HTML HW4

1. $w(t+1) \leftarrow w(t) - \eta \nabla (E_{in}(w(t)) + \frac{\lambda}{N} (w(t))^T \cdot w(t)$

$\Rightarrow w(t+1) \leftarrow w(t) - \eta \nabla E_{in}(w(t)) - \eta \frac{2\lambda}{N} w(t)$

$\Rightarrow w(t+1) \leftarrow \left(1 - \frac{2\eta\lambda}{N}\right) \cdot w(t) - \eta \nabla E_{in}(w(t))$ #

2. To find $\min_{w \in \mathbb{R}} \frac{1}{N} \sum_{n=1}^{N} (w-y_n)^2 + \frac{\lambda}{N} w^2 \Rightarrow$ find w^* satisfy $\nabla_w \left(\frac{1}{N} \sum_{n=1}^{N} (w-y_n)^2 + \frac{\lambda}{N} w^2 \right) = 0$

$\nabla_w \left(\frac{1}{N} \sum_{n=1}^{N} (w-y_n)^2 + \frac{\lambda}{N} w^2 \right) = \frac{1}{N} \sum_{n=1}^{N} (2w - 2y_n) + \frac{2\lambda}{N} w = 0$

$\Rightarrow \sum_{n=1}^{N} \cancel{2}w + \cancel{2}\lambda w = \sum_{n=1}^{N} \cancel{2} y_n \quad \Rightarrow w(N+\lambda) = \sum_{n=1}^{N} y_n \quad \Rightarrow w^* = \frac{\sum_{n=1}^{N} y_n}{N+\lambda}$

$C = (w^*)^2 = \left[\frac{\sum_{n=1}^{N} y_n}{N+\lambda} \right]^2$

3. $w = V \cdot \tilde{w} \quad \Rightarrow \tilde{w} = V^{-1} w$

1t λ $\min \frac{1}{N} \sum_{n=1}^{N} (\tilde{w}^T \phi(x_n) - y_n)^2 + \frac{\lambda}{N} (\tilde{w}^T \tilde{w})$

$= \min \frac{1}{N} \sum_{n=1}^{N} \left[(V^{-1} w)^T \cdot V \cdot x_n - y_n \right]^2 + \frac{\lambda}{N} (V^{-1} w)^T (V^{-1} w)$

$= \min \frac{1}{N} \sum_{n=1}^{N} \left[w^T (V^{-1})^T \cdot V \cdot x_n - y_n \right]^2 + \frac{\lambda}{N} w^T (V^{-1})^T V^{-1} w$

$= \min \frac{1}{N} \sum_{n=1}^{N} (w^T x_n - y_n)^2 + \frac{\lambda}{N} w^T (V^{-1})^2 \cdot w$

$\Rightarrow V = (V^{-1})^2$ #

4. $\min \mathbb{E} \left[\frac{1}{N} \sum_{n=1}^{N} (w^T \cdot (x_n + \epsilon) - y_n)^2 \right] = \min \mathbb{E} \left[\frac{1}{N} \sum_{n=1}^{N} (\underline{w^T x_n} + w^T \epsilon - \underline{y_n})^2 \right]$

$= \min \mathbb{E} \left[\frac{1}{N} \sum_{n=1}^{N} \left[(w^T x_n - y_n)^2 + 2 \cdot (w^T x_n - y_n) \cdot w^T \epsilon + (w^T \epsilon)^2 \right] \right]$

$= \min \frac{1}{N} \sum_{n=1}^{N} (w^T x_n - y_n)^2 + \mathbb{E} \left[\frac{2}{N} \cdot \sum_{n=1}^{N} (w^T x_n - y_n) \cdot w^T \epsilon + \frac{1}{N} \sum_{n=1}^{N} (w^T \epsilon)^2 \right]$

$\mathbb{E}(\epsilon) = \mu = 0, \quad \mathbb{E}(\epsilon^2) = Var(\epsilon) + [E(x)]^2 = \sigma^2$

$= \min \frac{1}{N} \sum_{n=1}^{N} (w^T x_n - y_n)^2 + 0 + \sigma^2 \cdot \|w\|^2$

