ML Hw5 B06611032 武敬祥

1. [d]

$D_{(x)} = \{(-2,-1),(0,1),(2,-1)\}$

$\phi(x)_2 = [1, x, x^2]^T$

$D_{(z)} = \{(1,-2,4,-1),(1,0,0,1),(1,2,4,-1)\}$

$Z = \begin{bmatrix} 1 & -2 & 4 \\ 1 & 0 & 0 \\ 1 & 2 & 4 \end{bmatrix} \quad y = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}$

$\because y_n(w^T x_n + b) \geq 1$

$-(b - 2w_1 + 4w_2) \geq 1 \qquad -b + 2w_1 - 4w_2 \geq 1$

$b \geq 1 \qquad b \geq 1$

$-(b + 2w_1 + 4w_2) \geq 1 \qquad -b - 2w_1 - 4w_2 \geq 1$

$2w_1 - 4w_2 \geq 2 \qquad -2w_1 - 4w_2 \geq 2$

$w_1 - 2w_2 \geq 1 \qquad -w_1 - 2w_2 \geq 1$

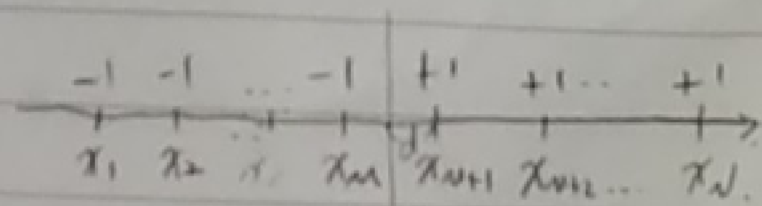
$2w_1 \geq 0, \; w_1 \geq 0. \times \qquad w_1 = 0, \; w_2 = \frac{-1}{2}, \; b = 1$

$-4w_2 \geq 2 \quad w_2 \leq \frac{-1}{2}$

2. [b]

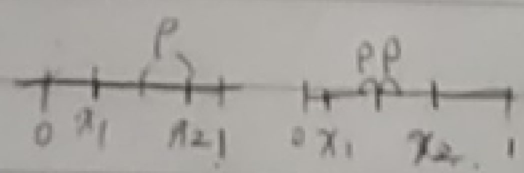
$\text{max margin} = \frac{1}{\|w\|} = \frac{1}{\sqrt{w^T w}} = \frac{1}{\sqrt{\frac{1}{4}}} = 2$

3. [e]



$d = \frac{1}{2}(x_{M+1} - x_M)$

4.



$\rho \in [0, 0.5]$

$\rho \in [0, \frac{x_2 - x_1}{2}]$

5. [C]

$$\mathcal{L} = \frac{1}{2}w^T w + \sum_{n=1}^{N} \alpha_n \Big[[\![y_n = 1]\!] \cdot (\rho^+ - y_n(w^T x_n + b)) + [\![y_n = -1]\!] \cdot (\rho^- - y_n(w^T x_n + b)) \Big]$$

$$\mathcal{L} = \frac{1}{2}w^T w + \alpha_n \sum_{n=1}^{N} \Big[[\![y_n = 1]\!] \cdot \rho^+ + [\![y_n = -1]\!] \cdot \rho^- - y_n(w^T x_n + b) \Big]$$

① $\frac{\partial \mathcal{L}(b, w, \alpha)}{\partial b} = 0 = -\sum_{n=1}^{N} \alpha_n y_n \Rightarrow \sum_{n=1}^{N} \alpha_n y_n = 0$

