ML_Hw5 R09946023 吳偉樂

1) (d) $w_1^* = 0$

1. (d) $w_1^* = 0$

$\{(x_n, y_n)\}_{n=1}^{3} = \{(-2,-1), (0,+1), (2,-1)\}$

$\phi(x) = [1, x, x^2]^T \Leftrightarrow$ the hyperplane is $w_0^* + w_1^* x + w_2^* x^2 = f(x)$, where $w_0^* = b^*$

$\Rightarrow \min_{b,w} \frac{1}{2} w^{*T} w^*$ subject to $y_n(b^* + w_1^* x_n + w_2^* x_n^2) \geq 1$ for $n = 1,2,3$, $w^* = [w_1^* \ w_2^*]$

observe $y_n(b^* + w_1^* x_n + w_2^* x_n^2) \geq 1$

(i) $(-2,-1)$: $-b^* + 2w_1^* - 4w_2^* \geq 1$

(ii) $(0,+1)$: $b^* \geq 1$

(iii) $(2,-1)$; $-b^* - 2w_1^* - 4w_2^* \geq 1$

$\Rightarrow$ (i), (ii) : $w_1^* - 2w_2^* \geq 1$ ~ (*)

(i), (iii) : $-b^* - 4w_2^* \geq 1 \overset{(ii)}{\Rightarrow} -4w_2^* \geq 2 \Rightarrow w_2^* \leq -\frac{1}{2}$

(ii), (iii) : $-w_1^* - 2w_2^* \geq 1$ ~ (**)

(*), (**) : $-4w_2^* \geq 2 \Rightarrow w_2^* \leq -\frac{1}{2}$

subject $w_2^* \leq -\frac{1}{2}$ into (*), (**) :

In (*) : $w_1^* \geq 0$

In (**) : $w_1^* \leq 0$

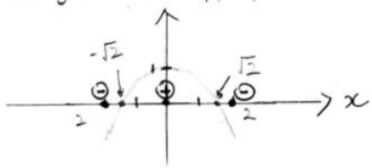
$\Rightarrow$ If $w_2^* \leq -\frac{1}{2}$ and $b^* \geq 1$, then w_1^* must equal to zero $(w_1^* = 0)$ to holds the inequality

only possible is $w_1^* = 0$

2) (b) 2

2. (b) 2

Use the results in (1.), take $w_1^* = 0$, $w_2^* = -\frac{1}{2}$ and $b^* = 1$, then we get the hyperplane $1 - \frac{1}{2}x^2 = f(x)$



$$\text{margin}(b^*, w^*) = \frac{1}{\|w\|} = \frac{1}{\sqrt{\frac{1}{4}}} = 2$$

(3) **(e)** $\frac{1}{2}(x_{m+1} - x_m)$

3. (e) $\frac{1}{2}(x_{m+1} - x_m)$

$\{(x_n, y_n)\}_{n=1}^{N}$ for $x_n \in \mathbb{R}$ (1-D examples)

$\underbrace{x_1 \le x_2 \le \dots \le x_m}_{y_n = -1} < \underbrace{x_{m+1} \le \dots \le x_N}_{y_n = +1}$ $\Rightarrow$ may coincide, may coincide

since the examples are linearly seperable, so the lines must pass through somewhere between x_m, x_{m+1} (eg. : , , , ...)

Since the linear margin is symmetric to the hyperplane (eg: same)

So the largest symmetric margin is the middle between x_m and x_{m+1} (x o) $\Rightarrow$ margin $= \frac{1}{2}(x_{m+1} - x_m)$

margin to the +ve examples equal to the margin to the -ve examples

(4) **(a)**

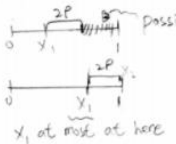
4. (a) $2 + 2(1-2\rho)^2$

case ①: if $|x_1 - x_2| < 2\rho$, then ρ 2ρ ρ 0 x_1 x_2 1 ← we may use the line with margin at least ρ to seperate it, but the line is either at right of both x_1, x_2 or left or both x_1, x_2 : or

$\Rightarrow m_H = 2$.

case ②. If $|x_1 - x_2| \geq 2\rho$, 2ρ 0 x_1 x_2 1, then we have $m_H = 2^2 = 4$ ($\because$ x_1 x_2 , x_1 x_2)

in case ②, if $x_1 < x_2$ (or $x_2 < x_1$), then



$\therefore E(x_2) = E(1 - x_1 - 2\rho)$ x_1 uniform (0,1)

