidean distance :-

যদি বড় value থাকে . $d(x,y) = \sqrt{\Sigma \cdot \frac{(x-y)^2}{(max-min)^2}}$ → $s \cdot \sigma^2$

A = (2, 1000) B = (4, 800).

so, $d(A,B) = \sqrt{\frac{(4-2)^2}{(4-2)^2} + \frac{(1000-800)^2}{200^2}} = \ldots$

(*) Mahalanobis Distance : dist btw point → distribution — taking correlation account

generalized Euclidean dist

$$d_M(x,\mu) = \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)}$$

Euclidean	vs	Mahalanobis
① dist btw 2 points in 2D space		① generalized euclidean dist btw point and its distribution
② Doesn't considers correlation		② Considers correlation
③ Treats feature independent		③ accounts feature correlation
④ assumes spherical cluster.		④ adapts to elliptical cluster

Both criterian fn to solve linear inequalities $a^t y > 0$

Batch Perceptron (vs)	Fixed Increment Single Sample perc
1. Computes cumulative correction (sum of misclassified samples)	1. Computes misclassified samples iteratively and updates then and there if found.
2. updates after cumu... correctn	2. updates when misclassified sample found
3. Computationally expensive	3. cheaper
4. All samples are checked in each iteration	4. Each sample is handled in each iteration and keeps iterating if misclassified
5. less noisy	5. More noisy
6. Converges earlier than fixed increment.	6. Convergence needs more iteration as each step is small

```
1. Initialize ← a(0), η←1
2. while (True):
3.     mis←0.
4.     for all sample y in dataset
5.         if label(a^t y)≤0
6.             mis←mis+1
7.             store y in np.array.
8.         endif
9.     end for
10.    a(new)←a(old)+Σ label.sample
11.    if mis==0.
12.        break;
13.    end if
14. end while
```

```
1. Initialize ← a(0), η←1.
2. while (True):
3.     mis←0
4.     for all sample is y in D.
5.         if label(a^t y)≤0
6.             mis ← mis+1
7.             a(new)←a(old)+η.label.
8.         end if
9.     end for
10.    if mis==0
11.        break
12.    end if
13. end while
```

Calculation of w in kernel:-

Dual form:

$$\max_{\lambda} \left[\sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{i=1}^{N} \lambda_i \lambda_j y_i y_j K(x_i, x_j) \right]$$

~~g(new) =~~ .

$$g(x_{new}) = \left(\sum_{i=1}^{N} \lambda_i y_i \phi(x_i) \right) \phi(x_{new}) + b .$$

$$= \sum_{i=1}^{N} \lambda_i y_i K(x_i, x_{new}) + b .$$

$$w = \sum \lambda_i y_i \phi(x_i)$$

$$b = y_i - w^T \phi(x_i) .$$

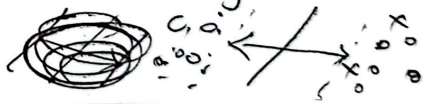
for b,

$$y_i \left(\sum_{i=1}^{n} \lambda_i y_i \phi(x_i) \cdot + b \right) = 1$$

$$\Rightarrow \sum_{i=1}^{n} \lambda_i y_i K(x_i, x_j) + b = y_i$$

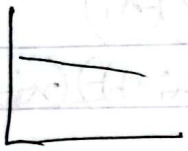
$$\Rightarrow b = y_i - \sum_{i=1}^{n} \lambda_i y_i K(x_i, x_j) .$$

Hard ~~if g(x_new) > 0 ...~~	Soft
(1) assumes linearly separable data	(1) messy data
(2) No misclassification allowed	(2) might face some.
(3) widest-possible margin gap searches.	(3) adds penalty for mis points (C) keeping largest margin regu.
(4) Rigid, strict	(4) flexible, balances.
(5) fails if noise/overlapping found as can't tolerate mistakes.	(5) gives penalties, tries to regularize using C, balances.

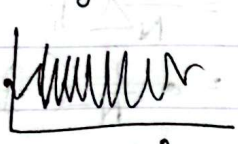


LR :

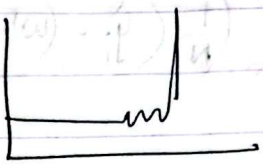
small



Larger(বেশি)



wild fluctuate



drastical increase

noisy ← Stochastic GD → batch size = 1 per itro. (random chose example).

min batch SGD → 1 < bat size < N.

