1. by lecture 12 - 7/43.

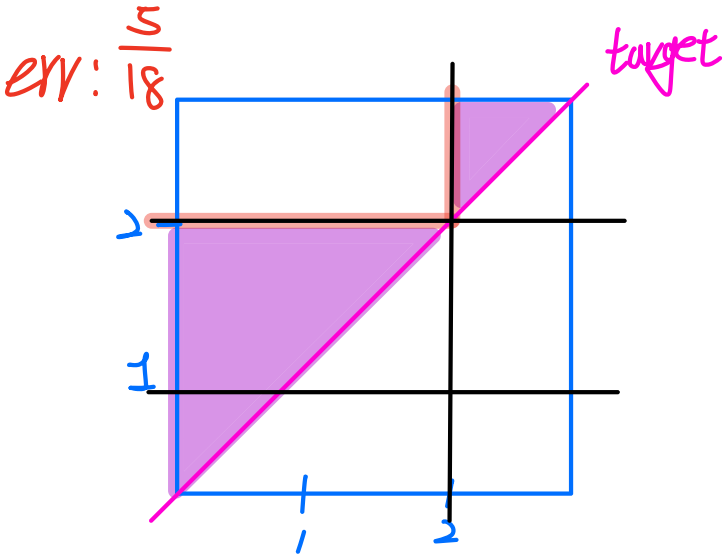
$$avg(E_{out}(g_t)) = avg(\mathcal{E}(g_t - G_1)^2) + E_{out}(G_1)$$

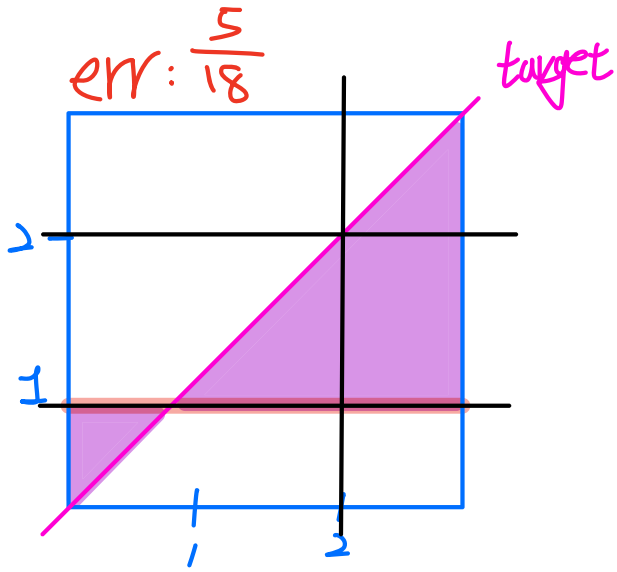
當 $avg(\mathcal{E}(g_t - G_1)^2) = 0$ 時, $E_{out}(G_1) = avg(E_{out}(g_t))$

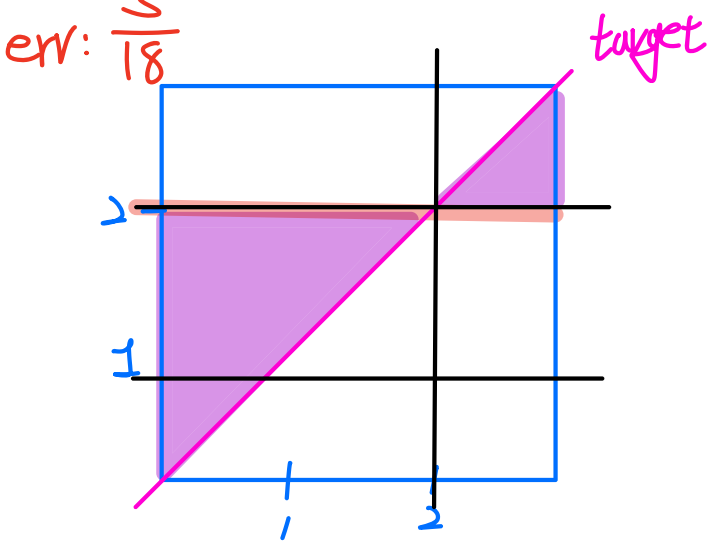
故 upper bound 為 $avg(E_{out}(g_t)) = \frac{1}{T}\sum_{t=1}^{T} e_t$ ans: d