$= \int_0^{1-2\rho} (1 - x_1 - 2\rho)\, dx_1$

$= (1-2\rho)x_1 - \frac{x_1^2}{2} \Big|_0^{1-2\rho} = \frac{(1-2\rho)^2}{2}$

By case ① & ②,

$\Rightarrow E(m_H)$

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

$= 2 + 2(1-2\rho)^2$

(5) (c) $-\sum_{n=1}^{N}\rho_{+}[\![y_n=+1]\!]\alpha_n \;-\; \sum_{n=1}^{N}\rho_{-}[\![y_n=-1]\!]\alpha_n$

5. (c) $-\sum_{n=1}^{N}\rho_{+}[\![y_n=+1]\!]\alpha_n - \sum_{n=1}^{N}\rho_{-}[\![y_n=-1]\!]\alpha_n$

$\min_{b,w} \frac{1}{2}w^Tw$ subject to $y_n(w^Tx_n+b) \geq \rho_+$ for n s.t. $y_n=+1$

$y_n(w^Tx_n+b) \geq \rho_-$ for n s.t. $y_n=-1$

$\Rightarrow$ Lagrange function with multipliers α_n,

$$L(b,w,\alpha) = \frac{1}{2}w^Tw + \sum_{n=1}^{N}\alpha_n[\![y_n=+1]\!]\left(\rho_+ - y_n(w^Tx_n+b)\right) + \sum_{n=1}^{N}\alpha_n[\![y_n=-1]\!]\left(\rho_- - y_n(w^Tx_n+b)\right)$$

Solving Lagrange Dual: $\max_{\text{all }\alpha_n\geq 0}\left[\min_{b,w} L(b,w,\alpha)\right]$

$\sum_{n=1}^{N} y_n\alpha_n = 0 \Rightarrow$ b can be removed

$$\frac{\partial L}{\partial b} = 0 = -\sum_{n=1}^{N}\alpha_n[\![y_n=+1]\!]y_n - \sum_{n=1}^{N}\alpha_n y_n[\![y_n=-1]\!] \Rightarrow \sum_{n=1}^{N}\alpha_n y_n\left[[\![y_n=+1]\!]+[\![y_n=-1]\!]\right] = 0$$

$\Rightarrow$ Now: $\max_{\substack{\text{all }\alpha_n\geq 0\\ \sum_{n=1}^{N}\alpha_n y_n=0}}$

$$\Bigg(\underbrace{\min_{w} \frac{1}{2}w^Tw + \sum_{n=1}^{N}\alpha_n[\![y_n=+1]\!](\rho_+ - y_nw^Tx_n) + \sum_{n=1}^{N}\alpha_n[\![y_n=-1]\!](\rho_- - y_nw^Tx_n)}_{L(w,\alpha)}\Bigg)$$

$$\frac{\partial L}{\partial w_i} = 0 = w_i - \sum_{n=1}^{N}\alpha_n[\![y_n=+1]\!]y_nx_{n,i} - \sum_{n=1}^{N}\alpha_n[\![y_n=-1]\!]y_nx_{n,i}$$

$$\Rightarrow w = \sum_{n=1}^{N}\alpha_n y_n[\![y_n=+1]\!]x_n + \sum_{n=1}^{N}\alpha_n[\![y_n=-1]\!]y_nx_n$$

$$L(w,\alpha) = \frac{1}{2}w^Tw + \sum_{n=1}^{N}\alpha_n[\![y_n=+1]\!]\rho_+ + \sum_{n=1}^{N}\alpha_n[\![y_n=-1]\!]\rho_-$$

$$= -\frac{1}{2}w^Tw + \sum_{n=1}^{N}\alpha_n[\![y_n=+1]\!]\rho_+ + \sum_{n=1}^{N}\alpha_n[\![y_n=-1]\!]\rho_-$$

Now: $\max_{\substack{\text{all }\alpha_n\geq 0\\ \sum y_n\alpha_n=0\\ w=\sum\alpha_ny_nx_n}} \left(-\frac{1}{2}\left\|\sum_{n=1}^{N}\alpha_ny_nx_n\right\|^2 + \sum_{n=1}^{N}[\![y_n=+1]\!]\rho_+\alpha_n + \sum_{n=1}^{N}[\![y_n=-1]\!]\rho_-\alpha_n\right)$

$$\Leftrightarrow \min_{\alpha} \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_my_ny_mx_n^Tx_m \underline{- \sum_{n=1}^{N}[\![y_n=+1]\!]\rho_+\alpha_n - \sum_{n=1}^{N}[\![y_n=-1]\!]\rho_-\alpha_n}$$

subject to $\sum_{n=1}^{N}y_n\alpha_n = 0$

$\alpha_n \geq 0$ for $n=1,2,\ldots,N$

6) (e)