↳ no. of sample

small batch → SGD-র মতো

Bang " → full GD (batch)

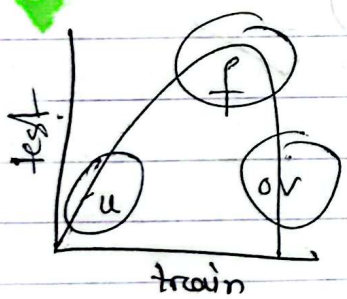
* Full batch → 1000 sample, 20 epoch, 20 update (once per epoch).
* SGD → " " " 20,000 " w, b.
* minibatch SGD → " " " , 100 batch size,

batches per epoch = $\frac{1000}{100} = 10$.

thus update · 10 × 20 = 200.

High LR → pushes w away from 0 (overfit)
reg → brings w near 0 (underfit)

* Regularization - যে, role is avoid overfitting

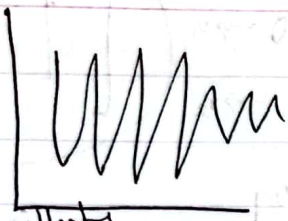


$\min(\text{loss fn} + \lambda(\text{regu}))$

$\text{regu}(\text{log loss}) = \text{log loss} + \lambda\Omega(w)$

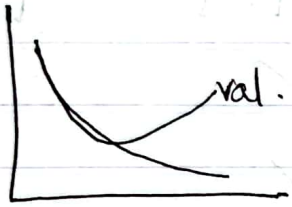
λ small → penalty minimal
fit train data — large w
overfits.
noise capturing
complex model

large λ →
penalty — significant
reduces w, loss closer to 0
don't fit noise
reduces complexity.
over simplifies
underfits.

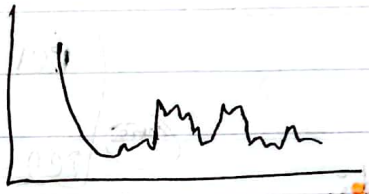


oscillate
LR high
↳ clip Grad, optimize

sharp rise.
LR too high, grad exploded
data noisy, model instable.



sharp val loss rise
Overfit, data leak.
Concept drift

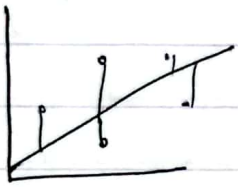


Chaotic repeated loss
LR high.
overfit (no dropout / on regu).
numeric instability.
unstable batch num

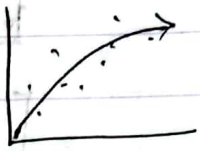
(14)

* Algo do not outperform each other always.
* compare — error rate on test data instead train.

Bias : diff btw predictions and correct value

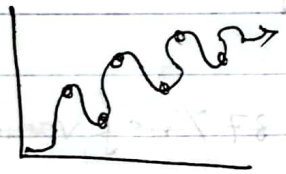


High bias
oversimplifies.
underfitted

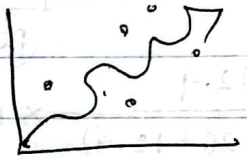


Low Bias.
overfits

Variance : measures how much pred chgs if on diff dataset



High variance
overfitted



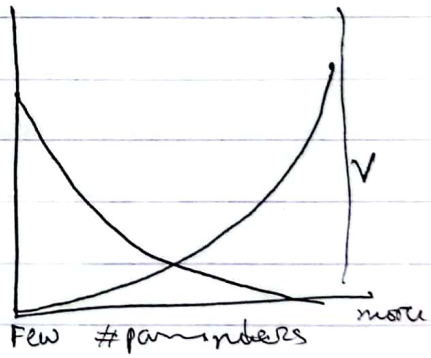
low variance.
better on test.

Bias-Variance tradeoff :-

HB - LV : model simple.
underfitted, both sets - loss high.

