Machine Learning (NTU, Fall 2023) HW5 R12631055 林東甫

1.

1. Let $h(x) = \text{sign}(x-\theta)$.

If h is a feasible solution, then $E_{in}(h) = 0$.

Consider that if θ is not between x_M and x_{M+1}, then $\exists\ \theta < x_n$, $y_n = -1$ or $\theta > x_n$, $y_n = +1$ s.t. $E_{in}(h) \neq 0$ $\Rightarrow\Leftarrow$ contradiction

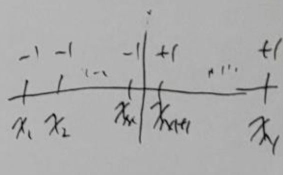
Therefore $x_M < \theta < x_{M+1}$, $\text{margin}(\theta) = \min_i |x_i - \theta| = \min(|x_{M+1}-\theta|, |x_M - \theta|)$

$\therefore \exists\ \epsilon \in [-\frac{1}{2}(x_{M+1} - x_M), \frac{1}{2}(x_{M+1} - x_M)]$,

s.t. $\theta = \frac{1}{2}(x_{M+1} + x_M) + \epsilon \in [x_M, x_{M+1})$

$\therefore$ if $\epsilon \geq 0$, $\text{margin}(\theta) = \frac{1}{2}(x_{M+1} - x_M) - \epsilon$

if $\epsilon < 0$, $\text{margin}(\theta) = \frac{1}{2}(x_{M+1} - x_M) + \epsilon$



$\therefore \text{margin}(\theta) = \frac{1}{2}(x_{M+1} - x_M) - |\epsilon| \leq \frac{1}{2}(x_{M+1} - x_M)$

There's optical solution where $\epsilon = 0$, $\hat{\theta} = \frac{1}{2}(x_M + x_{M+1})$

$\text{margin}(\hat{\theta}) = \frac{1}{2}(x_{M+1} - x_M)$ #

2.

2. $\min_{w,b} \frac{1}{2} w^T w$ subject to $(w^T x_n + b) \geq 1$ for $y_n = +1$

$-(w^T x_n + b) \geq \rho$ for $y_n = -1$

let $\mathcal{L}(b,w,a) = \frac{1}{2} w^T w + \sum_{n=1, y_n>0}^{N} \overset{[\![y_n=+1]\!]}{a_n}(1 - y_n(w^T x_n + b)) + \sum_{n=1, y_n<0}^{N} \overset{[\![y_n=-1]\!]}{a_n}(\rho - y_n(w^T x_n + b))$

Since dual problem : $\min_{b,w} \left(\max_{\text{all } a_n \geq 0} \mathcal{L}(b,w,a) \right) = \max_{\text{all } a_n \geq 0} \left(\min_{b,w} \mathcal{L}(b,w,a) \right)$

$\partial b \Rightarrow$

$$\frac{\partial \mathcal{L}(b,w,a)}{\partial b} = -\sum_{n=1}^{N} a_n [\![y_n=+1]\!] y_n - \sum_{n=1}^{N} a_n [\![y_n=-1]\!] y_n$$

$$= -\left(\sum_{n=1}^{N} a_n y_n ([\![y_n=+1]\!] + [\![y=-1]\!]) \right)$$

$$= 0$$

$\partial w \Rightarrow \frac{\partial \mathcal{L}(b,w,a)}{\partial w} = w - \sum_{n=1}^{N} a_n [\![y_n=+1]\!] y_n x_n - \sum_{n=1}^{N} a_n [\![y_n=-1]\!] y_n x_n$

$$= 0$$

Since $w = \sum_{n=1}^{N} a_n [\![y_n=+1]\!] y_n x_n - \sum_{n=1}^{N} a_n [\![y_n=-1]\!] y_n x_n$

$$\mathcal{L}(b,w,a) = -\frac{1}{2} w^T w + \sum_{n=1}^{N} a_n [\![y_n=+1]\!] + \sum_{n=1}^{N} a_n \rho [\![y_n=-1]\!]$$

