*. Normal Density & Discriminant function *.

$$g_i(x) = \ln p(x|\omega_i) + \ln p(\omega_i)$$

$$p(x|\omega_i) = \frac{1}{(2\pi)^{d/2} |\Sigma_i|^{1/2}} \exp\left[\frac{-1}{2}(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)\right]$$

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

independent of classes.

case II :- $\Sigma_i = \Sigma$

(*) Discriminant function :-

$P(\omega_i|x) > P(\omega_j|x)$; ω_1 and ω_2 are classes.

(-) Bayes minimum error classifier :-

In this we compute $P(\omega_i|x)$ for all the classes. and choose the class which have highesht value of $P(\omega_i|x)$.

(-) Bayes minimum risk classifier :-

In this we compute $R(\alpha_i|x)$ risk for all classes.

$$R(\alpha_i|x) = \sum_{\forall \omega_j} \lambda(\alpha_i|\omega_j)\, P(\omega_j|x)$$

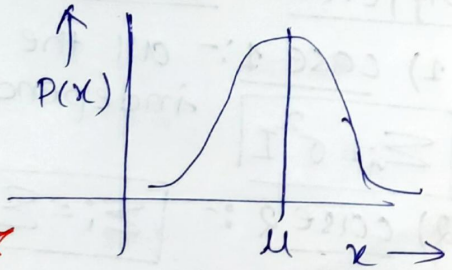
$$\lambda(\alpha_i|\omega_j) = 0 \quad i=j$$
$$= 1 \quad i \neq j$$

choose the class which have highest value of $-R(\alpha_i|x)$.

$$\Rightarrow \quad g_i(x) = P(\omega_i|x)$$
$$g_i(x) = P(x|\omega_i).P(\omega_i)$$
$$g_i(x) = \ln P(x|\omega_i) + \ln P(\omega_i)$$

(*) Discriminant function under multivariate normal distribution :-

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)}$$



Normal Distribution.

Let us assume that our feature vector is in d dimension.

$x \rightarrow d$ dimension

Then probability distribution,

$$P(x) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left[\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right]$$

$$P(x|w_i) = \frac{1}{(2\pi)^{d/2} |\Sigma_i|^{1/2}} \exp\left[\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)\right]$$

we know that,

$$g_i(x) = \ln P(x|w_i) + \ln(P(w_i))$$

$$\therefore g_i(x) = \ln\left(\frac{1}{(2\pi)^{d/2} |\Sigma_i|^{1/2}} \exp\left[\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)\right]\right) + \ln P(w_i)$$

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

independent of class so, it can be ignored.

$$\therefore g_i(x) = -\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i) - \frac{1}{2}\ln|\Sigma_i| + \ln P(w_i)$$

Different cases :-

1) case 1 :- all the components are statistically independent. which means variance = 0.

$\Sigma_i = \sigma^2 I$

2) case 2 :- $\Sigma_i = \Sigma$

3) case 3 :- Σ_i = arbitary

For case I:

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

$\Sigma_i = \sigma^2 I$.

$$\therefore g_i(x) = \frac{1}{2\sigma^2}(x-\mu_i)^T(x-\mu_i) + \ln P(\omega_i)$$

$$\therefore g_i(x) = \frac{1}{2\sigma^2}\left[x^Tx - 2\mu_i^Tx + \mu_i^T\mu_i\right] + \ln P(\omega_i)$$

class independent.

$$\therefore g_i(x) = \frac{\mu_i^Tx}{\sigma^2} - \frac{\mu_i^T\mu_i}{2\sigma^2} + \ln P(\omega_i)$$

$$\boxed{g_i(x) \simeq w_i^Tx + w_{i0}}$$ → Linear

$$w_i = \frac{\mu_i^T}{\sigma^2} \; ; \; w_{i0} = \frac{1}{2\sigma^2}\mu_i^T\mu_i + \ln P(\omega_i)$$

For boundary,

$$\boxed{g(x) = g_i(x) - g_j(x) = 0}$$

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

$$g_j(x) = \frac{-1}{2\sigma^2}(x-\mu_j)^T(x-\mu_j) + \ln P(\omega_j)$$

$$g(x) = \frac{\mu_i^Tx}{\sigma^2} - \frac{\mu_i^T\mu_i}{2\sigma^2} + \ln P(\omega_i) - \frac{\mu_j^Tx}{\sigma^2} + \frac{\mu_j^T\mu_j}{2\sigma^2} - \ln P(\omega_j)$$

$$g(x) = \frac{(\mu_i-\mu_j)^Tx}{\sigma^2} - \frac{\mu_i^T\mu_i}{2\sigma^2} + \frac{\mu_j^T\mu_j}{2\sigma^2} + \ln\frac{P(\omega_i)}{P(\omega_j)} = 0$$

$$\boxed{g(x) = (\mu_i-\mu_j)^Tx - \frac{1}{2}\left(\mu_i^T\mu_i - \mu_j^T\mu_j + \sigma^2\ln\frac{P(\omega_i)}{P(\omega_j)}\right) = 0}$$

$$g(x) \simeq w^T(x-x_0) = 0$$

$$w = (\mu_i-\mu_j) \;\&\; x_0 = \frac{1}{2}(\mu_i+\mu_j) - \frac{\sigma^2}{\|\mu_i-\mu_j\|^2}\ln\left(\frac{P(\omega_i)}{P(\omega_j)}\right)(\mu_i-\mu_j)$$

Physical significance of $\Sigma_i = \sigma^2 I$:-

1) Diff^{nt} component of feature vector are stasitically independent.
and every component has same variance.

2) spread → circle in 2D
→ sphere in 3D
→ Hypersphere in ND.

(-) Example :-

$\omega_1 \Rightarrow \begin{pmatrix}12\\4\end{pmatrix}\begin{pmatrix}12\\8\end{pmatrix}\begin{pmatrix}10\\6\end{pmatrix}\begin{pmatrix}14\\6\end{pmatrix}$ $\omega_2 \Rightarrow \begin{pmatrix}9\\10\end{pmatrix}\begin{pmatrix}9\\14\end{pmatrix}\begin{pmatrix}7\\12\end{pmatrix}\begin{pmatrix}11\\12\end{pmatrix}$

$$\mu_1 = \left[\begin{pmatrix}12\\4\end{pmatrix}+\begin{pmatrix}12\\8\end{pmatrix}+\begin{pmatrix}10\\6\end{pmatrix}+\begin{pmatrix}14\\6\end{pmatrix}\right] \div \begin{pmatrix}1\\4\end{pmatrix}$$

$$\mu_1 = \frac{\begin{pmatrix}48\\24\end{pmatrix}}{\begin{pmatrix}1\\4\end{pmatrix}} = \boxed{\begin{pmatrix}12\\6\end{pmatrix}} \qquad \boxed{\mu_2 = \begin{pmatrix}9\\12\end{pmatrix}}$$

$$(x_1-\mu_1)(x_1-\mu_1)^T = \begin{pmatrix}0\\-2\end{pmatrix}(0\ -2) = \begin{pmatrix}0&0\\0&4\end{pmatrix} = M_1$$

$$(x_2-\mu_2)(x_2-\mu_2)^T = \begin{pmatrix}0\\2\end{pmatrix}(0\ 2) = \begin{pmatrix}0&0\\0&4\end{pmatrix} = M_2$$

$$(x_3-\mu_3)(x_3-\mu_3)^T = \begin{pmatrix}-2\\0\end{pmatrix}(-2\ 0) = \begin{pmatrix}4&0\\0&0\end{pmatrix} = M_3$$

$$(x_4-\mu_4)(x_4-\mu_4)^T = \begin{pmatrix}2\\0\end{pmatrix}(2\ 0) = \begin{pmatrix}4&0\\0&0\end{pmatrix} = M_4$$

$$\Sigma_1 = \frac{1}{4}[M_1+M_2+M_3+M_4]$$

$$\Sigma_1 = \frac{1}{4}\begin{bmatrix}8&0\\0&8\end{bmatrix}$$

$$\boxed{\Sigma_1 = \begin{bmatrix}2&0\\0&2\end{bmatrix}}$$

$(x_1-\mu_2)(x_1-\mu_2)^T = \begin{pmatrix}0\\-2\end{pmatrix}(0\ -2) = \begin{pmatrix}0 & 0\\0 & 4\end{pmatrix} = m_1'$