6. (e) $\frac{P_+ + P_-}{2}\alpha^*$

In dual-inner optimal, we have $\sum_{n=1}^{N} y_n \alpha_n = 0$

$$\Rightarrow. \sum_{n=1}^{N} y_n \alpha_n = \sum_{n=1}^{N} \alpha_n [\![y_n = +1]\!] - \sum_{n=1}^{N} \alpha_n [\![y_n = -1]\!] = 0$$

$$\Rightarrow. \sum_{n=1}^{N} \alpha_n [\![y_n = +1]\!] = \sum_{n=1}^{N} \alpha_n [\![y_n = -1]\!] \quad —(*)$$

And we have $\sum_{n=1}^{N} \alpha_n = \sum_{n=1}^{N} \alpha_n [\![y_n = +1]\!] + \sum_{n=1}^{N} \alpha_n [\![y_n = -1]\!] = 2\sum_{n=1}^{N} \alpha_n [\![y_n = +1]\!] \quad — (1)$

even margin SVM: $\min_\alpha \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n\alpha_m y_n y_m x_n^T x_m - \sum_{n=1}^{N} \alpha_n$ same α_n $\forall n$

$$= \min_\alpha \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n\alpha_m y_n y_m x_n^T x_m - 2\sum_{n=1}^{N} \alpha_n [\![y_n = +1]\!]$$

uneven margin SVM: $\min_\alpha \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n\alpha_m y_n y_m x_n^T x_m - \left[P_+ \sum_{n=1}^{N} \alpha_n [\![y_n = +1]\!] + P_- \sum_{n=1}^{N} \alpha_n [\![y_n = -1]\!] \right]$

$$\overset{\text{use } (*)}{=} \min_\alpha \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n\alpha_m y_n y_m x_n^T x_m - \underbrace{\left[(P_+ + P_-)\sum_{n=1}^{N} [\![y_n = +1]\!] \alpha_n \right]}$$

$$:= \min_\alpha \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n\alpha_m y_n y_m x_n^T x_m - \sum_{n=1}^{N} \alpha_n^{New} \quad (2)$$

By (1): $\sum_{n=1}^{N} \alpha_n^{New} = 2\sum_{n=1}^{N} \alpha_n^{New} [\![y_n = +1]\!]$ in uneven SVM

$$\Rightarrow. \sum_{n=1}^{N} \alpha_n^{New} = 2\sum_{n=1}^{N} \alpha_n^{New} [\![y_n = +1]\!] \overset{\text{by } (2)}{=} \sum_{n=1}^{N} (P_+ + P_-)\alpha_n [\![y_n = +1]\!]$$

"α_n^* since it is optimal solution of even margin SVM

$$\Rightarrow. 2\alpha_n^{New} = (P_+ + P_-)\alpha_n^*$$

$$\alpha_n^{New} = \frac{P_+ + P_-}{2}\alpha_n^* \quad \forall n$$

$$\Rightarrow. \alpha^{New} = \frac{P_+ + P_-}{2}\alpha^*$$

(7) (d)

7. (d) $\log_2 K(x, x')$

Valid kernel must satisfies :

① Symmetric : $K(a,b) = K(b,a)$

② Matrix K must always be positive semi-definite (i.e all its eigenvalues are non-negative)

$K(x,x') = \phi(x)^T \phi(x') \in [0,2)$ (valid kernel function)

symmetric

Consider two one-dimensional vectors $x_1 = \frac{1}{2}$, $x_2 = \frac{1}{3}$, and $K(x,x') = (1+x^T x')^2$

$$K = \begin{bmatrix} K(x_1,x_1) & K(x_1,x_2) \\ K(x_2,x_1) & K(x_2,x_2) \end{bmatrix} = \begin{bmatrix} \frac{5}{4} & \frac{7}{6} \\ \frac{7}{6} & \frac{10}{9} \end{bmatrix} \leftarrow K(x_i,x_j) \in [0,2)$$

check: $\det\left(\begin{bmatrix} \frac{5}{4}-\lambda & \frac{7}{6} \\ \frac{7}{6} & \frac{10}{9}-\lambda \end{bmatrix} \right) = 0 \Rightarrow (\frac{5}{4}-\lambda)(\frac{10}{9}-\lambda) = \frac{49}{36} \Rightarrow \frac{50}{36} - \frac{85}{36}\lambda + \lambda^2 = \frac{49}{36}$

$\Rightarrow 36\lambda^2 - 85\lambda + 1 = 0 \Rightarrow \lambda = \frac{85 \pm \sqrt{7081}}{72} \Rightarrow \lambda_1 = \frac{85+\sqrt{7081}}{72} > 0$, $\lambda_2 = \frac{85-\sqrt{7081}}{72} > 0$ $\Rightarrow$ K is positive semi-definite

