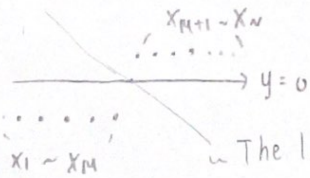
HTML HW5

1.



∴ The large-margin seperating hyperplane should be this line.

必通过$\left(\frac{x_M+x_{M+1}}{2}, 0\right)$

$\therefore h(x) = \left(x - \frac{x_M+x_{M+1}}{2}\right) = 0$

$w=1, \quad b = -\frac{x_M+x_{M+1}}{2}$ #

2. Choose (3): Based on Lecture 10, P.11

Choose (6): According to Lecture 10, p.26.

$\min \frac{1}{2} w^T w$ s.t. $y_n(w^T z_n + b) \geq 1$ for $n=1,2,\ldots,N$

with Lagrange multipliers, $\mathcal{L}(b,w,\alpha) = \frac{1}{2} w^T w + \sum_{n=1}^{N} \alpha_n (1 - y_n(w^T z_n + b))$

can be $\min(\max \mathcal{L}(b,w,\alpha))$

According to Lecture 10, p.29, 30.

$\frac{1}{2} w^T w = -\frac{1}{2} \|\sum_{n=1}^{N} \alpha_n y_n z_n\|^2 + \sum_{n=1}^{N} \alpha_n,$

$\Rightarrow \|w\|^2 = 2\sum_{n=1}^{N} \alpha_n - \|\sum_{n=1}^{N} \alpha_n y_n z_n\|^2$

$\Rightarrow \|w\|^{-1} =$ length of margin $= \left(2\sum_{n=1}^{N} \alpha_n - \|\sum_{n=1}^{N} \alpha_n y_n z_n\|^2\right)^{-\frac{1}{2}}$

∴ Choose (c) #

3. 因為只有4個 examples，所以直接將選項代入

(a) $y_1 = -1 \Rightarrow -(W^T \times X_1 - 1) = 1 \not\geq 4$ → 不成立

(b) $y_1 = -1 \Rightarrow -(W^T \times X_1 - 1) = 1 \not\geq 4$ → 不成立

(c) $y_1 = -1 \Rightarrow -(W^T \cdot X_1 - 4) = 4 \geq 4$ → v

$y_2 = -1 \Rightarrow -(W^T \cdot X_2 - 4) = 4 \geq 4$ → v

$y_3 = -1 \Rightarrow -(W^T \cdot X_3 - 4) = 4 \geq 4$ → v

$y_4 = +1 \Rightarrow (W^T \cdot X_4 - 4) = 1 \geq 1$ → v

(d) $y_1 = -1 \Rightarrow -(W^T X_1 - 4) = 4 \geq 4$ → v

$y_2 = -1 \Rightarrow -(W^T X_2 - 4) = 4 \geq 4$ → v

$y_3 = -1 \Rightarrow -(W^T X_3 - 4) = 4 \geq 4$ → v

$y_4 = 1 \Rightarrow (W^T X_4 - 4) = -\frac{19}{5} \not\geq 1$ → 不成立

∴ Choose (c) #

4. Find $\min\limits_{w,b} \frac{1}{2} W^T W$ s.t. $y_n(W^T x_n + b) \geq 1 \cdot [\![y_n = +1]\!] + 4 \cdot [\![y_n = -1]\!]$ for $n = 1, 2, \dots, N$.

Based on Lecture 10, p.26. Lagrange Function $\mathcal{L} = \frac{1}{2} W^T W + \sum_{n=1}^{N} (\alpha_n [(1 \cdot [\![y_n = +1]\!] + 4 \cdot [\![y_n = -1]\!]) - y_n(W^T x_n + b)])$

Find $\max\limits_{\text{all } \alpha_n \geq 0} \left(\underbrace{\min\limits_{b,w} \frac{1}{2} W^T W + \sum_{n=1}^{N} (\alpha_n [(1 \cdot [\![y_n = +1]\!] + 4 \cdot [\![y_n = -1]\!] - y_n(W^T x_n + b)])}_{} \right)$

$\frac{\partial \mathcal{L}}{\partial b} = 0 = -\sum_{n=1}^{N} \alpha_n y_n \Rightarrow b$ can be removed

$\Rightarrow \max \left(\underbrace{\min\limits_{b,w} \frac{1}{2} W^T W + \sum_{n=1}^{N} \alpha_n \cdot (1 \cdot [\![y_n = +1]\!] + 4 \cdot [\![y_n = -1]\!] - y_n W^T x_n)}_{} \right)$