$(x_2-\mu_2)(x_2-\mu_2)^T = \begin{pmatrix}0\\2\end{pmatrix}(0\ 2) = \begin{pmatrix}0 & 0\\0 & 4\end{pmatrix} = m_2'$

$(x_3-\mu_2)(x_3-\mu_2)^T = \begin{pmatrix}-2\\0\end{pmatrix}(-2\ 0) = \begin{pmatrix}4 & 0\\0 & 0\end{pmatrix} = m_3'$

$(x_4-\mu_2)(x_4-\mu_2)^T = \begin{pmatrix}2\\0\end{pmatrix}(2\ 0) = \begin{pmatrix}4 & 0\\0 & 0\end{pmatrix} = m_4'$

$\Sigma_2 = \frac{1}{4}[m_1'+m_2'+m_3'+m_4']$

$\Sigma_2 = \frac{1}{4}\begin{pmatrix}8 & 0\\0 & 8\end{pmatrix} = \begin{pmatrix}2 & 0\\0 & 2\end{pmatrix} = \boxed{2I}$ ✓

$\boxed{\Sigma_1 = \Sigma_2 = 2I} \simeq \sigma^2 I \qquad \therefore \boxed{\sigma = \sqrt{2}}$

Here, $\boxed{P(\omega_1) = P(\omega_2)}$ $[\because \#\omega_1 = \#\omega_2]$

$\therefore x_0 = \frac{1}{2}(\mu_1+\mu_2)$ and $w = \mu_1-\mu_2$.

$\therefore x_0 = \frac{1}{2}\left(\begin{pmatrix}12\\6\end{pmatrix}+\begin{pmatrix}9\\12\end{pmatrix}\right) = \frac{1}{2}\begin{pmatrix}21\\18\end{pmatrix}$ $\quad w = \begin{pmatrix}3\\-6\end{pmatrix}$

$\therefore \quad w^T(x-x_0) = 0$

$(3\ -6)\begin{pmatrix}x_1\\x_2\end{pmatrix} - \frac{(3\ -6)}{2}\begin{pmatrix}21\\18\end{pmatrix} = 0$

$3x_1 - 6x_2 - \frac{1}{2}(63-108) = 0$

$3x_1 - 6x_2 + \frac{45}{2} = 0$

$6x_1 - 12x_2 + 45 = 0$

$\boxed{3x_1 - 4x_2 + 15 = 0}$ ← Decision Boundary

For case II :- $\Sigma_i = \Sigma \neq \sigma^2 I$.

→ The component of feature vector may be statistically independent but not equal. or same.

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

class independent.

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

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

$$\therefore \boxed{g_i(x) \cong w_i^T x + w_{i0}}$$

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

For decision boundary,

$$g(x) = g_1(x) - g_2(x)$$

$$g_1(x) = \mu_1^T\Sigma^{-1}x - \frac{\mu_1^T\Sigma^{-1}\mu_1}{2} + \ln P(\omega_1)$$

$$g_2(x) = \mu_2^T\Sigma^{-1}x - \frac{\mu_2^T\Sigma^{-1}\mu_2}{2} + \ln P(\omega_2)$$

$$g(x) = \mu_1^T\Sigma^{-1}x - \mu_2^T\Sigma^{-1}x - \frac{\mu_1^T\Sigma^{-1}\mu_1}{2} + \frac{\mu_2^T\Sigma^{-1}\mu_2}{2} + \ln\frac{P(\omega_1)}{P(\omega_2)}$$

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

$$\boxed{g(x) \cong w^T(x-x_0) = 0}$$ → Linear

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

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

(-) Example :-

$w_1 \Rightarrow \begin{pmatrix}6\\2\end{pmatrix}\begin{pmatrix}9\\3\end{pmatrix}\begin{pmatrix}7\\5\end{pmatrix}\begin{pmatrix}10\\6\end{pmatrix}$

$w_2 \Rightarrow \begin{pmatrix}6\\11\end{pmatrix}\begin{pmatrix}9\\12\end{pmatrix}\begin{pmatrix}7\\14\end{pmatrix}\begin{pmatrix}10\\15\end{pmatrix}$

$\mu_1 \Rightarrow \begin{pmatrix}32\\16\end{pmatrix}/4 = \begin{pmatrix}8\\4\end{pmatrix}$

$\mu_2 = \begin{pmatrix}32\\52\end{pmatrix} \neq 4 = \begin{pmatrix}8\\13\end{pmatrix}$

$\boxed{\mu_1 = \begin{pmatrix}8\\4\end{pmatrix}}$

$\boxed{\mu_2 = \begin{pmatrix}8\\13\end{pmatrix}}$

$(x_1-\mu_1)(x_1-\mu_1)^T = \begin{pmatrix}-2\\-2\end{pmatrix}(-2\ -2) = \begin{pmatrix}4&4\\4&4\end{pmatrix} = m_1$

$(x_2-\mu_1)(x_2-\mu_1)^T = \begin{pmatrix}1\\-1\end{pmatrix}(1\ -1) = \begin{pmatrix}1&-1\\-1&1\end{pmatrix} = m_2$

$(x_3-\mu_1)(x_3-\mu_1)^T = \begin{pmatrix}-1\\1\end{pmatrix}(-1\ 1) = \begin{pmatrix}1&-1\\-1&1\end{pmatrix} = m_3$

$(x_4-\mu_1)(x_4-\mu_1)^T = \begin{pmatrix}2\\2\end{pmatrix}(2\ 2) = \begin{pmatrix}4&4\\4&4\end{pmatrix} = m_4$

$\Sigma_1 = \frac{1}{4}[m_1+m_2+m_3+m_4] = \frac{1}{4}\begin{bmatrix}10&6\\6&10\end{bmatrix} = \begin{bmatrix}5/2&3/2\\3/2&5/2\end{bmatrix}$

$\Sigma_1^{-1} = \frac{1}{8}\begin{bmatrix}5&-3\\-3&5\end{bmatrix}$

$(x_1-\mu_2)(x_1-\mu_2)^T = \begin{pmatrix}-2\\-2\end{pmatrix}(-2\ -2) = \begin{pmatrix}4&4\\4&4\end{pmatrix} = m_1'$

$(x_2-\mu_2)(x_2-\mu_2)^T = \begin{pmatrix}1\\-1\end{pmatrix}(1\ -1) = \begin{pmatrix}1&-1\\-1&1\end{pmatrix} = m_2'$

$(x_3-\mu_2)(x_3-\mu_2)^T = \begin{pmatrix}-1\\1\end{pmatrix}(-1\ 1) = \begin{pmatrix}1&-1\\-1&1\end{pmatrix} = m_3'$

$(x_4-\mu_2)(x_4-\mu_2)^T = \begin{pmatrix}2\\2\end{pmatrix}(2\ 2) = \begin{pmatrix}4&4\\4&4\end{pmatrix} = m_4'$

