1. $E_{aug} = E_{in} + \frac{\lambda}{N} w^T w$

$\Rightarrow \nabla E_{aug} = \nabla E_{in} + \frac{\lambda}{N} w$

$\Rightarrow W_{t+1} = W_t - \eta \left(\nabla E_{in} + \frac{2\lambda}{N} W_t \right)$

$= W_t - \eta \nabla E_{in} - \eta \frac{2\lambda}{N} W_t$

$= \left(1 - \eta \frac{2\lambda}{N}\right) W_t - \eta \nabla E_{in}(W_t)$

ans : c

2. $$\frac{d}{dw} \min_{w \in R} \frac{1}{N} \sum_{n=1}^{N} (w-y_n)^2 + \frac{\lambda}{N} w^2$$

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

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

$$\Rightarrow \sum_{n=1}^{N} (w-y_n) + \lambda w = 0$$

$$\Rightarrow \sum_{n=1}^{N} w - \sum_{n=1}^{N} y_n + \lambda w = 0$$

$$\Rightarrow N w + \lambda w - \sum_{n=1}^{N} y_n = 0$$

$$\Rightarrow (N+\lambda) w - \sum_{n=1}^{N} y_n = 0$$

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

ans: b

3. 令 $\tilde{w}^T V X = w^T X$, $\tilde{w} = (V^{-1})^T \cdot w$

$$\tilde{w}^T = w^T V^{-1} \Rightarrow \tilde{w}^T \tilde{w} = w^T V^{-1} V^{-1} w$$

$$= w^T (V^{-1})^2 w$$

$$= w^T U w$$

$$\Rightarrow \tilde{w}^T \tilde{w} = w^T (V^{-1})^2 w = w^T U w$$

$$\min_{\tilde{w} \in R^{d+1}} \frac{1}{N} \sum_{n=1}^{N} (\tilde{w}^T \Phi(x_n) - y_n)^2 + \frac{\lambda}{N} (\tilde{w}^T \tilde{w})$$

$$\Rightarrow \min_{w \in R^{d+1}} \frac{1}{N} \sum_{n=1}^{N} (w^T x_n - y_n)^2 + \frac{\lambda}{N} (w^T U w)$$

$$U = (V^{-1})^2$$

$\Rightarrow$ ans: d

4.

$$\min_{w \in \mathbb{R}^{d+1}} E\left[\frac{1}{N}\sum_{n=1}^{N}\left(w^T \Phi(x_n) - y_n\right)^2\right]$$

$$= \min_{w \in \mathbb{R}^{d+1}} \frac{1}{N}\sum_{n=1}^{N} E\left[\left(w^T x_n + w^T \varepsilon_n - y_n\right)^2\right],\ \text{令}\ w^T x_n - y_n = \Delta$$

$$\Rightarrow \min_{w \in \mathbb{R}^{d+1}} \frac{1}{N}\sum_{n=1}^{N} E\left[\Delta^2 + 2\Delta w^T \varepsilon_n + (w^T \varepsilon_n)^2\right]$$

$$\Rightarrow \min_{w \in \mathbb{R}^{d+1}} \frac{1}{N}\sum_{n=1}^{N} E\left[\Delta^2 + 2\|w\|_2^2 x_n^T \varepsilon_n - 2 y_n w^T \varepsilon_n + \|w\|_2^2 \varepsilon_n^2\right]$$

$$= \min_{w \in \mathbb{R}^{d+1}} \frac{1}{N}\sum_{n=1}^{N} E(\Delta^2) + \cancelto{0}{E(2\|w\|_2^2 x_n^T \varepsilon_n)} - \cancelto{0}{E(2 y_n w^T \varepsilon_n)} + \cancelto{\|w\|_2^2 \sigma^2}{E(\|w\|_2^2 \varepsilon_n^2)}$$

$$= \min_{w \in \mathbb{R}^{d+1}} \frac{1}{N}\sum_{n=1}^{N} \Delta^2 + \|w\|_2^2 \sigma^2$$