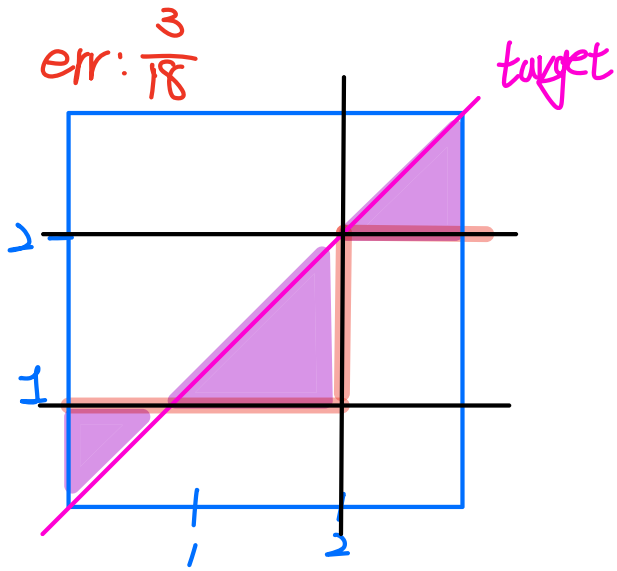
2.

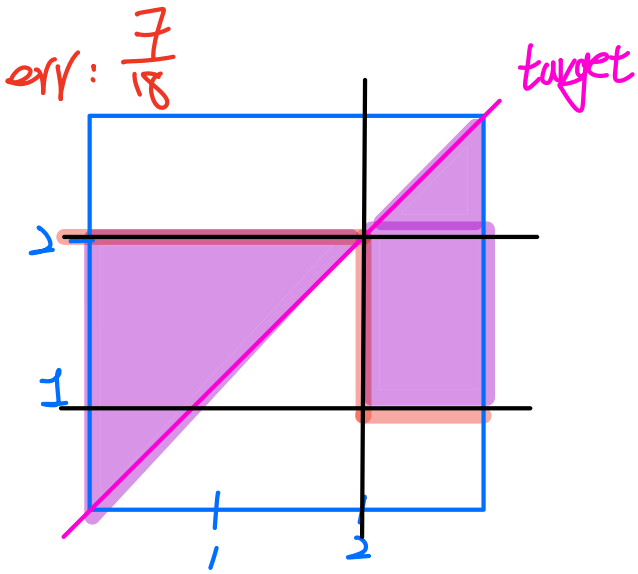
$g_1 \sim g_5$ 的所有可能如下：

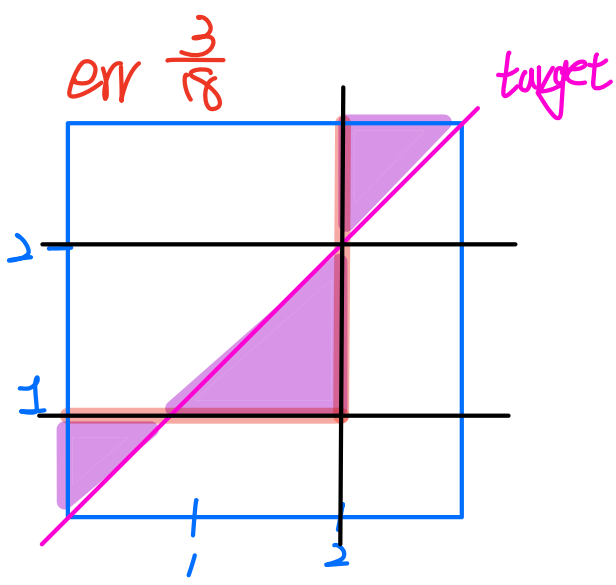




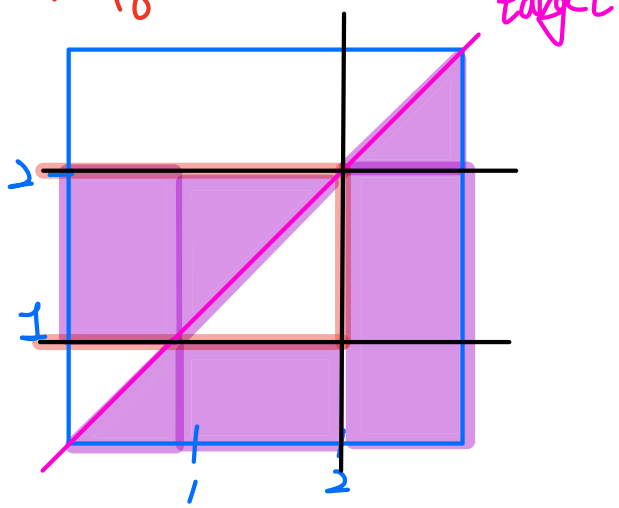




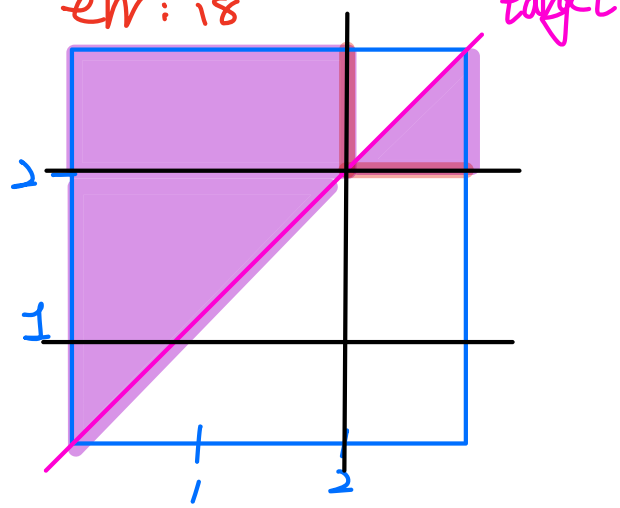




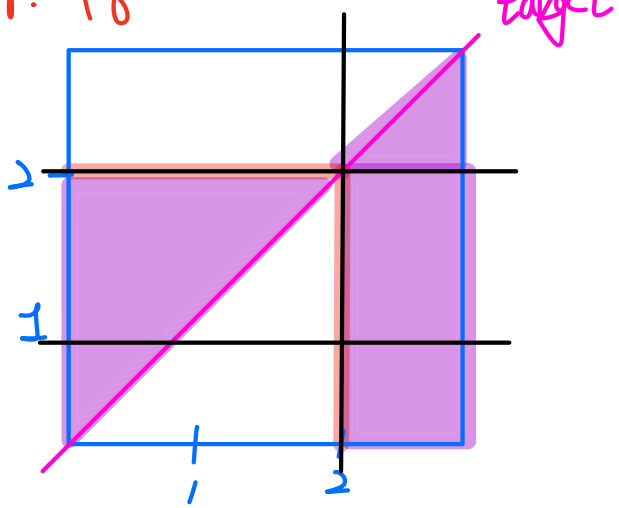
err: $\frac{11}{18}$



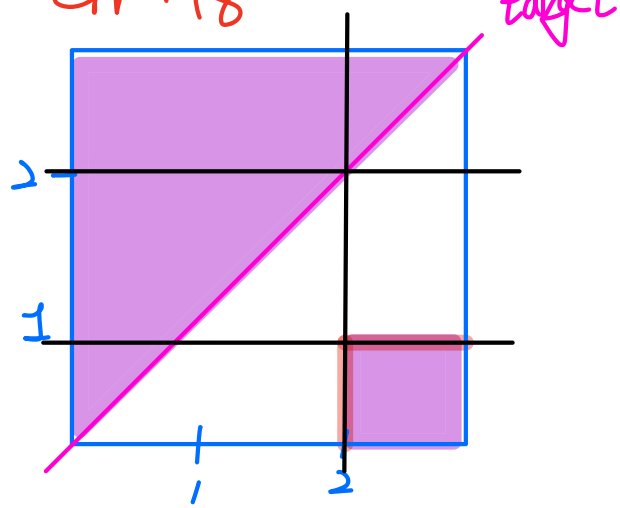
err: $\frac{9}{18}$



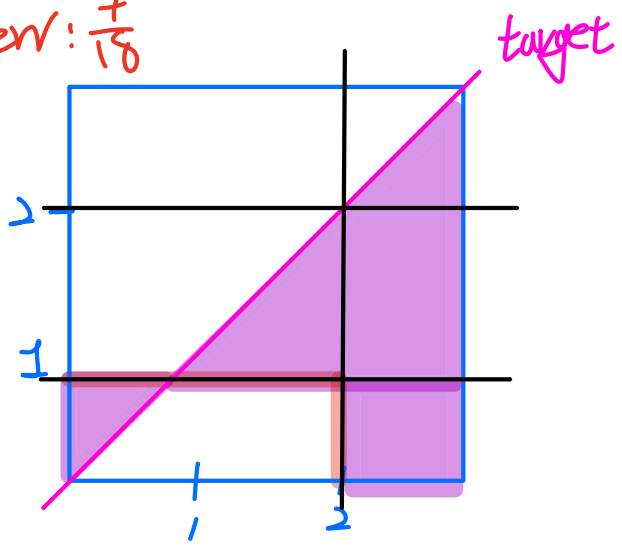