$\Sigma_2 = \begin{bmatrix}5/2&3/2\\3/2&5/2\end{bmatrix} \qquad \Sigma_2^{-1} = \begin{bmatrix}5/8&-3/8\\-3/8&5/8\end{bmatrix}$

$\therefore \boxed{\Sigma_1 = \Sigma_2 = \Sigma \neq \sigma^2 I}$

$$\therefore \quad x_0 = \left(\frac{\mu_1 + \mu_2}{2}\right) \qquad \left[\because \text{since } P(\omega_1) = P(\omega_2) \;\therefore \ln \frac{P(\omega_1)}{P(\omega_2)} = 0\right]$$

$$\therefore \quad w = \Sigma^{-1}(\mu_1 - \mu_2)$$

$$w = \begin{bmatrix} 5/8 & -3/8 \\ -3/8 & 5/8 \end{bmatrix} \begin{bmatrix} 0 \\ 9 \end{bmatrix} = \begin{bmatrix} -27/8 \\ 45/8 \end{bmatrix}$$

$$\therefore \quad g(x) = w^T x - w^T x_0$$

$$g(x) = \begin{bmatrix} -\frac{27}{8} & \frac{45}{8} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} -27/8 & 45/8 \end{bmatrix} \begin{bmatrix} 16/2 \\ 17/2 \end{bmatrix}$$

$$g(x) = -\frac{27x_1}{8} + \frac{45x_2}{8} - \left[\frac{-27 \times 16}{16} + \frac{45 \times 17}{8 \times 2}\right]$$

$$g(x) = \frac{-27x_1 + 45x_2}{8} - \frac{333}{16} = 0$$

$$\boxed{g(x) \quad 54x_1 - 90x_2 + 333 = 0} \quad \#$$

Decision Boundary

For case III :- Σ_i = arbitary

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

we can't ignore

Reason:- Different for Different classes.

$$\therefore g_i(x) = -\frac{1}{2}\left(x^T\Sigma_i^{-1}x - 2\mu_i^T\Sigma_i^{-1}x + \mu_i^T\Sigma_i^{-1}\mu_i\right) - \frac{1}{2}\ln|\Sigma_i| + \ln P(\omega_i)$$

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

$$\boxed{g_i(x) \simeq x^TA_ix + B_i^Tx + C_i} \rightarrow \text{Quadratic}$$

$$A_i = -\frac{1}{2}\Sigma_i^{-1} \qquad B_i = \mu_i^T\Sigma_i^{-1}$$

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

For Decision boundary,

$$g(x) = g_1(x) - g_2(x) = 0$$

$$g_1(x) = -\frac{1}{2}x^T\Sigma_1^{-1}x + \mu_1^T\Sigma_1^{-1}x - \frac{1}{2}\mu_1^T\Sigma_1^{-1}\mu_1 - \frac{1}{2}\ln|\Sigma_1| + \ln P(\omega_1)$$

$$g_2(x) = -\frac{1}{2}x^T\Sigma_2^{-1}x + \mu_2^T\Sigma_2^{-1}x - \frac{1}{2}\mu_2^T\Sigma_2^{-1}\mu_2 - \frac{1}{2}\ln|\Sigma_2| + \ln P(\omega_2)$$

$$\therefore g(x) = -\frac{1}{2}x^T(\Sigma_1^{-1} - \Sigma_2^{-1})x + (\mu_1^T\Sigma_1^{-1} - \mu_2^T\Sigma^{-1})x - \frac{1}{2}\ln\frac{|\Sigma_1|}{|\Sigma_2|} - \frac{1}{2}\mu_1^T\Sigma_1^{-1}\mu_1 + \frac{\mu_2^T\Sigma^{-1}\mu_2}{2} + \ln\frac{P(\omega_1)}{P(\omega_2)}$$

$$\therefore \boxed{g(x) \simeq x^TA_ix + B_i^Tx + C_i}$$

$$A_i = -\frac{1}{2}(\Sigma_1^{-1} - \Sigma_2^{-1})$$

$$B_i = (\mu_1^T\Sigma_1^{-1} - \mu_2^T\Sigma_2^{-1})$$

$$C_i = -\frac{1}{2}\mu_1^T\Sigma_1^{-1}\mu_1 + \frac{1}{2}\mu_2^T\Sigma^{-1}\mu_2 - \frac{1}{2}\ln\frac{|\Sigma_1|}{|\Sigma_2|} + \ln\frac{P(\omega_1)}{P(\omega_2)}.$$

(-) Example

$w_1 \Rightarrow \begin{pmatrix}5\\4\end{pmatrix}\begin{pmatrix}9\\2\end{pmatrix}\begin{pmatrix}9\\6\end{pmatrix}\begin{pmatrix}13\\6\end{pmatrix}$ $\quad w_2 \Rightarrow \begin{pmatrix}9\\10\end{pmatrix}\begin{pmatrix}9\\14\end{pmatrix}\begin{pmatrix}7\\12\end{pmatrix}\begin{pmatrix}11\\12\end{pmatrix}$

$\mu_1 = \begin{pmatrix}36\\18\end{pmatrix}/4$

$\boxed{\mu_1 = \begin{pmatrix}9\\9/2\end{pmatrix}}$

$\mu_2 = \begin{pmatrix}36\\48\end{pmatrix}/4$

$\boxed{\mu_2 = \begin{pmatrix}9\\12\end{pmatrix}}$

$(x_1-\mu_1)(x_1-\mu_1)^T$

$= \begin{pmatrix}4\\-1/2\end{pmatrix}(4 \quad -1/2)$

$= \begin{pmatrix}16 & -2\\-2 & 1/4\end{pmatrix} = m_1$

$(x_2-\mu_1)(x_2-\mu_1)^T$

$= \begin{pmatrix}0\\-5/2\end{pmatrix}(0 \quad -5/2)$

$= \begin{pmatrix}0 & 0\\0 & 25/4\end{pmatrix} = m_2$

$(x_3-\mu_1)(x_3-\mu_1)^T$

$= \begin{pmatrix}0\\3/2\end{pmatrix}(0 \quad 3/2)$

$= \begin{pmatrix}0 & 0\\0 & 9/4\end{pmatrix} = m_3$

$(x_4-\mu_1)(x_4-\mu_1)^T$

$= \begin{pmatrix}4\\3/2\end{pmatrix}(4 \quad 3/2)$

$= \begin{pmatrix}16 & 6\\6 & 9/4\end{pmatrix} = m_4$

$\Sigma_1 = \frac{1}{4}[m_1+m_2+m_3+m_4]$

$= \frac{1}{4}\begin{bmatrix}82 & 4\\4 & 11\end{bmatrix} = \begin{bmatrix}8 & 1\\1 & 11/4\end{bmatrix}$

$\Sigma_1^{-1} = \frac{1}{21}\begin{bmatrix}11/4 & -1\\-1 & 8\end{bmatrix}$ //

$(x_1-\mu_2)(x_1-\mu_2)^T$

$= \begin{pmatrix}0\\-2\end{pmatrix}(0 \quad -2) = \begin{pmatrix}0 & 0\\0 & 4\end{pmatrix} = m_1'$

$(x_2-\mu_2)(x_2-\mu_2)^T$

$= \begin{pmatrix}0\\2\end{pmatrix}(0 \quad 2) = \begin{pmatrix}0 & 0\\0 & 4\end{pmatrix} = m_2'$

$(x_3-\mu_2)(x_3-\mu_2)^T$

$= \begin{pmatrix}-2\\0\end{pmatrix}(-2 \quad 0) = \begin{pmatrix}4 & 0\\0 & 0\end{pmatrix} = m_3'$

$(x_4-\mu_2)(x_4-\mu_2)^T$

$= \begin{pmatrix}2\\0\end{pmatrix}(2 \quad 0) = \begin{pmatrix}4 & 0\\0 & 0\end{pmatrix} = m_4'$