$\Rightarrow$ Consider $\hat{K}(x,x') = \log_2 K(x,x')$

$$\hat{K} = \begin{bmatrix} \log_2 \frac{5}{4} & \log_2 \frac{7}{6} \\ \log_2 \frac{7}{6} & \log_2 \frac{10}{9} \end{bmatrix}$$

Note: Since $K(x,x') = K(x',x)$

So $\hat{K}(x,x') = \log_2 K(x,x') = \log_2 K(x',x) = \hat{K}(x',x)$

$\Rightarrow$ Eigenvalue of $\hat{K}$:

$\det\left(\begin{bmatrix} \log_2\frac{5}{4}-\lambda & \log_2\frac{7}{6} \\ \log_2\frac{7}{6} & \log_2\frac{10}{9}-\lambda \end{bmatrix} \right) = 0 \Rightarrow \log_2\frac{5}{4}\log_2\frac{10}{9} - \lambda[\log_2\frac{5}{4} + \log_2\frac{10}{9}] + \lambda^2 - [\log_2\frac{7}{6}]^2 = 0$

$\Rightarrow \lambda^2 - \lambda[\log_2\frac{25}{18}] + \log_2\frac{5}{4}\log_2\frac{10}{9} - (\log_2\frac{7}{6})^2 = 0$ 0.4761367

$\Rightarrow \lambda_1 = \frac{\log_2\frac{25}{18} + \sqrt{(\log_2\frac{25}{18})^2 - 4(\log_2\frac{5}{4}\log_2\frac{10}{9} - (\log_2\frac{7}{6})^2)}}{2} = \frac{0.4739 + 0.4761}{2} > 0$

$\lambda_2 = \frac{\log_2\frac{25}{18} - \sqrt{(\log_2\frac{25}{18})^2 - 4(\log_2\frac{5}{4}\log_2\frac{10}{9} - (\log_2\frac{7}{6})^2)}}{2} = \frac{0.4739 - 0.4761}{2} < 0$

$\Rightarrow$ $\hat{K}$ is not positive semi-definite since it has negative eigenvalue !

So we found a counter-example s.t $\hat{K}(x,x') = \log_2 K(x,x')$ is not always a valid kernel !

for (c) : $2+K(x,x')$, we have $\lambda_1 = \frac{229+\sqrt{52009}}{72}$, $\lambda_2 = \frac{229-\sqrt{52009}}{72}$, both >0

(a) : $2^{K(x,x')}$, we have $\lambda_1 = 2^{\frac{5}{4}} + 2^{\frac{10}{9}} + [2^{\frac{7}{3}} + 2^{\frac{5}{2}} + 2^{\frac{20}{9}} - 2^{\frac{85}{36}}]^{\frac{1}{2}}$

$\lambda_2 =$ 〃 $-$ 〃 } both >0

(b) $(2-K(x,x'))^{-2}$, we have $\lambda_1 = \frac{1753 + 576\sqrt{\frac{1774319409}{2073600000}}}{1152}$, $\lambda_2 = \frac{1753 - 576\sqrt{\frac{1774319409}{2073600000}}}{1152}$ both >0

(e) $(K(x,x'))^2$, we have $\lambda_1 = \frac{3625+\sqrt{12627409}}{2592}$, $\lambda_2 = \frac{3625-\sqrt{12627409}}{2592}$, both >0

8) (c) 2

8. (c) 2

$K(x, x') = \exp(-\gamma \| \phi(x) - \phi(x') \|^2)$

If $\phi(x) = \phi(x')$, then $\phi(x)^T \phi(x) = K(x, x) = 1 \quad \forall \gamma > 0$

$$\| \phi(x) - \phi(x') \|^2 = \phi(x)^T \phi(x) - 2 \phi(x)^T \phi(x') + \phi(x')^T \phi(x')$$
$$= K(x, x) - 2 K(x, x') + K(x', x')$$
$$= 1 + 1 - \underbrace{2 e^{-\gamma \| x - x' \|^2}}_{\in [0, 2]}$$

as $\gamma \to \infty$, $x = x'$

$\Rightarrow \; 0 \le \| \phi(x) - \phi(x') \|^2 \le 2$

9) (d)

9. (d) $\frac{\ln(N-1)}{\varepsilon^2}$

$d = 1, b = 0 \Rightarrow h_{1,0}(x) = h(x) = \text{sign}\left(\sum_{n=1}^{N} y_n K(x_n, x) \right)$

$\Rightarrow E_{in}(\hat{h}) = \frac{1}{N} \sum_{i=1}^{N} [\![h(x_i) \neq y_i]\!] = \frac{1}{N} \sum_{i=1}^{N} [\![\text{sign}\left(\sum_{n=1}^{N} y_n K(x_n, x_i) \right) \neq y_i]\!]$