$\Rightarrow \frac{\lambda}{N} = \sigma^2 \quad \Rightarrow \lambda = N\sigma^2$ #

5. To find $\min_{y \in R} \frac{1}{N} \sum_{n=1}^{N} (y-y_n)^2 + \frac{\alpha k}{N} \Omega(y) \Rightarrow \nabla_y \left(\frac{1}{N} \sum_{n=1}^{N} (y-y_n)^2 + \frac{\alpha k}{N} \Omega(y) \right) = 0$

$$\Rightarrow \frac{1}{N} \left(\sum_{n=1}^{N} 2y - 2y_n \right) + \frac{\alpha k}{N} \Omega'(y) = 0$$

Let $\Omega(y) = (y+a)^2$

$$\Rightarrow 2y - \frac{2}{N} \sum_{n=1}^{N} y_n + \frac{\alpha k}{N} \cdot 2(y+a) = 0$$

$$\Rightarrow \left(1 + \frac{\alpha k}{N}\right) \cdot y - \frac{1}{N} \sum_{n=1}^{N} y_n + \frac{\alpha k \cdot a}{N} = 0$$

$$\Rightarrow \left(\frac{N+\alpha k}{N} \right) \cdot \left(\frac{\sum_{n=1}^{N} y_n + \alpha}{N+\alpha k} \right) - \frac{1}{N} \sum_{n=1}^{N} y_n + \frac{\alpha k \cdot a}{N} = 0$$

$$\Rightarrow \cancel{\frac{1}{N} \sum_{n=1}^{N} y_n} + \frac{\alpha}{N} - \frac{1}{N} \sum_{n=1}^{N} y_n + \frac{\alpha k \cdot a}{N} = 0$$

$$\Rightarrow \frac{\alpha}{N} = - \frac{\alpha k a}{N} \Rightarrow a = -\frac{1}{k} \Rightarrow \Omega(y) = \left(y - \frac{1}{k}\right)^2 \#$$

6. $\nabla_w \tilde{E}_{aug}(w) = \cancel{\nabla E_{in}(w^*)} + \frac{1}{2} H \cdot 2(w-w^*) + \frac{2\lambda}{N} w$

To minimize $\Rightarrow \nabla_w \tilde{E}_{aug}(w) = 0$

$$\Rightarrow \frac{1}{2} H \cdot 2(w-w^*) + \frac{2\lambda}{N} w = 0$$

$$\Rightarrow \left(H + \frac{2\lambda}{N}\right) w = H w^* \Rightarrow w = \left(H + \frac{2\lambda}{N} I\right)^{-1} H w^* \#$$

7. If we choose positive one as validation, the label of validation set is positive, the Aminority is positive, the error $= \frac{1}{N} \sum_{i=1}^{N} 0 = 0$

If we choose negtive one as validation, the label of validation set is negative, the Aminority is negative, the error $= \frac{1}{N} \sum_{i=1}^{N} 0 = 0$

The E_{loocv} (Aminority) $= \frac{N}{2N} \times 0 + \frac{N}{2N} \times 0 = 0 \#$

HTML HW 4

8. $(x_1, y_1) = (2, 0)$ as validation. $(x_2, y_2) = (\rho, 2)$, $(x_3, y_3) = (-2, 0)$ as training

To minimize $\frac{1}{2}[(w_0 - y_2)^2 + (w_0 - y_3)^2] \Rightarrow (w_0 - y_2) + (w_0 - y_3) = 0 \Rightarrow w_0 = \frac{1}{2}(y_2 + y_3)$

$$E_{loocv}(\text{constant}) = \frac{1}{3}\Big[\underbrace{\left(\tfrac{1}{2}(2+0) - 0\right)^2}_{(x_1, y_1)\text{ as val}} + \underbrace{\left(\tfrac{1}{2}(0+0) - 2\right)^2}_{(x_2, y_2)\text{ as val}} + \underbrace{\left(\tfrac{1}{2}(0+2) - 0\right)^2}_{(x_3, y_3)\text{ as val}}\Big]$$