$\Sigma_2 = \frac{1}{4}[m_1'+m_2'+m_3'+m_4']$

$= \frac{1}{4}\begin{bmatrix}8 & 0\\0 & 8\end{bmatrix} = \begin{bmatrix}2 & 0\\0 & 2\end{bmatrix}$

$\Sigma_2^{-1} = \begin{bmatrix}1/2 & 0\\0 & 1/2\end{bmatrix}$

$|\Sigma_1| = 21$

$|\Sigma_2| = 4$

$\therefore g(x) =$

$$\Sigma_1^{-1} - \Sigma_2^{-1} = \begin{bmatrix} 11/84 & -1/21 \\ -1/21 & 8/21 \end{bmatrix} - \begin{bmatrix} 4_2 & 0 \\ 0 & 4_2 \end{bmatrix}$$

$$\Sigma_1^{-1} - \Sigma_2^{-1} = \begin{bmatrix} -31/84 & -1/21 \\ -1/21 & -5/42 \end{bmatrix}$$

$$\therefore g(x) = -\frac{1}{2} x^T (\Sigma_1^{-1} - \Sigma_2^{-1}) x + (\mu_1^T \Sigma_1^{-1} - \mu_2^T \Sigma_2^{-1}) x$$
$$-\frac{1}{2} \ln\left(\frac{21}{4}\right) + \ln\left(\frac{4_2}{4_2}\right) - \frac{1}{2} \mu_1^T \Sigma_1^{-1} \mu_1 + \frac{1}{2} \mu_2^T \Sigma^{-1} \mu_2 \cdots$$

$$\therefore g(x) = \frac{-1}{2} [x_1 \; x_2] \begin{bmatrix} -31/84 & -4/21 \\ -4/21 & -5/42 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$+ \left([9 \;\; 9/2] \begin{bmatrix} 11/84 & -1/21 \\ -4/21 & 8 \end{bmatrix} - [9 \;\; 12] \begin{bmatrix} 4_2 & 0 \\ 0 & 4_2 \end{bmatrix} \right) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$-\frac{1}{2} \left([9 \;\; 9/2] \begin{bmatrix} 11/84 & -1/21 \\ -4/21 & 8 \end{bmatrix} \begin{bmatrix} 9 \\ 9/2 \end{bmatrix} \right) + \frac{1}{2} \left([9 \;\; 12] \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} \begin{bmatrix} 9 \\ 12 \end{bmatrix} \right)$$

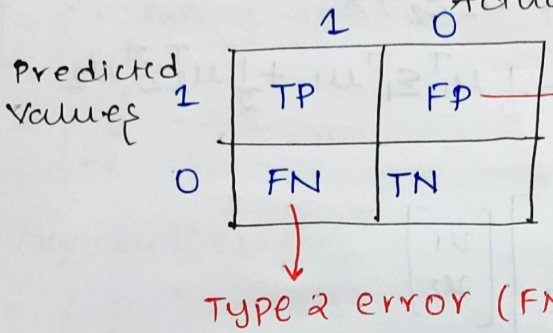
$$-\frac{1}{2} \ln(21/4) + D$$

$\therefore g(x) =$

* Metrics in classification *.

NOTE: when we have balanced dataset at that time we usually prefer accuracy as our metric.
for imbalance dataset we can either choose precision or recall. or F1-score.

(1) Confusion Metric :- → used in binary classification.



→ 2×2 matrix.

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

→ our aim should be reducing type 1 and type 2 error.

→ $$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

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

(2) Recall :- → It also mentioned by TPR, Sensitivity.
→ used for imbalance dataset.
→ It signifies that out of all actual positive values how much we predicted correctly.

→ $$Recall = \frac{TP}{TP + FN}$$

(3) Precision :- → used when there is imbalance

→ It is also known as positive prediction value.

→ out of ~~tru~~ positively predicted values what % of values are actually positive.

→ $$\text{Precision} = \frac{TP}{TP + FN}$$

*NOTE :- when to use what?

1) whenever (FP) is more important at that time we will use **Precision**.

2) whenever (FN) is more important at that time we will use **Recall**.

Examples:- In spam detection we will use precision as a measure and in cancer detection we will use recall as a measure.

(4) F-Beta score :-

→ our main aim is to select (β) value.

→ $$F_{Beta} = \frac{(1+\beta^2).\ Precision \times Recall}{Precision + Recall}$$

$$F1\text{-}score = \frac{2.\ Precision \times Recall}{Precision + Recall}$$ ($\beta = 1$) → Harmonic mean.

NOTE :- when to choose $\beta = 1$?

→ when (FP) and (FN) both are equally important.

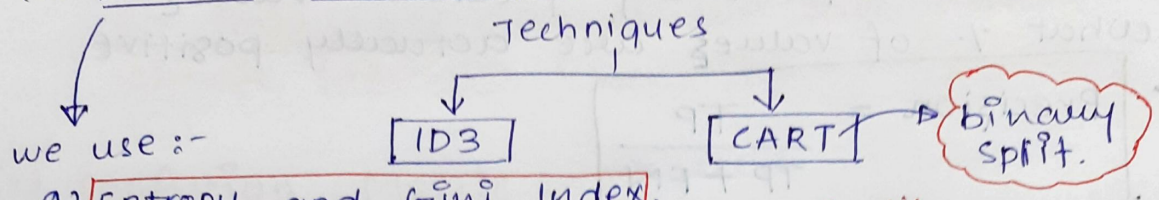
→ when (FP) more imp. we reduce β value.

→ when (FN) more imp we ↑ β value.

* Decision Tree *

NOTE :- Decision Tree classifier can be used for both classification and regression.

(*) Decision Tree classifier :-



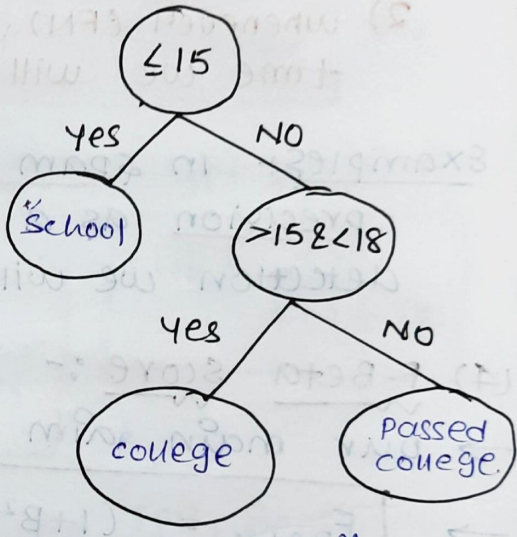
we use :-

a) Entropy and Gini Index. → purity of split.

b) Information gain → helps us to decide that using which feature we need to split.

Example :- Basic Decision Tree

```
age = 14
if age ≤ 15
    pf("school");
elif age ≥ 15 and ≤ 18
    pf("college")
else
    pf("passed college");
```



- Leaf Node :-

→ They are also called as "turminal node".
→ "Nodes that don't split into more nodes."
→ classes are assigned by majority vote.
→ classes are purely split.

→ for impure split we will further split the node with other features.

Q. How to check the split is pure or not?

→ using entropy and gini index./ gini impurity.

Q. How the features are selected?

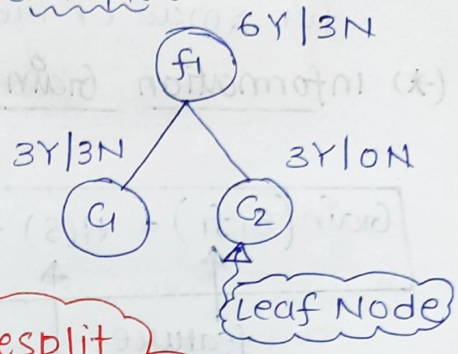
→ information gain.