$\Rightarrow E_{in}(\hat{h}) = 0$ if $\text{sign}\left(\sum_{n=1}^{N} y_n K(x_n, x_i) \right) = y_i \quad \forall i$

when $i = 1$: $\sum_{n=1}^{N} y_n K(x_n, x_1) = \underline{y_1} + y_2 K(x_2, x_1) + \dots + y_N K(x_N, x_1)$

$i = 2$: $\sum_{n=1}^{N} y_n K(x_n, x_2) = y_1 K(x_1, x_2) + \underline{y_2} + \dots + y_N K(x_N, x_2)$

$\vdots$

$i = N$: $\sum_{n=1}^{N} y_n K(x_n, x_N) = y_1 K(x_1, x_N) + y_2 K(x_2, x_N) + \dots + \underline{y_N}$

$K(x, x') = e^{-\gamma \| x - x' \|^2} \to 0$ when γ large enough, then $\sum_{n=1}^{N} y_n K(x_n, x_i) \to y_i$

$\Rightarrow \; \sum_{i=1}^{N} \sum_{n=1}^{N} y_n K(x_n, x_i) \le \sum_{i=1}^{N} y_i + \sum_{i=1}^{N} y_i (N-1) e^{-\gamma \varepsilon^2}$ since $\| x - x' \|^2 \ge \varepsilon^2$

$\Rightarrow$ We need $\sum_{i=1}^{N} \sum_{n=1}^{N} y_n K(x_n, x_i) \to \sum_{i=1}^{N} y_i$

$\Rightarrow$ So, $\sum_{i=1}^{N} y_i \ge \sum_{i=1}^{N} y_i (N-1) e^{-\gamma \varepsilon^2}$ as γ large enough!

$\frac{1}{N-1} \ge e^{-\gamma \varepsilon^2}$

$\ln(N-1) \le \gamma \varepsilon^2 \Rightarrow \gamma \ge \frac{\ln(N-1)}{\varepsilon^2} \Rightarrow \gamma$ at least $\frac{\ln(N-1)}{\varepsilon^2}$

(10) (c)

10. (c) $\alpha_{t+1} \leftarrow \alpha_t$ except $\alpha_{t+1,n(t)} \leftarrow \alpha_{t,n(t)} + y_{n(t)}$

When the current w_t makes a mistake on $(\phi(x_{n(t)}), y_{n(t)})$

$\Rightarrow$ update w_t to w_{t+1}:

$$w_{t+1} = w_t + y_{n(t)}\phi(x_{n(t)}) = \sum_{i=1}^{N} \alpha_{t,i}\,\phi(x_i) + y_{n(t)}\phi(x_{n(t)})$$

$$= \alpha_{t,1}\phi(x_1) + \alpha_{t,2}\phi(x_2) + \dots + \underbrace{[\alpha_{t,n} + y_{n(t)}]\phi(x_n)} + \dots + \alpha_{t,N}\phi(x_N)$$

Also, by definition:

$$w_{t+1} = \sum_{n=1}^{N} \alpha_{t+1,n}\,\phi(x_n) = \alpha_{t+1,1}\phi(x_1) + \alpha_{t+1,2}\phi(x_2) + \dots + \underbrace{\alpha_{t+1,n}\,\phi(x_n)} + \dots + \alpha_{t+1,N}\phi(x_N)$$

Each α_t updates to α_{t+1}, except the mistake terms, which $\alpha_{t,n} + y_{n(t)}$ update to $\alpha_{t+1,n}$

$\Rightarrow$ $\alpha_{t+1} \leftarrow \alpha_t$ except $\alpha_{t+1,n(t)} \leftarrow \alpha_{t,n} + y_{n(t)}$

(11) (a)

11. (a) $\sum_{n=1}^{N} \alpha_{t,n} K(x_n, x)$

$w_t = \sum_{n=1}^{N} \underbrace{\alpha_{t,n}}_{\text{Scalar}}\, \underbrace{\phi(x_n)}_{\text{Vector}} \iff w_t^T = \sum_{n=1}^{N} \alpha_{t,n}\,\phi(x_n)^T$ ← e.g. $aB = aB^T$ for Scalar a, Vector B

$\Rightarrow$ $w_t^T \phi(x) = \left[\sum_{n=1}^{N} \alpha_{t,n}\,\phi(x_n)^T\right]\phi(x) = \sum_{n=1}^{N} \alpha_{t,n}\,\phi(x_n)^T\phi(x) = \underline{\underline{\sum_{n=1}^{N} \alpha_{t,n} K(x_n, x)}}$

e.g. $(\Sigma a_i A_i) B = (a_1 A_1 + a_2 A_2) B = a_1 A_1 B + a_2 A_2 B = \Sigma a_i A_i B$

(12) (b)