That is : $\min -\left(-\frac{1}{2} \left\| \sum_{n=1}^{N} a_n y_n x_n \right\|^2 + \sum_{n=1}^{N} a_n [\![y_n=+1]\!] + \sum_{n=1}^{N} a_n \rho [\![y_n=-1]\!] \right)$

$\Rightarrow \min \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m y_n y_m x_n^T x_m - \sum_{n=1}^{N} a_n [\![y_n=+1]\!] - \sum_{n=1}^{N} a_n \rho [\![y_n=-1]\!]$

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

$a_n \geq 0 \;\forall\, n = 1, 2, \ldots, N$

3. If (b_1, w_1) is ~~the~~ a solution of even SVM:

$$\min_{w,b} \frac{1}{2} w^T w \quad \text{subject to } (w^T x_n + b) \geq 1 \text{ for } y_n = +1$$

$$-(w^T x_n + b) \geq 1 \text{ for } y_n = -1$$

the dual: $\min_{\alpha} \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m x_n^T x_m$

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

subject to $\sum_{n=1}^{N} y_n \alpha_n = 0 \quad \alpha_n \geq 0 \ \forall n = 1, 2, \cdots, N$

$$w_1 = \sum_{n=1}^{N} \alpha_{1n} y_n z_n = \sum_{SV} \alpha_{1n} y_n z_n$$

$b_1 = y_n - w_1^T z_n$ with any SV (z_n, y_n)

~~$w_{t+1} \leftarrow w_t + y_n \phi(x_{n(t)})$~~ $y_n = w_1^T x_n + b^*$

~~$w_t = \sum_{n=1}^{N} \alpha_{tn} \phi(x_n)$~~

uneven SVM

$w^T x_n + b = 1$ if $y_n = +1$ $\Rightarrow w^T x_n + b = (\frac{1}{2} + \frac{1}{2}\rho) w_1^T x_n$

$w^T x_n + b = \rho$ if $y_n = -1$

$\Rightarrow \alpha_{1126} = (\frac{1}{2} + \frac{1}{2}\rho)^{1126} \alpha_1$

4. Since α_1^* is an optimal solution of even SVM:

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

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

uneven SVM: $w^T x_n + b = 1$, if $y_n = +1$

$w^T x_n + b = \rho$, if $y_n = -1$

$$\Rightarrow w^{*T} x_n + b^* = \left(\frac{1}{2} + \frac{1}{2}\rho\right) w^{*T} x_n + \left(\frac{1+b^*}{2} - \frac{1-b^*}{2}\rho\right)$$

$$\Rightarrow w^T x_n = \left(\frac{1}{2} + \frac{1}{2}\rho\right) w^{*T} x_n, \quad \alpha = \left(\frac{1}{2}(1+\rho)\right)^n \alpha^*$$

$\forall \rho > 0$

5. Since $k_1(x, x') = \phi_1(x)^T \phi_1(x')$,

$k_2(x, x') = \phi_2(x)^T \phi_2(x')$ are valid kerne

$K = (x, x') = k_1(x, x') \cdot k_2(x, x')$

$= \phi_1(x)^T \phi_1(x') \cdot \phi_2(x)^T \phi_2(x')$

$= \phi_1(x)^T \phi_2(x)^T \cdot \phi_1(x') \phi_2(x') = \phi(x)^T \phi(x')$

$\therefore$ $k = \phi(x)^T \phi(x')$ is a valid kernel

6. the distance between x and x' is $\|\phi(x)-\phi(x')\|$

$\therefore$ the square distance is $\|\phi(x)-\phi(x')\|^2$ ~~$= k(x,x)$~~

~~$\|\phi(x)\| = \sqrt{k(x,x)}$~~

$\because \|\phi(x)\|^2 = \phi(x)^T\phi(x) = k(x,x)$, $\|\phi(x)\| = \sqrt{k(x,x)}$

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

$$= k(x,x) + k(x',x') - 2k(x,x')$$

Since $k_2(x,x') = (1+x^Tx')$ is always positive, $\therefore x, x'$ are unit vectors; let $x = \begin{bmatrix}0\\1\end{bmatrix}$, $x' = \begin{bmatrix}0\\-1\end{bmatrix}$

$$\|\phi(x)-\phi(x')\|^2 = \left(1+[0,1]\begin{bmatrix}0\\1\end{bmatrix}\right)^2 + \left(1+[0,-1]\begin{bmatrix}0\\-1\end{bmatrix}\right)^2 - 2\left(1+[0,1]\begin{bmatrix}0\\-1\end{bmatrix}\right)^2 = \underline{8} \text{ largest.}$$