(*) Entropy :-

Assuming binary classification,

$$H(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

Example :-



$$H(C_2) = -\frac{3}{3}\log_2\frac{3}{3} - 0$$

$$H(C_2) = 0$$

if H(S)=0 → puresplit → Leaf Node

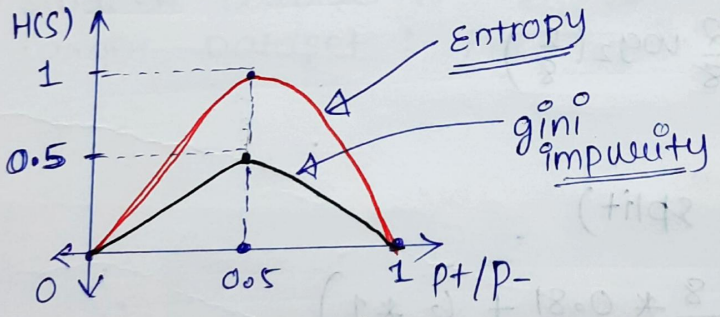
$$H(C_1) = -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6}$$

$$= -\log_2\frac{1}{2} = -(\log_2 1 - \log_2 2) = 1.$$

$$H(C_1) = 1$$

if H(S)=1 → Highly impure split

*) Graphically (Entropy / gini) :-



Entropy Graph.

*. NOTE :- Entropy ranges from 0 to 1

*) Gini Impurity :-

$$\text{Gini impurity} = 1 - \sum_{i=1}^{n} (p)^2$$

Assume binary classification,

$$\text{Gini impurity} = 1 - ((p_+)^2 + (p_-)^2)$$

NOTE: Gini impurity ranges from 0 to 0.5.

pure split — Highly impure split.

Q. What to use Entropy or Gini Impurity ?

→ for large dataset :- use gini impurity

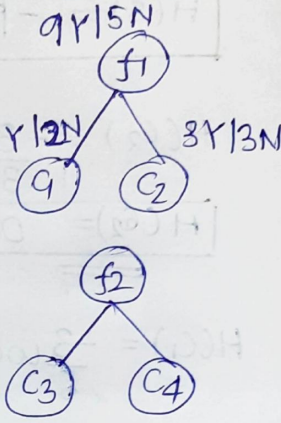
Reason :- log operation in entropy will take more time than the algebraic operation in gini impurity.

for small or medium dataset :- use entropy

(*) Information Gain :- (It tells us about feature selection).

$$Gain(S, f_1) = H(S) - \sum_{v \in value} \frac{|S_v|}{|S|} H(S_v)$$

feature; Entropy of root node; Total samp.; Entropy of category (v).



$$H(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

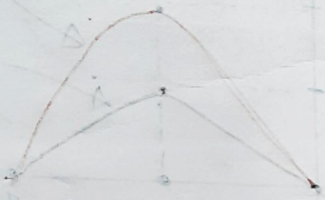
$$H(S) = -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14}$$

$$\boxed{H(S) \approx 0.94}$$

$$H(C_1) = -\frac{6}{8} \log_2 \frac{6}{8} - \frac{2}{8} \log_2 \left(\frac{2}{8}\right)$$

$$\boxed{H(C_1) \approx 0.81}$$

$$\boxed{H(C_2) = 1} \quad (\because \text{impure split})$$



$$Gain(S, f_1) = 0.94 - \left(\frac{8}{14} * 0.81 + \frac{6}{14} * 1\right)$$

$$\boxed{Gain(S, f_1) \approx 0.049} \quad //$$

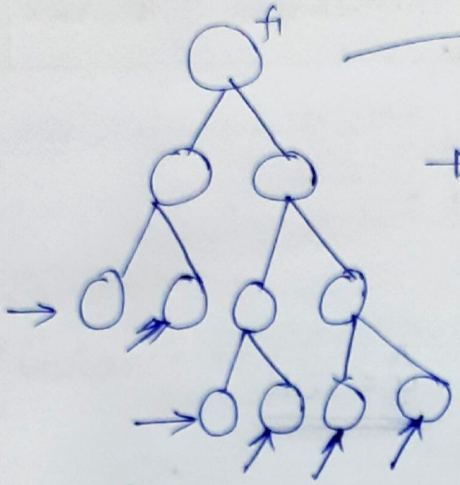
Similarly, calculate $Gain(S, f_2)$.

How to select feature?

if $Gain(S, f_2) > Gain(S, f_1)$
    go with f_2 feature.
else
    go with f_1 feature.

* Post and Pre-pruning in Decision Tree *.

Assume Decision Tree,



→ This kind of D.T. may lead to overfitting.

→ So, to overcome it.
we use i) Post Pruning.
ii) Pre Pruning.

(I) **Post Pruning** :- → also known as backward pruning.
→ Doing Hyperparameter tuning after construction of tree.
→ Example :- max_depth., min_samples_split

(II) **Pre-Pruning** :-
→ Doing Hyperparameter tuning simultaneously while constructing decision tree.

Q. **When to use what ?**

- Smaller Dataset :- Post Pruning.
- Larger Dataset :- Pre Pruning.

PrePruning is more efficient among both.

↓

Because it doesnot constructs whole D.T.

* Decision Tree Regressor *

sample dataset :-

Experience	Grap	salary
2	Yes	40K
2.5	Yes	42K
3	NO	52K
4	NO	60K
4.5	Yes	56K

independent features — output

$\hat{y} = 50K$

which split??

Tree1 :-

var = 60.8

≤2

variance = 100 (40)

42, 52, 60, 56

var = 51

Tree 2 :-

≤2.5

40, 42

52, 60, 56

→ we will use variance reduction

$$\text{Variance} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2$$

$\bar{y}$ = avg. output.

mse

variance (Root) $= \frac{1}{5}(100 + 64 + 4 + 100 + 36)$

$= \frac{304}{5} = \boxed{60.8}$

For Tree 1 :-

Variance (child 1)

$= \frac{1}{1} + (40-50)^2$

$= \boxed{100}$

Variance (child 2)

$= \frac{1}{4}[64 + 4 + 100 + 36]$

$= \frac{204}{4} = \boxed{51}$

For Tree 2 :-

var(C_1)

$= \frac{1}{2}[100 + 64]$

$= \frac{164}{2} = \boxed{82}$ ✓

var(C_2)

$= \frac{1}{3}[4 + 100 + 36]$

$= \frac{140}{3} = \boxed{46.67}$

*. variance reduction :-

$$\text{variance reduction} = \text{var(root)} - \sum w_i \, \text{var}(c_i)$$

//

For Tree1 :

$$\text{variance reduction} = 60.8 - \left(\frac{1}{5} * 100 + \frac{4}{5} * 51\right)$$

$$= 60.8 - \frac{304}{5} = 0$$

For Tree 2 :

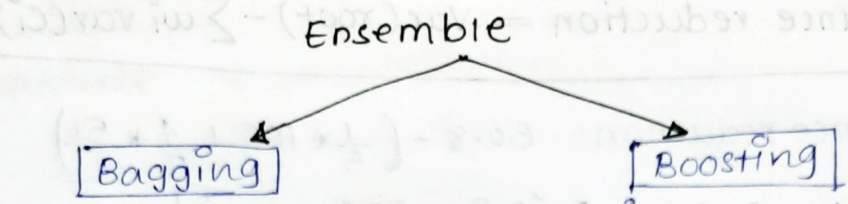
$$\text{variance reduction} = 60.8 - \left(\frac{2}{5} * 82 + \frac{3}{5} * 46.67\right)$$

$$= 60.8 - \left(\frac{164 + 140}{5}\right) \Rightarrow 0 \quad 0.004$$

we will choose split-2.