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

$\Rightarrow \max \left(\frac{1}{2} W^T W + \sum_{n=1}^{N} \alpha_n \cdot (1 \cdot [\![y_n = +1]\!] + 4 [\![y_n = -1]\!]) - W^T w \right)$

$\Rightarrow \max \left(-\frac{1}{2} \| \sum_{n=1}^{N} \alpha_n y_n x_n \|^2 + \sum_{n=1}^{N} \alpha_n (1 \cdot [\![y_n = +1]\!] + 4 [\![y_n = -1]\!]) \right)$

$\Rightarrow \min\limits_{\alpha} \left(\frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m x_n^T x_m - \sum_{n=1}^{N} \alpha_n (1 \cdot [\![y_n = +1]\!] + 4 [\![y_n = -1]\!]) \right)$

Choose (C) #

HTML HW5

5. For SVM, we should find $\min_{b,w} \frac{1}{2}w^T w$

For hard-margin SVM, $w = \sum_{n=1}^{N} \alpha_n y_n x_n$ (Lecture 10, p.30)

For uneven-margin SVM, $w = \sum_{n=1}^{N} \alpha_n y_n x_n$ (Based on Q4)

So, the optimal α^* would not be changed.

Choose (a) #

6. $K(x, x') = (x^T x')^Q = (x_1 x_1' + x_2 x_2' + x_3 x_3' + \cdots + x_d x_d')^Q$

把 $(x_1 x_1' + x_2 x_2' + \cdots + x_d x_d')^Q$ 展开来後，每項的 x $(x_1, x_2, \ldots, x_d)$ 屬於 $\phi(x)$

每項的 x' $(x_1', x_2' \ldots, x_d')$ 屬於 $\phi(x')$

Ex: $d=2, Q=2$: $(x_1 x_1' + x_2 x_2')^2 = (x_1 x_1')^2 + 2(x_1 x_1' x_2 x_2') + (x_2 x_2')^2$

$\phi(x) = (x_1^2, \sqrt{2} x_1 x_2, x_2^2)$, $\phi(x') = (x_1'^2, \sqrt{2} x_1' x_2', x_2'^2)$

$\therefore$ Dimension $= (x_1 x_1' + x_2 x_2' + \cdots + x_d x_d')^Q$ 的項數

Unique term: $(x_1 x_1')^{n_1} (x_2 x_2')^{n_2} \cdots (x_d x_d')^{n_d}$ for $n_1 \sim n_d \geq 0$ and $\sum_{i=1}^{d} n_i = Q$

$\Rightarrow$ 重複組合, $H^d_Q = C^{d+Q-1}_Q$, choose (a) #

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

又 $K_2(x, x') = \phi(x)^T \phi(x')$

$\Rightarrow K_2(x, x) - 2K_2(x, x') + K_2(x', x')$

$= (1 + x^T x)^2 - 2(1 + x^T x')^2 + (1 + (x')^T x')^2$, since x & x' are unit vector, $x^T x = (x')^T x' = 1$

$\Rightarrow (1+1)^2 + (1+1)^2 - 2(1 + x^T x')^2 = 8 - 2\underbrace{(1 + x^T x')^2}_{\text{must } \geq 0}$

So the upper bound $= 8$, choose (d) #

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

$$w_{t+1} = w_t + y_{n(t)}\Phi(x_{n(t)}) = \sum_{n=1}^{\tilde{N}} \alpha_t[n(t)]\,\Phi(x_{n(t)}) + y_{n(t)}\Phi(x_{n(t)})$$

$$= \sum_{n=1}^{\tilde{N}} \underbrace{(\alpha_t[n(t)] + y_{n(t)})}_{\hookrightarrow \alpha_{t+1}[n(t)]} \cdot \Phi(x_{n(t)})$$

So, choose (c) #

9. Lagrange Function (Lecture 10, p.26)

$$\mathcal{L}(b, w, \alpha, \varepsilon) = \frac{1}{2}w^T w + \sum_{n=1}^{\tilde{N}} u_n \cdot \varepsilon_n + \sum_{n=1}^{\tilde{N}} \alpha_n (1 - \varepsilon_n - y_n(w^T\Phi(x_n) + b))$$

Find $\max\limits_{\text{all } \alpha_n \ge 0} (\min\limits_{b,w,\varepsilon} \mathcal{L}(b, w, \alpha, \varepsilon))$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 = -\sum_{n=1}^{\tilde{N}} \alpha_n y_n$$

$$\Rightarrow \mathcal{L} = \max(\min \frac{1}{2}w^T w + \sum_{n=1}^{\tilde{N}} u_n \varepsilon_n + \sum_{n=1}^{\tilde{N}} \alpha_n (1 - \varepsilon_n - y_n(w^T\Phi(x_n))))$$