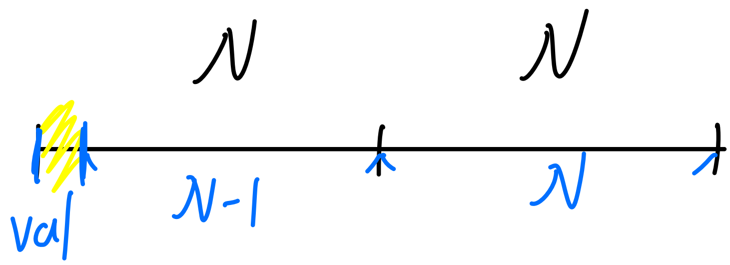
$$= \min_{w \in \mathbb{R}^{d+1}} \frac{1}{N}\sum_{n=1}^{N} (w^T x_n - y_n)^2 + \|w\|_2^2 \sigma^2$$

$$\sigma^2 = \frac{\lambda}{N}$$

$$\lambda = N \cdot \sigma^2$$

ans : a

7.



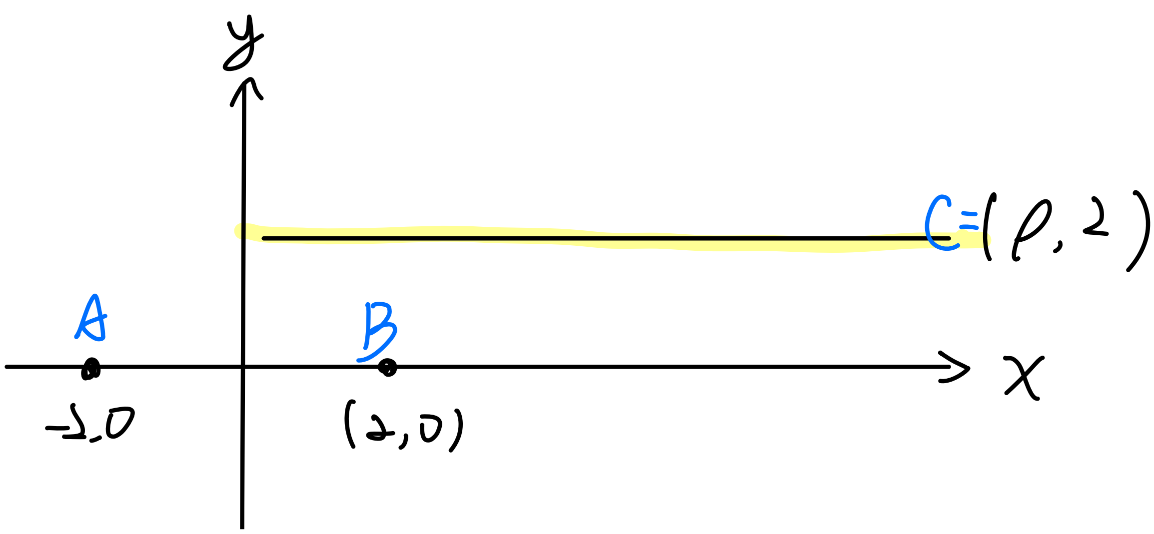
$$E_{val}^{(n)}(g_n^-) = err(g_n^-(x_n), y_n) = e_n = 1$$

(A 在 $E_{val}^{(n)}$ 永遠会 predict 錯，故 $e_n = 1$)

$$E_{loocv}(\mathcal{H}, \mathcal{A}) = \frac{1}{2N}\sum_{n=1}^{2N} e_n = \frac{1}{2N} 2N = 1$$

ans : e

8.