Reason :- VR(2) > VR(1) → 0.004 > 0.000

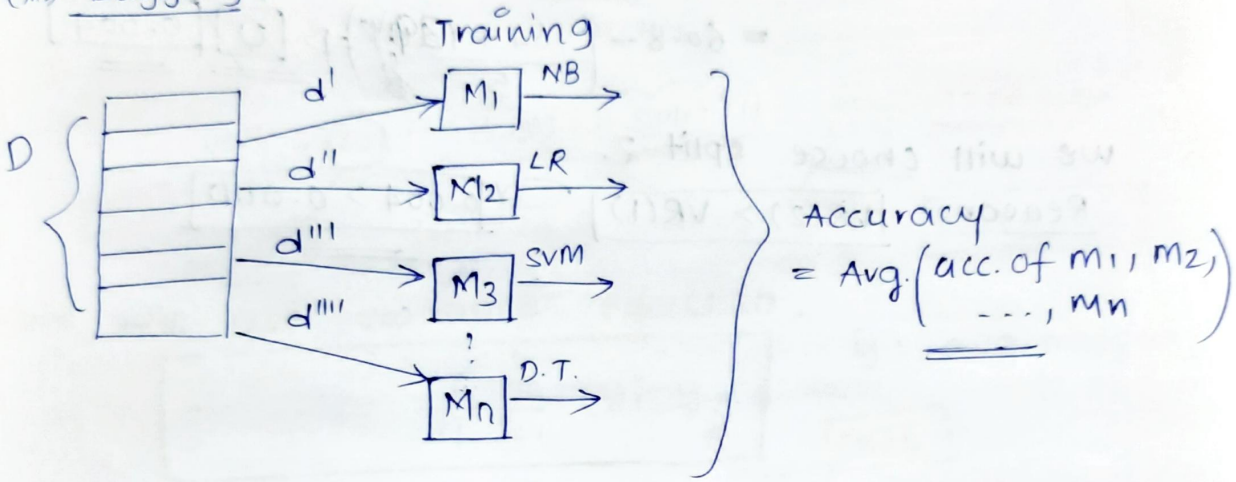
* Ensemble Techniques *.



i) Random Forest

i) Ada Boost.
ii) Gradient Boost.
iii) XGBoost.
iv) CAT Boost.

(*) Bagging :-

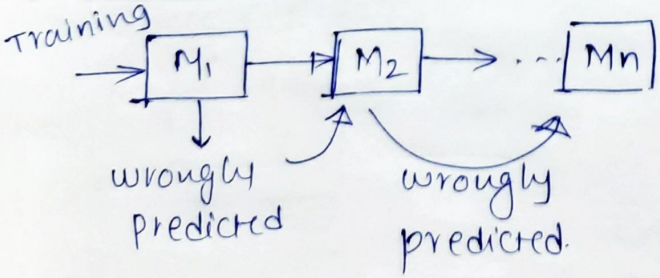


In case of Test Data:-

(I) In case of classification, max-voting is used. maximum-voted label will be the output label.

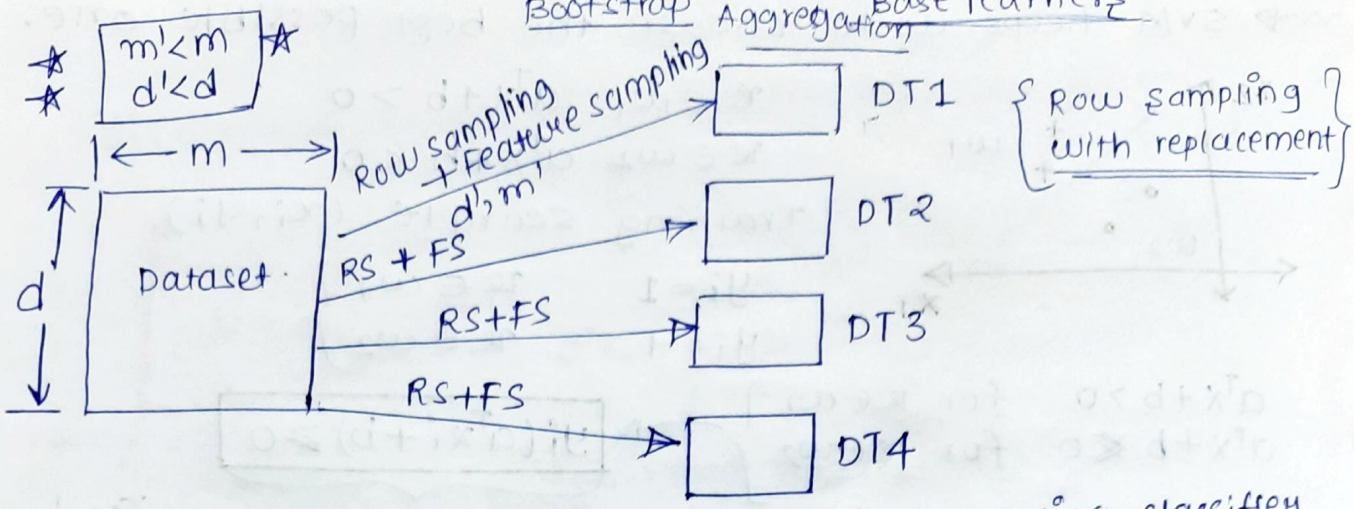
(II) In case of regression, avg of all output is the output of test data.

(*) Boosting :- {weak learners}.



* Random Forest Regression and classification *.

→ Base learners are decision tree models.



→ In case of classification, we use majority voting classifier.

→ In case of regression, we compute avg of all outputs.

Q. Why we are using Random Forest over Decision Trees?

→ In Decision Trees, we get

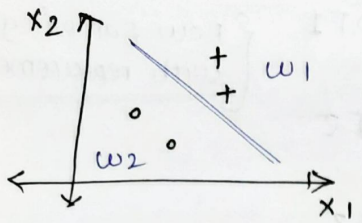
{ Low Bias, High variance } overfitting.

We need to convert this High variance to low variance in which ~~decisio~~ Random forest helps us.

Because, in Random Forest we are using multiple decision trees with Row sampling and Feature sampling.

* Support Vector Machines *

- There are multiple separating surfaces.
- SVM helps us to choose the best possible one.



$x \in \omega_1 \quad a^T x + b > 0$

$x \in \omega_2 \quad a^T x + b < 0$

Training sample (x_i, y_i).

$y_i = 1 \quad x \in \omega_1$

$y_i = -1 \quad x \in \omega_2$

$a^T x + b > 0$ for $x \in \omega_1$

$a^T x + b \not< 0$ for $x \in \omega_2$

$\rightarrow \boxed{y_i (a^T x_i + b) \geq 0}$ if correctly classified.

NOTE $\Rightarrow$ Distance of feature vector belongs to class ω_1 and class ω_2 from the separating surface should be maximize.

→ In otherwords, margin should be maximized.

→ Feature vector near to the separating surface will have less confidence of being correctly classified than the feature vector farther from that.

- Separating Plane :-

$a^t x + b = 0$ gives the location of sep. plane.

a orthogonal to separating plane.

→ change in vector (a) leads to change in orientation of separating plane.

→ change in (b) leads to change in position / loc. of sep. plane.

(*) Benefits of SVM :-

1) Robust to outliers.
2) Can work on non-linear data.
3) simple.
4) sparse solution
5) Global optimum solution.