Let $x = \begin{bmatrix}0\\1\end{bmatrix}$, $x' = \begin{bmatrix}0\\1\end{bmatrix}$

$$\|\phi(x)-\phi(x')\|^2 = \left(1+[0,1]\begin{bmatrix}0\\1\end{bmatrix}\right)^2 + \left(1+[0,1]\begin{bmatrix}0\\1\end{bmatrix}\right)^2 - 2\left(1+[0,1]\begin{bmatrix}0\\1\end{bmatrix}\right)^2 = \underline{0} \text{ smallest}$$

7

7. Let $\tilde{\phi}(x) = \left(1, \sqrt{\frac{2}{1!}}x, \sqrt{\frac{2^2}{2!}}x^2, \cdots\right)$

~~由~~ $e^{\sqrt{2}x} = 1 + \frac{\sqrt{2}}{1!}x + \frac{\sqrt{2}^2}{2!}x^2 + \cdots$

$\therefore$

$\phi(x) = \exp(-x^2)\cdot\left(1, \sqrt{\frac{2}{1!}}x, \cdots\right) = \exp(-x^2)\tilde{\phi}(x)$

$$\exp(-x^2) = \frac{\phi(x)}{\tilde{\phi}(x)} = \frac{\phi(x)}{\|\tilde{\phi}\|\cdot\|\phi\|} = \frac{1}{\|\tilde{\phi}(x)\|}$$

8. Let $\cos(x, x')$ measure the cosine of the angle θ. between $x, x' \in \mathbb{R}^d$. $\|x\|, \|x'\| \neq 0$

That is $\cos\theta = \frac{x \cdot x'}{\|x\|\|x'\|} = \frac{x_1 x_1' + x_2 x_2' + \cdots + x_d x_d'}{\|x\| \cdot \|x'\|}$

~~wi~~ which is linear, $\therefore$ $\cos(x, x')$ is valid kernel

9

```python
Clist = [0.1, 1, 10]
Qlist = [2, 3, 4]
minimun = 65535
c_minimun = 0
q_minimun = 0
for q in Qlist:
    for c in Clist:
        train_y , train_x = svm_read_problem("satimage.scale.txt")



        for i in range(len(train_y)):
            if train_y[i] != 4:
                train_y[i] = -1
            else:
                train_y[i] = 1

    p = svm_problem(train_y , train_x)
    sentence = svm_parameter(f"-t 1 -c {c} -d {q} -r 1 -g 1")

    model = svm_train(p ,sentence)
    SV = model.get_SV()
    if len(SV) < minimun:
        minimun = len(SV)
        c_minimun = c
        q_minimun = q

print("4 versus not 4" ,", C : ",c_minimun,", Q : ", q_minimun, "number of support vectors : ",minimun)
```

✓ 2.4s

```
4 versus not 4 , C :  10 , Q :  4 number of support vectors :  629
```

(C,Q)=(10,4) 629

10

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

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

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

✓ 1.8s

```
Accuracy = 95.4% (1908/2000) (classification)
0.01  E_out  :  4.6000000000000085
Accuracy = 98.8% (1976/2000) (classification)
0.1  E_out  :  1.2000000000000028
Accuracy = 99.5% (1990/2000) (classification)
1  E_out  :  0.5
Accuracy = 99.4% (1988/2000) (classification)
10  E_out  :  0.5999999999999943
Accuracy = 99.45% (1989/2000) (classification)
100  E_out  :  0.5499999999999972
```