② $\frac{\partial \mathcal{L}(b, w, \alpha)}{w_i} = 0, = w_i - \sum_{n=1}^{N} \alpha_n y_n x_{n,i} \Rightarrow w = \sum_{n=1}^{N} \alpha_n y_n x_n$

$$SVM = \max\Big(-\frac{1}{2}w^T w + \sum_{n=1}^{N} \alpha_n [\![y_n = 1]\!] \cdot \rho^+ + [\![y_n = -1]\!] \cdot \rho^- \Big)$$

$$SVM = \min\Big(\frac{1}{2}w^T w - \sum_{n=1}^{N} \alpha_n ([\![y_n = 1]\!] \cdot \rho^+ + [\![y_n = -1]\!] \cdot \rho^-) \Big)$$

$$= \min\Big(\frac{1}{2}w^T w - \sum_{n=1}^{N} \alpha_n [\![y_n = 1]\!] \cdot \rho^+ - \sum_{n=1}^{N} \alpha_n [\![y_n = -1]\!] \cdot \rho^- \Big)$$

6. [a]

diffrent value of ρ doesn't change the SV
in other words, the slope of hyperplane doesn't change.
$w^* = \sum_{SV} \alpha_n^* y_n x_n$ => the w doesn't change
=> the α_n doesn't change.

7.

$K_1(x, x') = \phi_1(x)^T \phi_1(x') = 1 + x^T x' + (x^T x')(x^T x')$

$K_2(x, x') = (\zeta + \gamma x^T x')^2 = \zeta^2 + 2\zeta\gamma x^T x' + \gamma^2 x^T x'$

(a) $2^{K(x,x')} = 2^{\zeta^2} \cdot 2^{2\zeta\gamma x^T x'} \cdot 2^{\gamma^2 x^T x'}$

(d) $\log_2 K(x,x') = \log_2(\zeta + \gamma x^T x') + \log_2(\zeta + \gamma x^T x')$

8. [c]

$K_G(x, x') = \phi(x)^T \phi(x') = \exp(-\gamma \|x - x'\|^2)$

the distance^2 in Z-space:

$\|\phi(x) - \phi(x')\|^2 = \phi(x)^T \phi(x) - 2\phi(x)^T \phi(x') + \phi^T(x')\phi(x')$

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

$\|\phi(x) - \phi(x')\|^2 = 1 + 1 - 0$

$= 2$

9. [d]

$h_{\alpha,b}(x) = \text{sign}(\sum_{n=1}^{N} y_n \alpha_n K(x_n, x) + b)$

$h_{1,0}(x) = \text{sign}(\sum_{n=1}^{N} y_n K(x_n, x))$

$h_{1,0}(x) = \text{sign}(\sum_{n=1}^{N} y_n e^{(-\gamma\|x_n - x\|^2)})$

assume (x_n, y_n) . $n \in [0, 1, \dots N]$

$h_{1,0}(x_k) = \text{sign}(y_1 e^{-\gamma\|x_1 - x_k\|^2} + y_2 e^{-\gamma\|x_2 - x_k\|^2} + \dots y_N e^{-\gamma\|x_N - x_k\|^2})$

when $n = 1$, $E_{in} = 0$, $y_1 \in \{-1, 1\}$

$h_{1,0}(x_1) = \text{sign}(y_1 + y_2 e^{-\gamma\|x_2 - x_1\|^2} + \dots y_N e^{-\gamma\|x_N - x_1\|^2})$

if $|y_1| > |y_2 e^{-\gamma\|x_2 - x_1\|^2} \dots| = |\sum_{n=2}^{N} y_n e^{-\gamma\|x_n - x_1\|^2}|$

assure that $h_{1,0}(x_1) = y_1$

$\because \|x_n - x_1\|^2 \geq \varepsilon^2$

$|y_1| > |\sum_{n=2}^{N} y_n e^{-\gamma\varepsilon^2}| \geq |\sum_{n=2}^{N} y_n e^{-\gamma\|x_n - x_1\|^2}|$

worst case y_n $n \in [2, 3 \dots N]$ is all positive/negative

and y_1 is negative/positive

$1 > (N-1) \cdot 1 e^{-\gamma\varepsilon^2}$

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

$\ln\frac{1}{N-1} > -\gamma\varepsilon^2$

$-\ln(N-1) > -\gamma\varepsilon^2$

$\gamma > \frac{\ln(N-1)}{\varepsilon}$

Date :