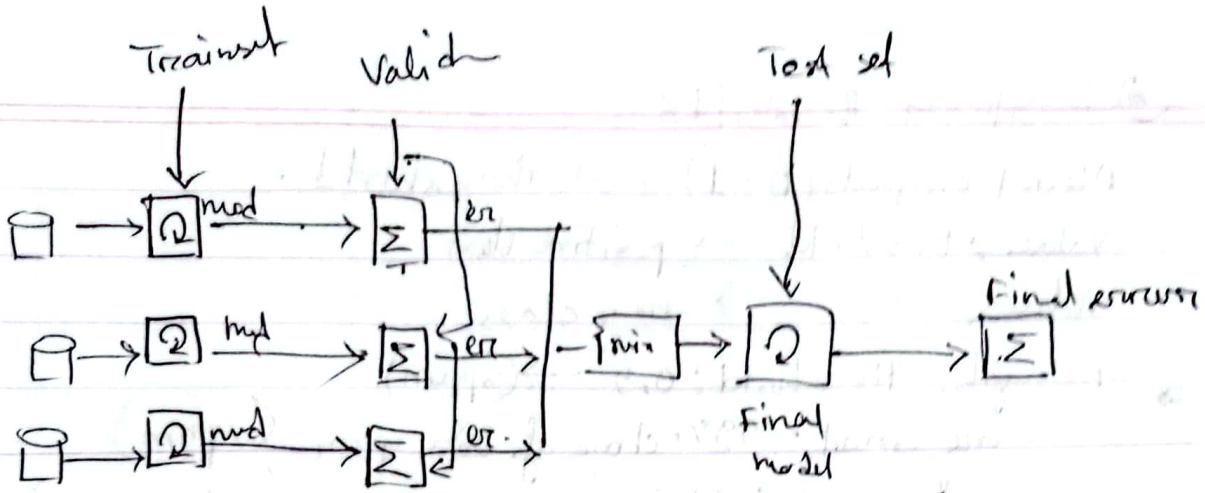
LB - HV : model complex.
overfitted

Balance B-V : moderate complexity.

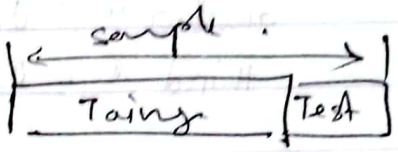


Cross validation :

train 1. Train set : optimizes parameters (w, b)
evaluate 2. Validation set : " hyperparam (no. of NN, length of trees)
assess 3. Testing " : final assessment



CV Methods:



① Holdout- : simpler, test-train
↓
learns → performs on test set.

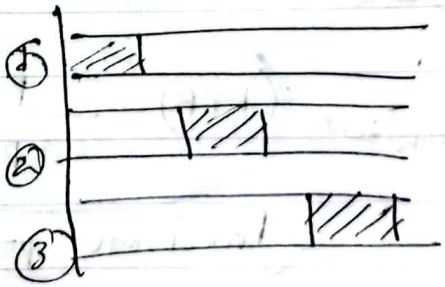
② Random Subsample:

* repeats holdout randomly selecting example.
* error rate from test sample

③ k-Fold CV:

* k equal sized subsets.
* one subset — test
* rest " — train
* repeated k times on each subset.
* Total error = Avg of all errors.
* like random subsample but.
  all examples are used for both train test

k=3.



Classification Threshold:

Model outputs (0-1), set threshold.
value > threshold → positive class.
value < " → neg class.

Example: threshold: 0.5 → (spam)
one email = 99% chance of being spam (spam).
another " = 0.44% " " " " (not spam).
third " = 0.51% " " " " (spam)

* **Confusion Matrix:** summarizes performance comparing actual vs predicted labels.

A \ P	+	−
+	TP	FN
−	FP	TN

TP → correct prediction → actually yes – pred yes
TN → " " → " no – pred no.
FP → incorrect " → act yes – pred no.
FN → " " → act no – pred yes

Evaluation: ~~Accuracy~~ = $\frac{\text{correct classify}}{\text{total classify}} = \frac{TP+TN}{TP+TN+FP+FN}$

perfect model – so, FP = FN = 0.

pre(zero)
recall(zero)
TP | FN
FP | TN
FPR(zero)

Recall (TPR): $\frac{\text{correct- actual positive}}{\text{all actual pos}} = \frac{TP}{TP+FN}$

$(FPR) = \frac{FP}{FP+TN}$

Precision = $\frac{\text{~~positive~~ correct positives}}{\text{total classified positive}} = \frac{TP}{TP+FP}$

F1 score: avg of pre+recall (harmonic mean)

$\frac{2TP}{2TP+FN+FP}$.