12. (b) $\min_{n: y_n > 0} \left(1 - \sum_{m=1}^{N} y_m \alpha_m K(x_n, x_m) \right)$

Consider the complementary slackness and condition

$\alpha_n (1 - \xi_n - y_n(w^T z_n + b)) = 0$, $\xi_n \geq 0 \ \forall n$

And sine we have optimal α^* s.t. $\alpha_n^* = C \ \forall n$

$\Rightarrow$. $\alpha_n^* (1 - \xi_n - y_n(w^T z_n + b^*)) = 0$

$\Rightarrow$. $1 - \xi_n - y_n(w^T z_n + b^*) = 0 \Rightarrow$. $\xi_n = 1 - y_n(w^T z_n + b^*) \geq 0$

$\Rightarrow$. $1 \geq y_n(w^T z_n + b^*)$

case ①. if $y_n > 0 \Rightarrow$. $b^* \leq \frac{1}{y_n} - w^T z_n = \frac{1}{y_n} - \sum_{m=1}^{N} y_m \alpha_m K(x_n, x_m)$

Case ②: if $y_n < 0 \Rightarrow$. $b^* \geq \frac{1}{y_n} - w^T z_n$

We reject case ② since b^* no upper bound in case ② !!

$\Rightarrow$. Case ①: $b^* \leq \underbrace{\frac{1}{y_n} - \sum_{m=1}^{N} y_m \alpha_m K(x_n, x_m)}$ $\rightarrow \therefore b^*$ has n choices $n = 1, 2, \ldots, N$

$\Rightarrow$. $b^* = \min_{n: y_n > 0} \left[1 - \sum_{m=1}^{N} y_m \alpha_m K(x_n, x_m) \right]$ ※.

13) (e)

13. (e) $K(x_n, x_m) + \frac{1}{2C}[\![n=m]\!]$

$\min_{w,b,\xi} \frac{1}{2} w^T w + C\sum_{n=1}^{N} \xi_n^2$ subject to $y_n(w^T\phi(x_n)+b) \geq 1-\xi_n$ for $n=1,2,\dots,N$

$\Rightarrow L(b,w,\xi,\alpha) = \frac{1}{2}w^Tw + C\sum_{n=1}^{N}\xi_n^2 + \sum_{n=1}^{N}\alpha_n[1-\xi_n - y_n(w^T\phi(x_n)+b)]$

$\Rightarrow$ Need to $\max_{\alpha_n \geq 0}\left(\min_{b,w,\xi} L(b,w,\xi,\alpha)\right)$ subject to some conditions.

$\frac{\partial L}{\partial b} = 0 = -\sum_{n=1}^{N}\alpha_n y_n \Rightarrow \sum_{n=1}^{N}\alpha_n y_n = 0 \rightsquigarrow$ Condition!

$\Rightarrow L(b,w,\xi,\alpha) = L(w,\xi,\alpha) = \frac{1}{2}w^Tw + C\sum_{n=1}^{N}\xi_n^2 + \sum_{n=1}^{N}\alpha_n[1-\xi_n - y_n(w^T\phi(x_n))]$

$\frac{\partial L}{\partial \xi_n} = 0 = 2C\xi_n - \alpha_n \Rightarrow \xi_n = \frac{\alpha_n}{2C}$

$\Rightarrow L(w,\xi,\alpha) = L(w,\alpha) = \frac{1}{2}w^Tw + \underbrace{C\sum_{n=1}^{N}\frac{\alpha_n^2}{(2C)^2} - 2C\sum_{n=1}^{N}\frac{\alpha_n^2}{(2C)^2}} + \sum_{n=1}^{N}\alpha_n[1-y_n(w^T\phi(x_n))]$

$= \frac{1}{2}w^Tw - \frac{1}{C}\sum_{n=1}^{N}\frac{\alpha^2}{2^2} + \sum_{n=1}^{N}\alpha_n[1-y_n(w^T\phi(x_n))]$

$\frac{\partial L}{\partial w_i} = 0 \Rightarrow w = \sum_{n=1}^{N}\alpha_n y_n\phi(x_n)$

$\Rightarrow L(w,\alpha) = \frac{1}{2}w^Tw - \frac{1}{2}\frac{1}{2C}\sum_{n=1}^{N}\alpha^2 + \sum_{n=1}^{N}\alpha_n - w^Tw = -\frac{1}{2}w^Tw - \frac{1}{C}\sum_{n=1}^{N}\frac{\alpha^2}{2^2} + \sum_{n=1}^{N}\alpha_n$

$= -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m \phi(x_n)^T\phi(x_m) - \frac{1}{2}\frac{1}{2C}\sum_{n=1}^{N}\sum_{m=1}^{N}[\![n=m]\!]\alpha_n\alpha_m y_n y_m + \sum_{n=1}^{N}\alpha_n$