err: $\frac{9}{18}$



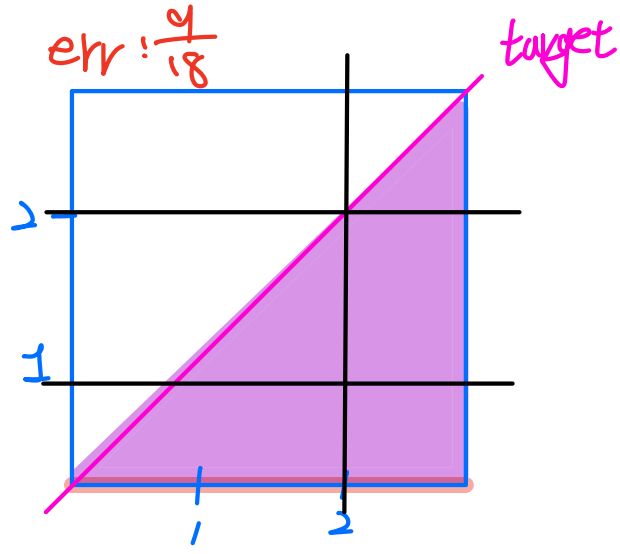
err: $\frac{11}{18}$



err: $\frac{7}{18}$



err: $\frac{9}{18}$



ans: $\frac{3}{18}$

3. (1) $\vec{g}(X) = g_{+1,-1,2L+1},\ g_{+1,1,2L+3} \cdots g_{+1,1,2R-1}$

$g_{+1,-2,2L+1},\ g_{+1,2,2L+3} \cdots g_{+1,2,2R-1}$

$\vdots \qquad \vdots \qquad \vdots$

$g_{+1,-d,2L+1},\ g_{+1,d,2L+3} \cdots g_{+1,d,2R-1}$

$g_{-1,-1,2L+1},\ g_{1,1,2L+3} \cdots g_{-1,1,2R-1}$

$g_{-1,-2,2L+1},\ g_{1,2,2L+3} \cdots g_{-1,2,2R-1}$

$\vdots \qquad \vdots \qquad \vdots$

$g_{-1,-d,2L+1},\ g_{1,d,2L+3} \cdots g_{-1,d,2R-1}$

共 $d\left[\frac{(2R-1)-(2L+1)}{2}+1\right] \times 2$ 個項

$\Rightarrow d \times (R-L) \times 2 = 2d(R-L)$ —①

(2) 當 $\theta \Rightarrow x_i' < \theta < x_i$ 時, $g_{si\theta}(X) \times g_{si\theta}(X')$

為 $s \cdot sign(x_i - \theta) \times s \cdot sign(x_i' - \theta) \Rightarrow -1$ —②