10. [C]

$$w_{t+1} \leftarrow w_t + y_{n(t)}\,\phi(x_{n(t)})$$

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

when current w_t makes a mistake.

$$y_{n(t)} w_t\, \phi(x_{n(t)}) < 0 \quad \Rightarrow y_n w_{t+1}\, \phi(x_n) > 0.$$

$$y_n(t) \sum_{n=1}^{N} \alpha_{t,n} \cdot \phi(x_n) < 0. \quad \Rightarrow y_n \sum_{n=1} -\alpha_{t,n} \cdot \phi(x_n) > 0.$$

$$\sum_{n=1}^{N} \alpha_{t+1,n}\, \phi(x_n) = \sum_{n=1}^{N} \alpha_{t,n} \cdot \phi(x_n) + y_n\, \phi(x_n)$$

$$\sum_{n=1}^{N} \alpha_{t+1,n}\, \phi(x_n) = \sum_{n=1}^{N} \alpha_{t,n} \cdot \phi(x_n) \left\{ \begin{matrix} \text{if } n = \text{mistake sample.} \\ (\alpha_{t,n} + y_n(t))\, \phi(x_n) \end{matrix} \right\}$$

11. [a]

$$w_t = \sum_{n=1}^{N} \alpha_{t,n} \cdot \phi(x_n), \quad K(x, x') = \phi(x)^T \phi(x'),$$

$$w_t^T \phi(x) = \left(\sum_{n=1}^{N} \alpha_{t,n} \cdot \phi(x_n) \right)^T \cdot \phi(x)$$

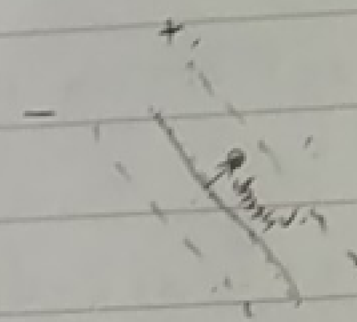
$$= \sum_{n=1}^{N} \alpha_{t,n} \cdot K(x_n, x)$$

No.
Date

$\alpha_n y_n z_n$

12. [b]

bounded SV $\Rightarrow \alpha_n = C$



$b = y - y\xi - w^T z$

$= y_{sv} - y_{sv}\xi_{sv} - \sum_{n=1}^{N} \alpha_n y_n K(x_m, x_{sv})$

$= y_{sv} - y_{sv}\xi_{sv} - \sum_{m=1}^{N} y_m \alpha_m K(x_m, x_{sv})$

$= y_{sv}(1-\xi_{sv}) - \sum_{m=1}^{N} y_m \alpha_m K(x_{sv}, x_m)$

if $y_{sv} = 1$, $b = 1 - \xi_{sv} - \sum_{m=1}^{N} y_m \alpha_m K(x_{sv}, x_m) < 1 - \sum_{m=1}^{N} y_m \alpha_m K(x_{sv}, x_m)$

$y_{sv} = -1$, $b = -(1-\xi_{sv}) - \sum_{m=1}^{N} y_m \alpha_m K(x_{sv}, x_m) > -1 - \sum_{m=1}^{N} y_m \alpha_m K(x_{sv}, x_m)$

$b^* < \min_{y_n=1, n=1,2\ldots N} \left(1 - \sum_{m=1}^{N} y_m \alpha_m K(x_{sv}, x_m)\right)$

13. [e]