$\Rightarrow \max_{\alpha} L(\alpha) \Leftrightarrow \min_{\alpha} -L(\alpha) = \min_{\alpha}\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m\left[\phi(x_n)^T\phi(x_m) + \frac{1}{2C}[\![n=m]\!]\right] + \sum_{n=1}^{N}\alpha_n$

$= \min_{\alpha}\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m\underline{\left[K(x_n,x_m) + \frac{1}{2C}[\![n=m]\!]\right]} + \sum_{n=1}^{N}\alpha_n$

subject to $\sum_{n=1}^{N}y_n\alpha_n = 0$, $\alpha_n \geq 0$ for $n=1,2,\dots,N$

Note: $y_n y_m = 1$ when $n=m$

14) (e)

14. (e) $\frac{1}{2C}\alpha^*$

(P_2) written in Lagrange Dual form: $L(b,w,\xi,\alpha)$

find best ξ^*: $\frac{\partial L}{\partial \xi_n} = 0 = \xi_n = \frac{\alpha_n}{2C}$

$\Rightarrow \xi = \begin{bmatrix}\xi_1\\ \vdots \\ \xi_n\end{bmatrix} = \frac{1}{2C}\begin{bmatrix}\alpha_1\\ \vdots\\ \alpha_n\end{bmatrix} = \frac{1}{2C}\alpha \rightsquigarrow$ optimal: $\alpha = \alpha^*$

$\Rightarrow \xi^* = \underline{\underline{\frac{1}{2C}\alpha^*}}$

15) (d) 8.5

```
# 15 (d) 8.5
y, x = svm_read_problem('C:/Users/USER/Desktop/satimage_train.txt')

for i in range(len(y)) :
    if y[i] != 3:
        y[i] = -1.0

prob = svm_problem(y, x) #,isKernel = True  # Defalt is False
m = svm_train(prob, '-t 0 -c 10')
p_labels, p_acc, p_vals = svm_predict(y, x, m)
ACC, MSE, SCC = evaluations(y, p_labels)

alpha = m.get_sv_coef()
Ind = m.get_SV()

SV_Val = []
for j in range(len(Ind)):
    SV = list(Ind[j].keys())
    Index = [i+1 for i in range(36)]
    Miss = [i for i in Index if i not in SV]
    Target = list(Ind[j].values())
    for i in Miss:
        Target.insert(i-1,0)
    SV_Val.append(Target)

W = np.array([alpha[i][0] for i in range(len(alpha))]).dot(SV_Val)
norm_W = np.sqrt(np.array([np.power(i,2) for i in W]).dot(np.array([1 for i in range(36)])))
print(norm_W)
```

```
Accuracy = 95.8061% (4249/4435) (classification)
8.459972213043251
```

16) (b) 2 vs not 2

17) (c) 700

```python
# 16  (b) 2 vs not 2 (Mininum is 0)
# 17  (c) 700 (maximum is 711)

for i in range(5):
    y, x = svm_read_problem('C:/Users/USER/Desktop/satimage_train.txt')
    print(f'{i+1} vs not {i+1}')
    for j in range(len(y)):
        if y[j] != i+1:
            y[j] = -1.0
    prob = svm_problem(y, x)
    m = svm_train(prob, '-t 1 -c 10 -g 1 -d 2 -r 1')
    p_label, (acc,mse,u), p_val = svm_predict(y, x, m)
    print('E_in = ', 1-acc*0.01)
    print('Number of SV :',m.get_nr_sv())
```

```
1 vs not 1
Accuracy = 99.9324% (4432/4435) (classification)
E_in =  0.0006764374295377129
Number of SV : 145
2 vs not 2
Accuracy = 100% (4435/4435) (classification)
E_in =  0.0
Number of SV : 87
3 vs not 3
Accuracy = 97.7678% (4336/4435) (classification)
E_in =  0.022322435174746302
Number of SV : 433
4 vs not 4
Accuracy = 95.9865% (4257/4435) (classification)
E_in =  0.040135287485907556
Number of SV : 711
5 vs not 5
Accuracy = 99.3236% (4405/4435) (classification)
E_in =  0.006764374295377684
Number of SV : 258
```