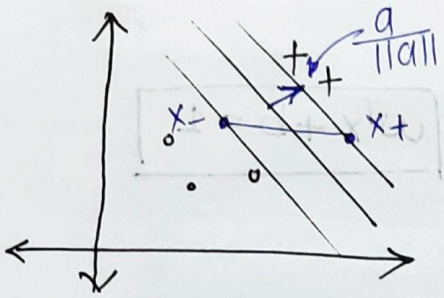
$a^T x_i + b > d$ $\quad a^T x_i + b < -d$.

threshold.

$\therefore$ $y_i(a^T x_i + b) > d$

correctly classifying.

$y_i(a^T x_i + b) \geq 1$.



Margin $\Rightarrow \frac{a^T}{\|a\|}(x_+ - x_-)$ ——— (*)

$a^T x_+ + b = 1$ ——— ①

$a^T x_- + b = -1$ ——— ②

subtract ② from ①.

$a^T(x_+ - x_-) = 2$ ——— ③

substitute eqn-(3) in (*)....

Margin $\Rightarrow \frac{2}{\|a\|}$ //

from here we observe that to increase the margin we have to $\downarrow a$.

But $\downarrow a$ such that it follows $y_i(a^T x_i + b) \geq 1$.

Now, this becomes a constraint optimization prblm. It can be solved using langrangian.

$$L = \frac{\|a\|^2}{2} - \sum_{\forall i} \alpha_i [y_i(a^T x_i + b) - 1]$$

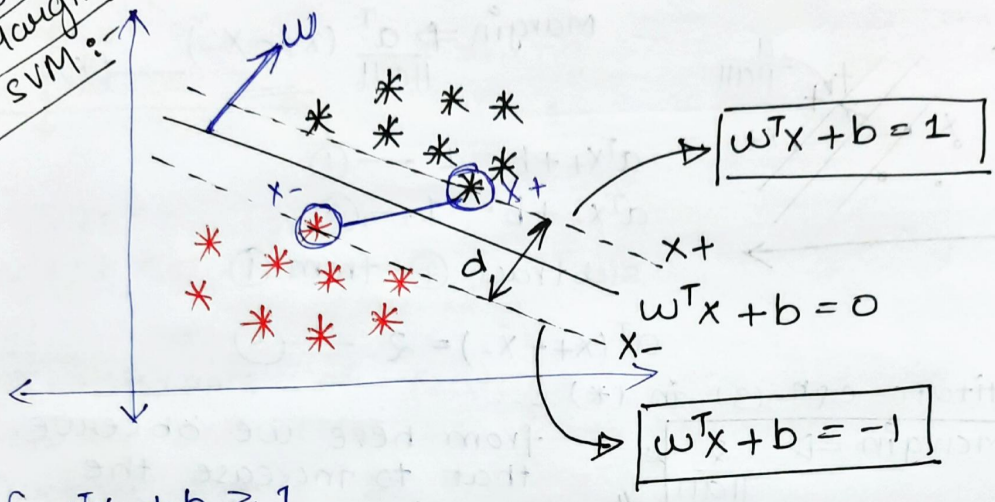
$$\frac{\partial L}{\partial a} = a - \sum_{\forall i} \alpha_i y_i x_i = 0 \qquad \frac{\partial L}{\partial b} = -\sum_{\forall i} \alpha_i y_i = 0$$

$$\Rightarrow a = \sum_{\forall i} \alpha_i y_i x_i \qquad \sum_{\forall i} \alpha_i y_i = 0$$

for any x_i

$$y_i = \begin{cases} +1 & \text{if } w^T x_i + b \geq 0 \\ -1 & \text{if } w^T x_i + b < 0 \end{cases}$$

* Hard Margin SVM :-



$$\begin{cases} w^T x_+ + b \geq 1 \\ w^T x_- + b \leq -1 \end{cases}$$

$$\{ y_i (w^T x + b) \geq 1 \} \text{ —— ÷}$$

$y_i (w^T x_i + b) = 1$ —— (*)

for support vector.

for correctly classifying

$$\frac{w^T}{\|w\|} (x_+ - x_-) = d \text{ —— (4)}$$

from (*)

(1) $(w^T x_+ + b) = 1$ —— (1)

(-1) $(w^T x_- + b) = 1$ —— (2)

add (1) and (2)

$w^T (x_+ - x_-) = 2$ —— (3)

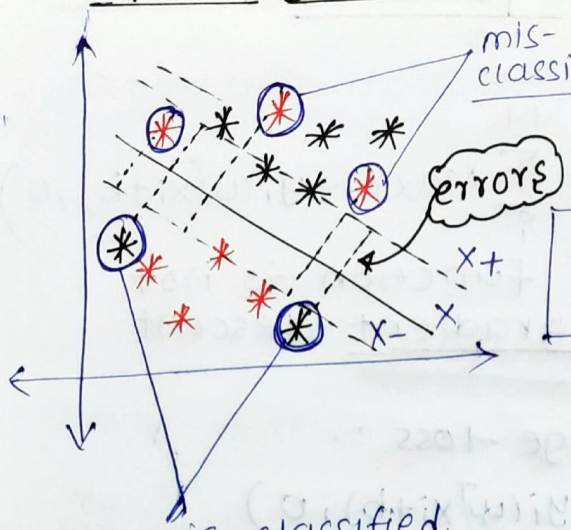
substitute (3) in (4)

$$d = \frac{2}{\|w\|}$$ ← width of margin ✓

argmax $\frac{2}{\|w\|}$ such that (w^*, b^*)

$y_i (w^T x + b) \geq 1$

(*) Soft Margin SVM :-



$$\underset{(w^*,b^*)}{\operatorname{argmin}} \frac{\|w\|}{2}$$

↳ Hard Margin.

$$\underset{(w^*,b^*)}{\operatorname{argmin}} \frac{\|w\|^2}{2} + C\sum_{i=1}^{l} \xi_i$$

(ξ = zeta) slack variable.

$\xi_i^* = 0$ } for correctly classified points.

SVM error = Margin Error + classification Error

$$J = \underset{(w^*,b^*)}{\operatorname{argmin}} \lambda \frac{\|w\|^2}{2} + C\sum_{i=1}^{l} \xi_i$$

Hinge loss.

regularized...

subject to $y_i(w^T x_i + b) \geq 1 - \xi_i$

↳ Hyperparameter.

$$\xi_i = \max\left(1 - y_i(w^T x_i + b),\ 0\right)$$

~~$\nabla_w J(w,b) = \lambda w - \dots$~~

NOTE:

1) $y_i(w^T x_i + b) > 1$.
   - point outside margin.
   - No contribution to loss.

2) $y_i(w^T x_i + b) < 1$.
   - point violates margin constraints.
   - contributes to loss.

3) $y_i(w^T x_i + b) = 1$
   - point is on margin.

* NOTE:

1) small C allow constraints to be easily ignored (large margin)
2) large C → narrow margin...
3) $C = \infty$ → Hard Margin.

(*) Gradient Descent Algo for SVM :-

optimization problem

$$\min_{w} J(w) = \frac{\lambda}{2}\|w\|^2 + \frac{1}{N}\sum_{i}^{N} \max\left(1 - y_i(w^T x_i + b), 0\right)$$

Because, the hinge loss function is not differentiable, a sub-gradient descent is used.