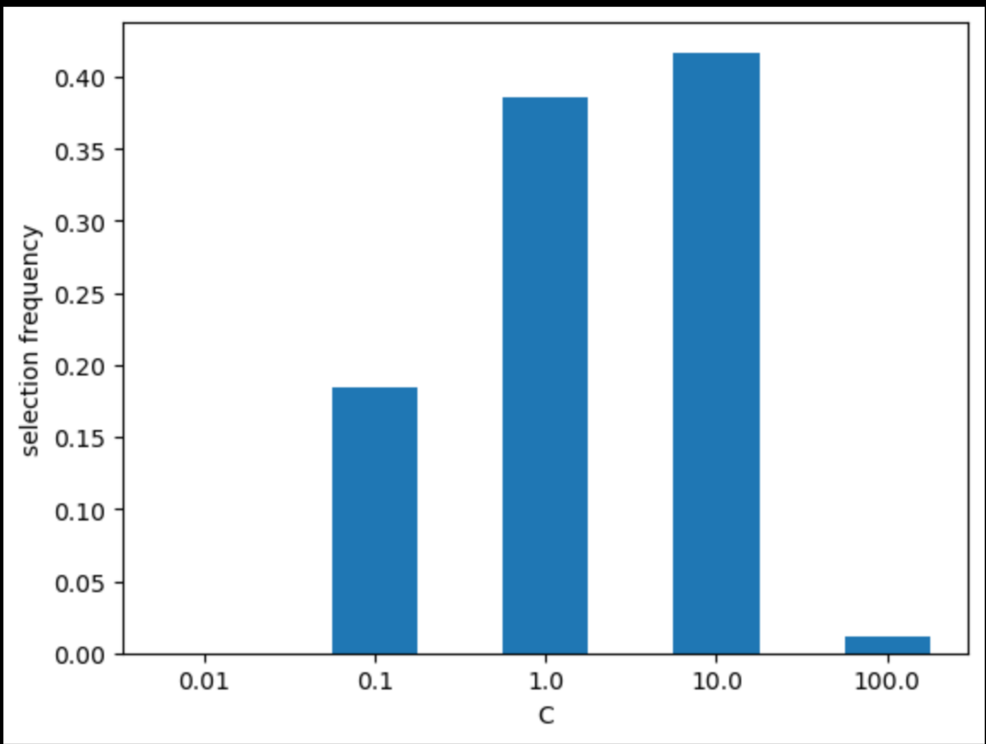
C=1 Eout=0.5

11.其實這裡的分配與上一題的 Accu 有一點相似

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

train_x = x[200:]
train_y = y[200:]
valid_x = x[:200]
valid_y = y[:200]

for c in C:
    p = svm_problem(train_y, train_x)
    sentence = svm_parameter(f'-t 2 -c {c} -r 1 -g 1')
    model = svm_train(p, sentence)
    p_labs, p_acc, p_vals = svm_predict(valid_y, valid_x, model)


    if p_acc[0]/100 >= acc_max:
        if p_acc[0]/100 == acc_max and c < c_min:
            c_min = c
        elif p_acc[0]/100 == acc_max and c > c_min:
            continue
        else:
            acc_max = p_acc[0]/100
            c_min = c

C_count[C.index(c_min)] += 1
print(f"c = {c_min}")
```

```
numbers of C = [0.01, 0.1, 1, 10, 100] : [0, 185, 386, 417, 12]
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...



12.取過 log 以後,幾乎就是很完美的正比

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

p = svm_problem(train_y, train_x)
sentence = svm_parameter(f'-t 2 -c {C[i]} -r 1 -g 1')
model = svm_train(p, sentence)
support_vector_coefs = [i[0] for i in model.get_sv_coef()]
sv = model.get_SV()
Wn = np.zeros(36)
for n in range(len(sv)):
    coef = support_vector_coefs[n]
    SV = []
    for k in range(36):
        if k+1 in sv[n]:
            SV.append(sv[n][k+1])
        else:
            SV.append(0)
    Wn += np.array(SV) * coef

print(len(support_vector_coefs))
p_label, p_acc, p_val = svm_predict(train_y, train_x, model)
test_y, test_x = svm_read_problem('satimage.scale.t')
for k in range(len(test_y)):
    if test_y[k] != 3:
        test_y[k] = -1
    else:
        test_y[k] = 1
p_labs, p_acc, p_vals = svm_predict(test_y, test_x, model)
```

```
[17.591375503037987, 46.4458723194913, 87.4856176980916, 102.58834218938105, 159.00656294579625]
```