18) (d) 10

```python
# 18 (d) 10  and (e) 100
y, x = svm_read_problem('C:/Users/USER/Desktop/satimage_train.txt')
y_test, x_test = svm_read_problem('C:/Users/USER/Desktop/satimage.scale.t')
for i in range(len(y)) :
    if y[i] != 6:
        y[i] = 0

for i in range(len(y_test)) :
    if y_test[i] != 6:
        y_test[i] = 0

C = [0.01,0.1,1,10,100]
for i in C:
    print(f'when C = {i}')
    prob = svm_problem(y, x)
    m = svm_train(prob, f'-t 2 -c {i} -g 10 ')
    print('Train acc:')
    p_label, (acc,mse,u), p_val = svm_predict(y, x, m)
    print('Test acc:')
    p_label, (acc,mse,u), p_val = svm_predict(y_test, x_test, m)
    print('E_out = ', 1-acc*0.01,'\n')
```

```
when C = 0.01
Train acc:
Accuracy = 76.5953% (3397/4435) (classification)
Test acc:
Accuracy = 76.5% (1530/2000) (classification)
E_out =  0.235

when C = 0.1
Train acc:
Accuracy = 84.6449% (3754/4435) (classification)
Test acc:
Accuracy = 83.65% (1673/2000) (classification)
E_out =  0.16349999999999998

when C = 1
Train acc:
Accuracy = 99.9324% (4432/4435) (classification)
Test acc:
Accuracy = 89.35% (1787/2000) (classification)
E_out =  0.10650000000000004

when C = 10
Train acc:
Accuracy = 100% (4435/4435) (classification)
Test acc:
Accuracy = 90.3% (1806/2000) (classification)
E_out =  0.09699999999999998

when C = 100
Train acc:
Accuracy = 100% (4435/4435) (classification)
Test acc:
Accuracy = 90.3% (1806/2000) (classification)
E_out =  0.09699999999999998
```

19) (b) 1

```python
# 19 (b) 1
y, x = svm_read_problem('C:/Users/USER/Desktop/satimage_train.txt')
y_test, x_test = svm_read_problem('C:/Users/USER/Desktop/satimage.scale.t')
for i in range(len(y)) :
    if y[i] != 6:
        y[i] = 0

for i in range(len(y_test)) :
    if y_test[i] != 6:
        y_test[i] = 0

Gam = [0.1,1,10,100,1000]
for i in Gam:

    print(f'when gamma = {i}')
    prob = svm_problem(y, x) #,isKernel = True  # Defalt is False
    m = svm_train(prob, f'-t 2 -c 0.1 -g {i} ')
    print('Train acc:')
    p_label, (acc,mse,u), p_val = svm_predict(y, x, m)
    print('Test acc:')
    p_label, (acc,mse,u), p_val = svm_predict(y_test, x_test, m)
    print('E_out = ', 1-acc*0.01,'\n')
```

```
when gamma = 0.1
Train acc:
Accuracy = 92.0406% (4082/4435) (classification)
Test acc:
Accuracy = 90.15% (1803/2000) (classification)
E_out =  0.09850000000000003

when gamma = 1
Train acc:
Accuracy = 93.4386% (4144/4435) (classification)
Test acc:
Accuracy = 93% (1860/2000) (classification)
E_out =  0.06999999999999995

when gamma = 10
Train acc:
Accuracy = 84.6449% (3754/4435) (classification)
Test acc:
Accuracy = 83.65% (1673/2000) (classification)
E_out =  0.16349999999999998

when gamma = 100
Train acc:
Accuracy = 76.5953% (3397/4435) (classification)
Test acc:
Accuracy = 76.5% (1530/2000) (classification)
E_out =  0.235

when gamma = 1000
Train acc:
Accuracy = 76.5953% (3397/4435) (classification)
Test acc:
Accuracy = 76.5% (1530/2000) (classification)
```



```
E_out =  0.235
```

20) (b) 1

```
# 20 (b) 1
y, x = svm_read_problem('C:/Users/USER/Desktop/satimage_train.txt')
for i in range(len(y)) :
    if y[i] != 6:
        y[i] = 0

ADD = [0,0,0,0,0]
i = 0
while i != 1000:

    XY = [[y[j],x[j]] for j in range(len(y))]
    Val = random.sample(XY,200)
    Val_x = [Val[j][1] for j in range(len(Val))]
    Val_y = [Val[j][0] for j in range(len(Val))]

    Tra = [j for j in XY if j not in Val ]
    Tra_x = [Tra[j][1] for j in range(len(Tra))]
    Tra_y = [Tra[j][0] for j in range(len(Tra))]

    prob = svm_problem(Tra_y, Tra_x)

    ACC_all = []
    for j in [0.1,1,10,100,1000]:
        m1 = svm_train(prob, f'-t 2 -c 0.1 -g {j} ')
        (acc1,mse1,u1) = svm_predict(Val_y, Val_x, m1)[1]
        ACC_all.append(1-acc1*0.01)
    IND = ACC_all.index(min(ACC_all))
    for j in range(len(ADD)):
        if IND == j:
            ADD[j] +=1
    i+=1
print(ADD)
```

```
[278, 722, 0, 0, 0]
```