(3) 由 $\frac{(2R-1)-(2L-1)+1}{2}+1 = R-L$ 得知, 2個取之間的

奇取個取為 $R-L$, 而 $(R-L) \times 2 = 2R-2L$

$= |x_i - x_i'|$

$\Rightarrow$ 當 $s=+1$, $X_i' < \theta < X_i$, num of $\theta = \frac{|X_i - X_i'|}{2}$

by ①, $\sum_{i=1}^{d} g_{+1,i,\theta}(X) \cdot g_{+1,i,\theta}(X') = |X - X'| \times \frac{1}{2}$

$\Rightarrow$ 當 $s=-1$, $X_i' < \theta < X_i$, num of $\theta = \frac{|X_i - X_i'|}{2}$

by ①, $\sum_{i=1}^{d} g_{+1,i,\theta}(X) \cdot g_{+1,i,\theta}(X') = |X - X'| \times \frac{1}{2}$

$\Rightarrow$ 故 $\sum_{i=1}^{d} g_{s,i,\theta}(X) \cdot g_{s,i,\theta}(X') = |X - X'|$ — ③.

(4) $\sum_{i=1}^{d} [g_{s,i,\theta}(X) g_{s,i,\theta}(X') = 1] + \sum_{i=1}^{d} [g_{s,i,\theta}(X) g_{s,i,\theta}(X') = -1]$

$= 2d(R-L)$

$\Rightarrow \Phi(X)\Phi(X') = 2d(R-L) - 2\sum_{i=1}^{d} [g_{s,i,\theta}(X) g_{s,i,\theta}(X') = -1]$

$= 2d(R-L) - 2|X - X'|$

4. 令 $\sum[y_n = g(x)] = 99$

$\sum[y_n \neq g(x)] = 1$

$$\varepsilon_t = \frac{\sum u_n^{(t)}[y_n \neq g x_n]}{\sum u_n^{(t)}} = \frac{\frac{1}{N}}{1} = \frac{1}{N} = 0.01$$

$$\diamond t = \sqrt{\frac{1-0.01}{0.01}}$$

$$\frac{\sum_{n: y_n > 0} u_n^{(2)}}{\sum_{n: y_n < 0} u_n^{(2)}} = \frac{99 \times u_n \times 1/\diamond t}{1 \times u_n \times \diamond t}$$

$$= 99 \times \frac{1}{\diamond t^2} = 99 \times \frac{1}{99} = 1$$

6. 令 $\sum[y_n \neq g(x)] = K$，則 $\sum[y_n = g(x)] = N-K$

令 $\sum u_n^t = U \Rightarrow \varepsilon_t = \frac{K}{N} \cdot \frac{U}{U} = \frac{K}{N}$

$$\alpha_t = \sqrt{\frac{1-\frac{K}{N}}{\frac{K}{N}}} = \sqrt{\frac{N-K}{K}}$$

$\frac{K}{N}$ 錯 $\Rightarrow \frac{K}{N} \cdot \alpha_t$ —①

$\frac{N-K}{N}$ 對 $\Rightarrow \frac{N-K}{N} \cdot 1/\alpha_t$ —②

$$①+② = \frac{K}{N}\left(\frac{\frac{N-K}{K}}{\sqrt{\frac{N-K}{K}}}\right) + \frac{N-K}{N} \cdot \frac{1}{\sqrt{\frac{N-K}{K}}}$$

$$= 2\left(\frac{N-K}{N}\right)\frac{1}{\sqrt{\frac{N-K}{K}}} = 2(1-\varepsilon_t) \times \frac{1}{\alpha_t}$$

$$= 2 \cdot (1-\varepsilon_t)\frac{\sqrt{\varepsilon_t}}{\sqrt{1-\varepsilon_t}}$$

$$= 2\sqrt{(1-\varepsilon_t) \cdot \varepsilon_t}$$

ans: b

7

8. (1) 共抽 N' 筆，每次可選 N 種，共 $N^{N'}$ 組合

(2) 不重複為 $N!/N'!$

(3) $\frac{N^{N'} - (N!/N'!)}{N^{N'}}$ 為抽到重複的机率

```
a is 0.23538550315526954
b is 0.3227683607339432
c is 0.4136862454558523
d is 0.5038926083901621
```