$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{n=1}^{N} \xi_n^2$

$\xi_1^2 + \xi_2^2 + \xi_3^2 \cdots \xi_N^2$

subject to. $y_n(w^T \phi(x_n) + b) \geq 1 - \xi_n$

$\geq 1 \ \xi_n$

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

$\max_{\alpha_n \geq 0} \left(\min_{b, w, \xi} L(b, w, \xi, \alpha) \right)$

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

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

$\Rightarrow \sum_{n=1}^{N} \xi_n (C\xi_n - \alpha_n)$

$\Rightarrow \sum_{n=1}^{N} \xi_n (2C\xi_n - \alpha_n)$

No.
Date

$$\max_{\alpha_n \ge 0,\ 2C\xi_n = \alpha_n} \left(\min_{b,w} \frac{1}{2}w^T w + \sum_{n=1}^{N} \alpha_n (1 - y_n(w^T \phi(x_n) + b)) - \sum_{n=1}^{N} \frac{1}{4C}\alpha_n^2 \right)$$

$$\xi_n = \frac{1}{2C}\alpha_n.$$

$$\max_{\alpha_n \ge 0\ 2C\xi_n - \alpha_n = 0} \left(\min_{b,w} \frac{1}{2}w^T w + \sum_{h=1}^{N} \alpha_n (1 - y_n(w^T\phi(x_n) + b)) - \sum_{n=1}^{N} \frac{1}{4C}\alpha_n^2 \right)$$

$$\frac{\partial L(b,w,\alpha)}{\partial b} = -\sum_{h=1}^{N} \alpha_n y_n = 0,$$

$$\max_{\alpha_n \ge 0,\ 2C\xi_n - \alpha_n = 0,\ \sum y_n\alpha_n = 0} \left(\min_{w} \frac{1}{2}w^T w + \sum_{n=1}^{N} a_n (1 - y_n(w^T\phi(x_n))) - \sum_{h=1}^{N} \frac{1}{4C}\alpha_n^2 \right)$$

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

$$\max \left(\frac{1}{2} w^T w + \sum_{n=1}^{N} \alpha_n - w^T w - \sum_{n=1}^{N} \frac{1}{4C}\alpha_n^2 \right)$$

$$= \min_{\alpha} \frac{1}{2}\sum_{h=1}^{N}\sum_{m=1}^{N} \alpha_n\alpha_m y_n y_m z_n^T z_m - \sum_{h=1}^{N}\alpha_n + \frac{1}{4C}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n\alpha_m [\![n=m]\!]$$

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

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

14. [e]

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

$$\xi_n = \frac{1}{2C}\alpha_n$$

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

15

```python
from libsvm.python.svmutil import*
import numpy as np

train_y , train_x = svm_read_problem("satimage_train.txt")
W = np.zeros(36)
W_square = 0
W_norm = 0

for i in range(len(train_y)):
    if train_y[i] != 3:
        train_y[i] = -1
    else:
        train_y[i] = 1

model = svm_train(train_y , train_x ,"-c 10 -s 0 -t 0 -q")
nSV = model.get_nr_sv()
alpha = model.get_sv_coef()
SV = model.get_SV()

for i in range(nSV):
    X = np.zeros(36)
    for j in range(1,37):
        X[j-1] = SV[i].get(j,0)
    W += alpha[i][0] * X

for i in range(36):
    W_square += W[i] ** 2
W_norm = W_square ** 0.5
print(W_norm)
```

16.

```
from libsvm.python.svmutil import*
import numpy as np

choice = range(1,6)
for number in choice:
    train_y , train_x = svm_read_problem("satimage_train.txt")

    W = np.zeros(36)
    W_square = 0
    W_norm = 0

    for i in range(len(train_y)):
        if train_y[i] != number:
            train_y[i] = -1
        else:
            train_y[i] = 1

    model = svm_train(train_y , train_x ,"-c 10 -s 0 -t 1 -d 2 -q")
    p_labs, p_acc, p_vals = svm_predict(train_y, train_x, model)

    print(number," versus not ", number , " : " , 100-p_acc[0])
```