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

$$\Rightarrow \mathcal{L} = \max(\min \frac{1}{2}w^T w + \sum_{n=1}^{\tilde{N}} u_n \varepsilon_n + \sum_{n=1}^{\tilde{N}} \alpha_n (1 - \varepsilon_n) - w^T w)$$

$$= \max(\min -\frac{1}{2}w^T w + \sum_{n=1}^{\tilde{N}} u_n \varepsilon_n + \sum_{n=1}^{\tilde{N}} \alpha_n (1 - \varepsilon_n))$$

$$\frac{\partial \mathcal{L}}{\partial \varepsilon_n} = 0 = u_n - \alpha_n \Rightarrow u_n = \alpha_n$$

$$\Rightarrow \mathcal{L} = \max(-\frac{1}{2}w^T w + \sum_{n=1}^{\tilde{N}} \alpha_n)$$

$$= \min \frac{1}{2}\sum_{n=1}^{\tilde{N}}\sum_{m=1}^{\tilde{N}} \alpha_n \alpha_m y_n y_m \Phi(x_n)^T \Phi(x_m) - \sum_{n=1}^{\tilde{N}} \alpha_n$$

$\therefore$ Choose (a) #

10. Within $(0,1)$, $E_{hinge}(\rho) = 1 - \rho$, $E_{smooth}(\rho) = \frac{1}{2}(1-\rho)^2$

Uniformly-average squared difference: $\int_0^1 [(1-\rho) - \frac{1}{2}(\underbrace{1-\rho)^2}_{\rho^2 - 2\rho + 1}]^2 d\rho = \int_0^1 (-\frac{1}{2}\rho^2 + \frac{1}{2})^2 d\rho$

$$= \int_0^1 (\frac{1}{4}\rho^4 - \frac{1}{2}\rho^2 + \frac{1}{4}) d\rho = \left[\frac{\rho^5}{20} - \frac{\rho^3}{6} + \frac{\rho}{4}\right]_0^1 = \frac{1}{20} - \frac{1}{6} + \frac{1}{4} = \frac{3 - 10 + 15}{60} = \frac{8}{60} = \frac{2}{15}$$