$$= \frac{1}{3}(1 + 4 + 1) = 2$$

For linear, $\begin{cases} y_2 = w_0 + w_1 x_2 \\ y_3 = w_0 + w_1 x_3 \end{cases} \Rightarrow w_1 = \dfrac{y_3 - y_2}{x_3 - x_2}, \quad w_0 = \dfrac{y_2 x_3 - y_3 x_2}{x_3 - x_2}$

$$E_{loocv}(\text{linear}) = \frac{1}{3}\Big[\underbrace{\left[\frac{-2}{-2-\rho} \cdot 2 + \frac{-4-0}{-2-\rho} - 0\right]^2}_{(x_1, y_1)\text{ as val}} + \underbrace{\left[\frac{0-0}{-2-2} \cdot \rho + \frac{0-0}{-2-2} - 2\right]^2}_{(x_2, y_2)\text{ as val}} + \underbrace{\left[\frac{2-0}{\rho-2} \cdot (-2) + \frac{0-4}{\rho-2} - 0\right]^2}_{(x_3, y_3)\text{ as val}}\Big]$$

$$= \frac{1}{3}\left[\left(\frac{-8}{-2-\rho}\right)^2 + 4 + \left(\frac{-8}{\rho-2}\right)^2\right] = \frac{1}{3}\left[\left(\frac{64}{(\rho+2)^2} + 4 + \frac{64}{(\rho-2)^2}\right)\right]$$

$$= \frac{1}{3}\left[\frac{64[(\rho-2)^2 + (\rho+2)^2]}{(\rho+2)^2 \cdot (\rho-2)^2} + 4\right]$$

$E_{loocv}(\text{constant}) = E_{loocv}(\text{linear})$: $2 = \frac{1}{3}\left[\frac{64(2\rho^2 + 8)}{(\rho+2)^2 \cdot (\rho-2)^2} + 4\right] \Rightarrow 2 = \frac{64 \cdot 2 \cdot (\rho^2 + 4)}{(\rho+2)^2 \cdot (\rho-2)^2}$

$\Rightarrow (\rho+2)^2 \cdot (\rho-2)^2 = 64(\rho^2 + 4)$

For (a): left = 2960, right = 1463.0625

(b): left = 3856, right = 2730.0625

(c): left = 4880, right = 4658.0625

(d): left = 6032, right = 7439.0625

(e): left = 7312, right = 11289.0625, choose C #

9. $\mathbb{E}\left(\frac{1}{K}\sum_{n=N-K+1}^{N}(y_n-\bar{y})^2\right) = \mathbb{E}\left(\frac{1}{K}\sum_{n=N-K+1}^{N}(y_n^2-2y_n\bar{y}+\bar{y}^2)\right)$

$$= \mathbb{E}\left(\frac{1}{K}\sum_{n=N-K+1}^{N} y_n^2\right) - 2\,\mathbb{E}\left(\frac{1}{K}\sum_{n=N-K+1}^{N} y_n\bar{y}\right) + \mathbb{E}\left(\frac{1}{K}\cdot K\cdot \bar{y}^2\right)$$

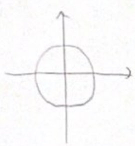
$$= \sigma^2 - 2\cdot\left[\ \mathbb{E}(\bar{y})\cdot \underbrace{\mathbb{E}\left(\frac{1}{K}\sum_{n=N-K+1}^{N} y_n\right)}_{=0}\right] + \mathbb{E}(\bar{y}^2)$$

$$\mathbb{E}(\bar{y}^2) = Var(\bar{y}) + (\mathbb{E}(\bar{y}))^2 = Var(\bar{y})$$

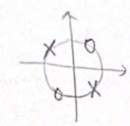
$$Var(\bar{y}) = Var\left(\frac{1}{N-K}\right)\sum_{n=1}^{N-K} y_n = \left(\frac{1}{N-K}\right)^2 \sum_{n=1}^{N-K} Var(y_n) = \left(\frac{1}{N-K}\right)\sigma^2$$

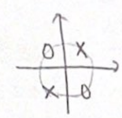
$$= \sigma^2 - 0 + \left(\frac{1}{N-K}\right)\sigma^2$$ #

10. This problem can be viewed as find "How many kinds of lines for four points?" in handout-04 page 9.



There are 2 situations that cannot be linear seperable.