① constant: $err(c) = \frac{1}{3}\left(2^2 + 1^2 + 1^2\right)$

② Linear: $err(l) = \frac{1}{3}\left((2-0)^2 + \left(\frac{2}{\rho+2}\cdot 4 - 0\right)^2 + \left(\frac{2}{\rho-2}(-4) - 0\right)^2\right)$

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

val C: $y = 0$

val B: $y - 0 = m(x+2)$, $m = \frac{2}{\rho+2}$

$y = \frac{2}{\rho+2}(x+2)$, $y(2) = \frac{2}{\rho+2}(4)$

val A: $y - 0 = m(x-2)$, $m = \frac{2}{\rho-2}$

$y = \frac{2}{\rho-2}(x-2)$, $y(-2) = \frac{2}{\rho-2}(-4)$

by ①.②

$$\Rightarrow \left(\frac{8}{\rho+2}\right)^2 + \left(\left(\frac{-8}{\rho-2}\right)^2 + 4\right) = 6$$

$$\Rightarrow \rho = \pm 2\sqrt{9+4\sqrt{6}}, \pm 2i\sqrt{-9+4\sqrt{6}}$$

$$\Rightarrow \rho = 2\sqrt{9+4\sqrt{6}} \simeq 8.5$$, ans : C

9.

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

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

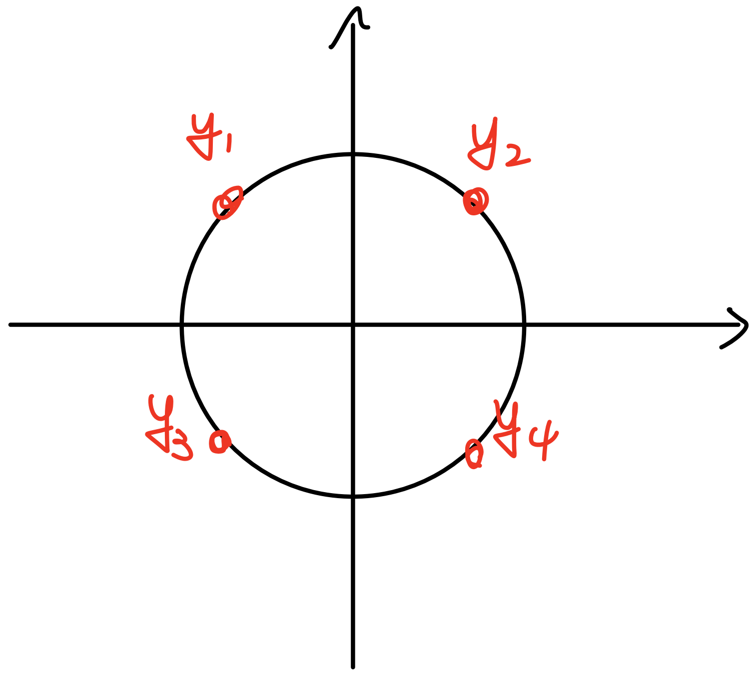
由於 y_n 來自 iid 且 mean $=0$.

故 $\bar{y}=0$

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

$$=E(\sigma^2)=\sigma^2$$, ans : a

10.



只有 $y_1, y_2, y_3, y_4 = \{(0,x,x,0),(x,0,0,x)\}$

時才會無法 Linear seperable. 此時 err 為 $\frac{1}{4}$

已知共16種組合，其中2組無法 Linear seperable

$$\Rightarrow \underbrace{\left(\frac{0}{4}+\frac{0}{4}+\cdots+\frac{0}{4}+\frac{1}{4}+\frac{1}{4}\right)}_{共16個} \times \frac{1}{16} = \frac{2}{4}\times\frac{1}{16} = \frac{1}{32}$$

ans: a

11. $P(y=+1)=P$

$P(y=-1)=1-P$

故 $E_{out}(g)=P\varepsilon_+ + (1-P)\varepsilon_-$

已知 $E_{out}(g_-)=P$

當 P 為何，使得 $E_{out}(g)=P$?