Choose (e) #

```python
#Q11-Q16
from libsvm.svmutil import *
y_train, x_train = svm_read_problem('satimage.scale.txt')
y_test, x_test = svm_read_problem('satimage.scale.t.txt')
```

```python
# Q11
y_train_q11 = []
for y_ in y_train:
    if y_ != 5.0:
        y_train_q11.append(-1)
    else:
        y_train_q11.append(1)
y_test_q11 = []
for y_ in y_test:
    if y_ != 5.0:
        y_test_q11.append(-1)
    else:
        y_test_q11.append(1)

x_train_q11 = []
x_test_q11 = []
for x in x_train:
    b = []
    for i in range(1,37):
        if i not in list(x.keys()):
            b.append(0)
        else:
            b.append(x[i])
    x_train_q11.append(b)
for x in x_test:
    b = []
    for i in range(1,37):
        if i not in list(x.keys()):
            b.append(0)
        else:
            b.append(x[i])
    x_test_q11.append(b)

import numpy as np
x_train_q11 = np.array(x_train_q11,  dtype=np.float)
y_train_q11 = np.array(y_train_q11)

from sklearn.svm import SVC
clf = SVC(C = 10, kernel = 'linear', gamma='auto')
clf.fit(x_train_q11,y_train_q11)
w_norm = np.linalg.norm(clf.coef_)
print('w = ',w_norm)
#choose a
```

```python
#Q12
def y_labels(positive_label):
    y_label_positive_label = []
    for y_ in y_train:
        if y_ != positive_label:
            y_label_positive_label.append(-1.0)
        else:
            y_label_positive_label.append(1.0)
    return y_label_positive_label

y_of_2 = y_labels(2.0)
y_of_3 = y_labels(3.0)
y_of_4 = y_labels(4.0)
y_of_5 = y_labels(5.0)
y_of_6 = y_labels(6.0)

model_2 = svm_train(y_of_2, x_train, '-t 1 -c 10 -d 3 -r 1 -g 1')
p_label, p_acc, p_val = svm_predict(y_of_2, x_train, model_2)
print('2: ', p_acc)
model_3 = svm_train(y_of_3, x_train, '-t 1 -c 10 -d 3 -r 1 -g 1')
p_label, p_acc, p_val = svm_predict(y_of_3, x_train, model_3)
print('3: ', p_acc)
model_4 = svm_train(y_of_4, x_train, '-t 1 -c 10 -d 3 -r 1 -g 1')
p_label, p_acc, p_val = svm_predict(y_of_4, x_train, model_4)
print('4: ', p_acc)
model_5 = svm_train(y_of_5, x_train, '-t 1 -c 10 -d 3 -r 1 -g 1')
p_label, p_acc, p_val = svm_predict(y_of_5, x_train, model_5)
print('5: ', p_acc)
model_6 = svm_train(y_of_6, x_train, '-t 1 -c 10 -d 3 -r 1 -g 1')
p_label, p_acc, p_val = svm_predict(y_of_6, x_train, model_6)
print('6: ', p_acc)
#choose c
```

```python
#Q13
support_vectors_2 = model_2.get_SV()
print('2:', len(support_vectors_2))
support_vectors_3 = model_3.get_SV()
print('3:', len(support_vectors_3))
support_vectors_4 = model_4.get_SV()
print('4:', len(support_vectors_4))
support_vectors_5 = model_5.get_SV()
print('5:', len(support_vectors_5))
support_vectors_6 = model_6.get_SV()
print('6:', len(support_vectors_6))
#choose e
```

```python
#Q14
y_of_1 = y_labels(1.0)

y_of_1_test = []
for y_ in y_test:
    if y_ != 1.0:
        y_of_1_test.append(-1.0)
    else:
        y_of_1_test.append(1.0)

model_a = svm_train(y_of_1, x_train, '-t 2 -c 0.01 -g 10')
p_label, p_acc, p_val = svm_predict(y_of_1_test, x_test, model_a)
print('a: ', p_acc)
model_b = svm_train(y_of_1, x_train, '-t 2 -c 0.1 -g 10')
p_label, p_acc, p_val = svm_predict(y_of_1_test, x_test, model_b)
print('b: ', p_acc)
model_c = svm_train(y_of_1, x_train, '-t 2 -c 1 -g 10')
p_label, p_acc, p_val = svm_predict(y_of_1_test, x_test, model_c)
print('c: ', p_acc)
model_d = svm_train(y_of_1, x_train, '-t 2 -c 10 -g 10')
p_label, p_acc, p_val = svm_predict(y_of_1_test, x_test, model_d)
print('d: ', p_acc)
model_e = svm_train(y_of_1, x_train, '-t 2 -c 100 -g 10')
p_label, p_acc, p_val = svm_predict(y_of_1_test, x_test, model_e)
print('e: ', p_acc)
# choose d
```

```python
#Q15
model_a = svm_train(y_of_1, x_train, '-t 2 -c 0.1 -g 0.1')
p_label, p_acc, p_val = svm_predict(y_of_1_test, x_test, model_a)
print('a: ', p_acc)
model_b = svm_train(y_of_1, x_train, '-t 2 -c 0.1 -g 1')
p_label, p_acc, p_val = svm_predict(y_of_1_test, x_test, model_b)
print('b: ', p_acc)
model_c = svm_train(y_of_1, x_train, '-t 2 -c 0.1 -g 10')
p_label, p_acc, p_val = svm_predict(y_of_1_test, x_test, model_c)
print('c: ', p_acc)
model_d = svm_train(y_of_1, x_train, '-t 2 -c 0.1 -g 100')
p_label, p_acc, p_val = svm_predict(y_of_1_test, x_test, model_d)
print('d: ', p_acc)
model_e = svm_train(y_of_1, x_train, '-t 2 -c 0.1 -g 1000')
p_label, p_acc, p_val = svm_predict(y_of_1_test, x_test, model_e)
print('e: ', p_acc)
# choose b
```

```python
#Q16
import random

def get_data(seed):
    x_train_new = []
    y_train_new = []
    x_val = []
    y_val = []
    random.seed(seed*11+2)
    randomlist = random.sample(range(0, len(x_train)), 200)
    for i in range(len(x_train)):
        if i in randomlist:
            x_val.append(x_train[i])
            y_val.append(y_of_1[i])
        else:
            x_train_new.append(x_train[i])
            y_train_new.append(y_of_1[i])
    return x_train_new, y_train_new, x_val, y_val


gammas = [0.1, 1, 10, 100, 100]
select_num = [0, 0, 0, 0, 0]
for t in range(1000):
    acc = []
    x_train_new, y_train_new, x_val, y_val = get_data(t)
    for gamma in gammas:
        model = svm_train(y_train_new, x_train_new, f'-t 2 -c 0.1 -g {gamma}')
        p_label, p_acc, p_val = svm_predict(y_val, x_val, model)
        acc.append(p_acc[0])
    select_num[np.argmax(acc)] += 1
select_num

#choose a
```