⇒ sub-gradient for hinge loss :-

$$L(x_i, y_i, w) = \max\left(1 - y_i(w^T x_i + b), 0\right)$$

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

⇒ do iterative

$$w_{t+1} \leftarrow w_t - \eta \nabla_{w_t} J(w_t)$$

$$\leftarrow w_t - \eta \frac{1}{N}\left(\lambda w_t + \nabla_w L(x_i, y_i, w)\right)$$

$$\therefore \quad w_{t+1} \leftarrow w_t - \eta(\lambda w_t - y_i x_i) \quad \text{if } y_i(w^T x_i + b) < 1$$

otherwise

$$\leftarrow w_t - \eta(\lambda w_t)$$

(*.) PCA :-

→ assume that the data is centered. which means every variable has mean (0).

→ for 1-D projection. Let's take a line with unit vector along it $\vec{w}$ and projection data vector $\vec{x_i}$ onto the line $\vec{x}.\vec{w}$ which is scalar.

→ The actual cordinate in p-dimensional space is $(\vec{x}.\vec{w})\vec{w}$.

→ mean of the projection will be zero. since the mean of data vector $(\vec{x_i})$ is zero.

$$\frac{1}{n}\sum_{i}^{n}(\vec{x}.\vec{w})\vec{w} = \left(\left(\frac{1}{n}\sum_{i}^{n}\vec{x}\right)\vec{w}\right)\vec{w} \quad \text{— (*)}$$

→ error of the projection for data vector $\vec{x_i}$.

given by,

$$\|\vec{x_i} - (\vec{w}.\vec{x_i})\vec{w}\|^2 = (x_i - (w.x_i)w)(x_i - (w_i.x_i)w)$$

$$= \vec{x_i}.\vec{x_i} - \vec{x_i}.(\vec{w_i}.\vec{x_i})\vec{w} - (\vec{w_i}.\vec{x_i})\vec{w_i}.\vec{x_i} + (\vec{w}.\vec{x_i})\vec{w}.(\vec{w}.\vec{x_i}).\vec{w}$$

$$= \|\vec{x_i}\|^2 - 2(\vec{w}.\vec{x_i})^2 + (\vec{w}.\vec{x_i})^2\vec{w}.\vec{w}$$

$$= \|\vec{x_i}\|^2 - 2(\vec{w}.\vec{x_i})^2 + (\vec{w}.\vec{x_i})^2 \quad [\vec{w}.\vec{w}=1]$$

$$= \vec{x_i}.\vec{x_i} - (\vec{w}.\vec{x_i})^2$$

NOW, $$MSE(\vec{w}) = \frac{1}{n}\left(\underbrace{\sum_{i}^{n}\|\vec{x_i}\|^2}_{\text{independent of } (w).} - \sum_{i}^{n}(\vec{w_i}.\vec{x_i})^2\right)$$

independent of (w). will not matter in minimizing MSE.

Thus to minimize the MSE, we need to maximize second term. i.e.,

$$\text{Max} \left[\frac{1}{n} \sum_{i}^{n} (\vec{w} \cdot \vec{x_i})^2 \right] \} \text{ Mean of } (\vec{w} \cdot \vec{x_i})^2$$

(*) Mean of square = square of mean + variance.

$$\text{Max} \frac{1}{n} \sum_{i}^{n} (\vec{w} \cdot \vec{x_i})^2 = \left(\frac{1}{n} \sum_{i}^{n} \vec{x_i} \cdot \vec{w} \right)^2 + \text{Var} [\vec{w} \cdot \vec{x_i}]$$

we know that mean of projection is zero [from (*)]
Thus, minimizing MSE would result in maximizing the variance of projections.

⇒ Maximizing the variance:-

- n data vectors
- $X_{n \times p}$ matrix
- projection is given by Xw $n \times 1$ matrix.

variance is $\sigma_{\vec{w}}^2 = \frac{1}{n} \sum_{i}^{n} (\vec{x_i} \cdot \vec{w_*})^2$

$$= \frac{1}{n} (Xw)^T (Xw)$$

$$= \frac{1}{n} w^T X^T X w$$

$$= w^T \frac{X^T X}{n} w$$

$$\boxed{\sigma_w^2 = w^T \Sigma w}$$

Σ = covariance matrix

$\frac{X^T X}{n} = \Sigma$ ($\because$ X is centered)

→ we need to find vector w such that it maximizes $\sigma_{\vec{w}}^2$. for that we use langragian multiplier (λ).

$$L(w, \lambda) = \sigma_{\vec{w}}^2 - \lambda(w^T w - 1) = w^T \Sigma w - \lambda(w^T w - 1)$$

$$\frac{\partial L}{\partial w} = w^T w - 1 = 0 \qquad \frac{\partial L}{\partial w} = 2\Sigma w - 2\lambda w = 0$$

$$\boxed{w^T w = 1} \qquad \boxed{\Sigma w = \lambda w}$$

$$\Sigma w = \lambda w$$

w is the eigen vector of covariance matrix (Σ). maximizing the vector is associated with max. eigen value.

Thus,

> The eigen vectors of Σ are the Principal components of the data.

→ The 1st principle component ~~with~~ will be the largest eigen vector which goes with largest eigen value (λ).

* Softmax Regression *

(Multinomial LR)

Softmax function =

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

K = # class.

$$0 < \sigma < 1$$

$$\sigma(z)_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}} \qquad \sigma(z)_2 = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$\sigma(z)_3 = \frac{e^{z_3}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$\sigma(z)_1 + \sigma(z)_2 + \sigma(z)_3 = 1$$

In softmax regression, loss is the sum of distance between labels and output probability distribution. which is known as cross entropy.

$$\text{cross entropy}_{x_i} = -\sum_{j=1}^{K} \left(I(y^i = K) \cdot \log \left(\frac{e^{w^T x^i}}{\sum_{j=1}^{K} e^{w^T x^K}} \right) \right)$$

K = classes

Total cross-entropy or loss,

$$\text{Loss} = -\sum_{i=1}^{m} \sum_{j=1}^{K} \left(I(y^i = K) \cdot \log \frac{e^{w^T x_i}}{\sum_{j=1}^{K} e^{w^T x_j}} \right)$$

$$\nabla_w \text{loss} = -\frac{1}{m} \sum_{i=1}^{m} x^i \left(I(y^i = K) - \frac{e^{w^T x_i}}{\sum_{j=1}^{K} e^{w^T x_j}} \right)$$

New parameter for class K,

$$W_{K+1} = W_K + \alpha \cdot \frac{1}{m} \cdot \sum_{i=1}^{m} x^i \left(I(y^i = K) - \frac{e^{w^T x_i}}{\sum_{j=1}^{K} e^{w^T x_j}} \right)$$

#

*. K-means clustering :-

Advantages :

1) simple to implement.
2) Easily adapt new examples.
3) guarantees convergence by trying to minimize the total SSE.
4) Fast and efficient. complexity → $O(n*k*d)$

Disadvantage :-

1) clustering outliers.
2) Dependent on initial values.
3) choosing k-manually.
4) Scaling with number of dimensions.

objective Function is

$$\text{Min } J = \sum_{i=1}^{m} \sum_{j=1}^{K} w_{ik} \| x^i - \mu_k \|^2$$

μ_k ↑ centroid of class K.

Two steps of minimization :

1) minimize w_{ik} given that μ_k fixed. (cluster assgnmt)
2) find μ_k given that w_{ik} fixed. (centroid comput)

$$\frac{\partial L}{\partial w_k} = \sum_{i=1}^{m} \sum_{j=1}^{K} \| x_i - \mu_j \|^2$$

$$\Rightarrow w_k = \begin{cases} 1 & \text{if } k = \arg\min_j \| x_i - \mu_j \|^2 \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{\partial L}{\partial \mu_k} = 2 \sum_{i=1}^{m} w_{ik} (x_i - \mu_k) = 0$$

$$\Rightarrow \mu_k = \frac{\sum_{i=1}^{m} w_{ik} x_i}{\sum_{i=1}^{m} w_{ik}}$$