令 $P\varepsilon_+ + (1-P)\varepsilon_- = P$

$\Rightarrow P(\varepsilon_+ - \varepsilon_-) + \varepsilon_- = P$

$\Rightarrow \varepsilon_- = P(1-\varepsilon_+ + \varepsilon_-)$

$\Rightarrow P = \dfrac{\varepsilon_-}{1-\varepsilon_+ + \varepsilon_-}$ ans: b

```python
import numpy as np
from sklearn.preprocessing import PolynomialFeatures
from liblinear.liblinearutil import *



def third_order_polytrans(datapath):

    data = np.loadtxt(datapath)
    x,y = data[:,:-1],data[:,-1]
    trans = PolynomialFeatures(degree=3)
    x = trans.fit_transform(x)
    y = list(map(int,y))
    data_bag = []
    for data in x:
        dict_ = {}
        for key,val in enumerate(data,1):
            dict_[key] = val
        data_bag.append(dict_)
    return y,data_bag


def Q12(lambd):

    acc_bag = []
    for i in range(5):
        model = train(test_y,test_data,f"-s 0 -c {1/(2*lambd[i])} -e 0.000001 -q")
        p_label, p_acc, p_test = predict(test_y, test_data, model)
        acc_bag.append(p_acc[0])
    return acc_bag

def Q13(lambd):
    acc_bag = []
    for i in range(5):
        model = train(tra_y,tra_data,f"-s 0 -c {1/(2*lambd[i])} -e 0.000001 -q")
        p_label, p_acc, p_test = predict(tra_y, tra_data, model)
        acc_bag.append(p_acc[0])
    return acc_bag

def Q14(lambd,tra_data,tra_y,test_data,test_y):
    acc_bag = []
    tra_d,val_d = tra_data[:120],tra_data[120:]
    tra_y ,val_y = tra_y[:120],tra_y[120:]
    for i in range(5):
        model = train(tra_y,tra_d,f"-s 0 -c {1/(2*lambd[i])} -e 0.000001 -q")
        p_label, p_acc, p_test = predict(val_y, val_d, model)
        acc_bag.append(p_acc[0])

    # find best and last index lambda
    best_lambda = 0
    best_acc = acc_bag[0]
    for i,_ in enumerate(acc_bag):
        if _ >= best_acc:
            best_acc = _
            best_lambda  = i
    print("==================================testing=====================================")
    model = train(tra_y, tra_d, f"-s 0 -c {1 / (2 * lambd[best_lambda])} -e 0.000001 -q")
    p_label, p_acc, p_test = predict(test_y, test_data, model)
    return best_lambda,p_acc[0]

def Q15(lambd,tra_data,tra_y,test_data,test_y):
    # Q14的best_lambda 是 lambd[3]
    model = train(tra_y,tra_data,f"-s 0 -c {1/(2*lambd[3])} -e 0.000001 -q")
    p_label, p_acc, p_test = predict(test_y, test_data, model)
    return p_acc[0]

def Q16(lambd,tra_data,tra_y):
    split_data = [tra_data[:40],tra_data[40:80],tra_data[80:120],tra_data[120:160],tra_data[160:]]
    split_y = [tra_y[:40],tra_y[40:80],tra_y[80:120],tra_y[120:160],tra_y[160:]]
    average_acc_bag = []

    for l in range(5):
        acc = 0
        for i in range(5):
            val_data = split_data[i]
            val_y = split_y[i]
            tra_d = []
            tra_y = []
            for j in range(5):
                if j == i:pass
                else:
                    tra_d+=split_data[j]
                    tra_y+=split_y[j]

            model = train(tra_y, tra_d, f"-s 0 -c {1 / (2 * lambd[l])} -e 0.000001 -q")
            p_label, p_acc, p_test = predict(val_y, val_data , model)
            acc+=p_acc[0]
        average_acc_bag.append(acc/5)
    return average_acc_bag

if __name__ == '__main__':

    tra_datapath = "./hw4_train.dat.txt"
    tra_y,tra_data = third_order_polytrans(tra_datapath)
    test_datapath = "./hw4_test.dat.txt"
    test_y,test_data = third_order_polytrans(test_datapath)

    log_lambd = [-4,-2,0,2,4]
    lambd = [10**i for i in log_lambd]

    Q12_acc = Q12(lambd)
    Q13_acc = Q13(lambd)
    Q14_lambda,Q14_acc = Q14(lambd,tra_data,tra_y,test_data,test_y)
    Q15_acc = Q15(lambd,tra_data,tra_y,test_data,test_y)
    Q16_acc = Q16(lambd,tra_data,tra_y)
```