(

, 

) $\Rightarrow E_{in} = \frac{1}{4}$ for each situation.

So, the expectation is $\frac{1}{16} \times \frac{1}{4} \times 2 = \frac{1}{32}$ #

11. $E_{out}(g) = P(y=+1)\cdot P(g(x)=-1 \mid y=+1) + P(y=-1)\cdot P(g(x)=+1 \mid y=-1)$

$$= p\cdot\epsilon_+ + (1-p)\cdot\epsilon_-$$

g be as good as the constant classifier g_- in terms of E_{out}

$\Rightarrow E_{out}(g) = p\cdot\epsilon_+ + (1-p)\cdot\epsilon_- = E_{out}(g_-) = p$

$\Rightarrow p(\epsilon_+ - \epsilon_-) + \epsilon_- = p$

$\Rightarrow \epsilon_- = p(1-\epsilon_+ + \epsilon_-)$

$\Rightarrow p = \dfrac{\epsilon_-}{1-\epsilon_+ + \epsilon_-}$ #

```python
#data pre-processing for Q12-Q16

from liblinear.liblinearutil import *
f = open('hw4_train.dat.txt', 'r')
train_data = f.read().splitlines()
f.close()
f = open('hw4_test.dat.txt', 'r')
test_data = f.read().splitlines()
f.close()

train_x = []
train_y = []
for line in train_data:
    line = line.split(' ')
    x = []
    for i in range(6):
        x.append(float(line[i]))
    train_x.append(x)
    train_y.append(float(line[6]))

test_x = []
test_y = []
for line in test_data:
    line = line.split(' ')
    x = []
    for i in range(6):
        x.append(float(line[i]))
    test_x.append(x)
    test_y.append(float(line[6]))

import numpy as np
from sklearn.preprocessing import PolynomialFeatures
poly = PolynomialFeatures(3)
train_x_trans = poly.fit_transform(np.array(train_x))
test_x_trans = poly.fit_transform(np.array(test_x))
lamb = [10**(-4), 10**(-2), 10**(0), 10**(2), 10**(4)]
C = [1/(2*l) for l in lamb]
```

```python
#Q12
for c in C:
    m = train(train_y, train_x_trans, (f'-s 0 -c {str(c)} -e 0.000001'))
    p_label, p_acc, p_val = predict(test_y, test_x_trans, m)

#Q13
for c in C:
    m = train(train_y, train_x_trans, (f'-s 0 -c {str(c)} -e 0.000001'))
    p_label, p_acc, p_val = predict(train_y, train_x_trans, m)

#Q14
for c in C:
    m = train(train_y[:120], train_x_trans[:120], (f'-s 0 -c {str(c)} -e 0.000001'))
    p_label, p_acc, p_val = predict(train_y[120:], train_x_trans[120:], m)

m = train(train_y[:120], train_x_trans[:120], (f'-s 0 -c {C[3]} -e 0.000001'))
p_label, p_acc, p_val = predict(test_y, test_x_trans, m)

#Q15
m = train(train_y, train_x_trans, (f'-s 0 -c {C[3]} -e 0.000001'))
p_label, p_acc, p_val = predict(test_y, test_x_trans, m)

#Q16
train_x_cv = [train_x_trans[:40], train_x_trans[40:80], train_x_trans[80:120], train_x_trans[120:160],
train_x_trans[160:]]
train_y_cv = [train_y[:40], train_y[40:80], train_y[80:120], train_y[120:160], train_y[160:]]
for c in C:
    E = 0
    for i in range(5):
        train_y_tmp = []
        train_x_tmp = []
        val_y_tmp = train_y_cv[i]
        val_x_tmp = train_x_cv[i]
        for j in range(5):
            if j!=i:
                train_y_tmp.extend(train_y_cv[j])
                train_x_tmp.extend(train_x_cv[j])

        m = train(train_y_tmp, train_x_tmp, (f'-s 0 -c {c} -e 0.000001'))
        p_label, p_acc, p_val = predict(val_y_tmp, val_x_tmp, m)
        E += (1-p_acc[0]/100)
    print(C.index(c), E/5)
```