17.

```
from libsvm.python.svmutil import*
import numpy as np

choice = range(1,6)
for number in choice:
    train_y , train_x = svm_read_problem("satimage_train.txt")

    W = np.zeros(36)
    W_square = 0
    W_norm = 0

    for i in range(len(train_y)):
        if train_y[i] != number:
            train_y[i] = -1
        else:
            train_y[i] = 1

    model = svm_train(train_y , train_x ,"-c 10 -s 0 -t 1 -d 2 -q")
    nSV = model.get_nr_sv()

    print(number," versus not ", number , " : " , nSV)
```

18.

```python
from libsvm.python.svmutil import*
import numpy as np

train_y , train_x = svm_read_problem("satimage_train.txt")
test_y , test_x = svm_read_problem("satimage_test.txt")

for i in range(len(train_y)):
    if train_y[i] != 6:
        train_y[i] = -1
    else:
        train_y[i] = 1
    if i < len(test_y):
        if test_y[i] != 6:
            test_y[i] = -1
        else:
            test_y[i] = 1

C = [0.01 , 0.1 , 1 , 10 , 100]
for c in C:
    model = svm_train(train_y , train_x , f"-c {c} -s 0 -t 2 -g 10 -q")
    p_labs, p_acc, p_vals = svm_predict(test_y, test_x, model)

    print(c , " E_out"," : " , 100-p_acc[0])
```

19.

```
from libsvm.python.svmutil import*
import numpy as np

train_y , train_x = svm_read_problem("satimage_train.txt")
test_y , test_x = svm_read_problem("satimage_test.txt")

for i in range(len(train_y)):
    if train_y[i] != 6:
        train_y[i] = -1
    else:
        train_y[i] = 1
    if i < len(test_y):
        if test_y[i] != 6:
            test_y[i] = -1
        else:
            test_y[i] = 1

Gamma = [0.1 , 1 , 10 , 100 , 1000]
for ga in Gamma:
    model = svm_train(train_y , train_x , f"-c 0.1 -s 0 -t 2 -g {ga} -
q")
    p_labs, p_acc, p_vals = svm_predict(test_y, test_x, model)

    print(ga , " E_out"," : " , 100-p_acc[0])
```

```
20.
from libsvm.python.svmutil import*
import numpy as np
from random import randint

Gamma_choice = [0,0,0,0,0]
times = 0
while times < 1000:
    print(times)
    train_y , train_x = svm_read_problem("satimage_train.txt")
    valid_x = []
    valid_y = []
    ran = []
    i = 0
    while i < 200:
        rand = randint(0,4435)
        # if rand >= i:
        #     rand = rand - i
        if rand not in ran:
            valid_x.append(train_x[rand-i])
            valid_y.append(train_y[rand-i])
            train_x.remove(train_x[rand-i])
            train_y.remove(train_y[rand-i])
            ran.append(rand)
            i += 1

    for i in range(len(train_y)):
        if train_y[i] != 6:
            train_y[i] = -1
        else:
            train_y[i] = 1
        if i < len(valid_y):
            if valid_y[i] != 6:
                valid_y[i] = -1
            else:
                valid_y[i] = 1

    Gamma = [0.1 , 1 , 10 , 100 , 1000]
```

```
    for ga in Gamma:
        min_Eval = 100
        best_gamma = None
        model = svm_train(train_y , train_x , f"-c 0.1 -s 0 -t 2 -
g {ga} -q")
        p_labs, p_acc, p_vals = svm_predict(valid_y, valid_x, model ,"-
q")
        if (100-p_acc[0]) < min_Eval:
            best_gamma = ga
            min_Eval = p_acc[0]

    if best_gamma == 0.1:
        Gamma_choice[0] += 1
    elif best_gamma == 1:
        Gamma_choice[1] += 1
    elif best_gamma == 10:
        Gamma_choice[2] += 1
    elif best_gamma == 100:
        Gamma_choice[3] += 1
    elif best_gamma == 1000:
        Gamma_choice[4] += 1
    times += 1

print(Gamma_choice)
```