在 $N'=40$ 時机率為 50.3% > 50%

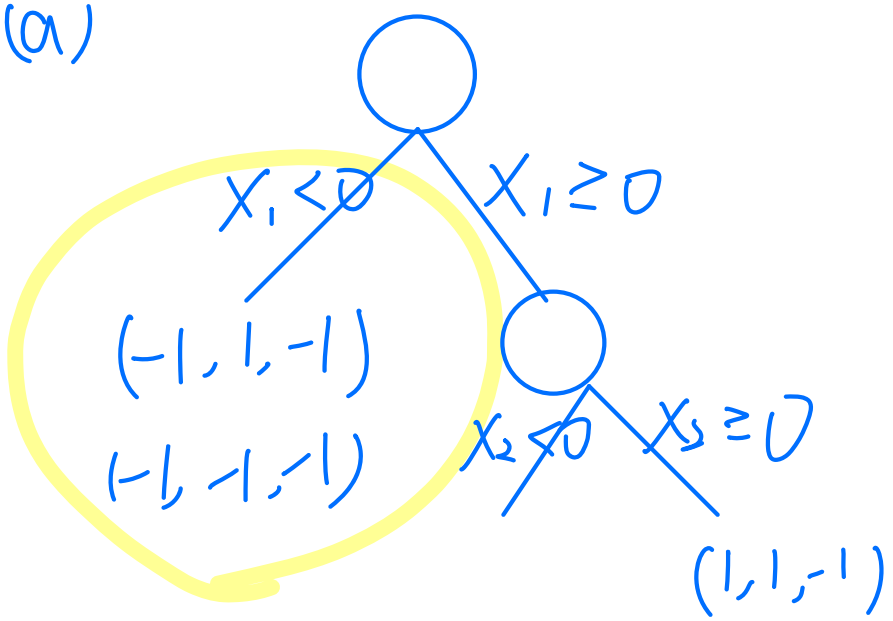
9. by lecture 13 24/51

$$\frac{OOB}{1126} = \left(1 - \frac{1}{1126}\right)^{(1126 \times 2)}$$

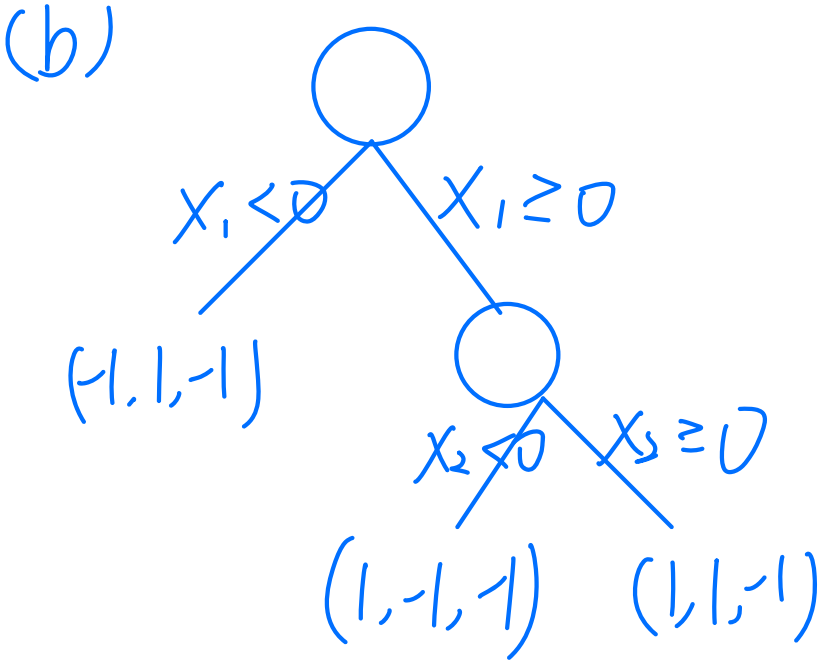
$$\simeq 13.5\ \%\ ,\ \text{ans is d}\ .$$

10.

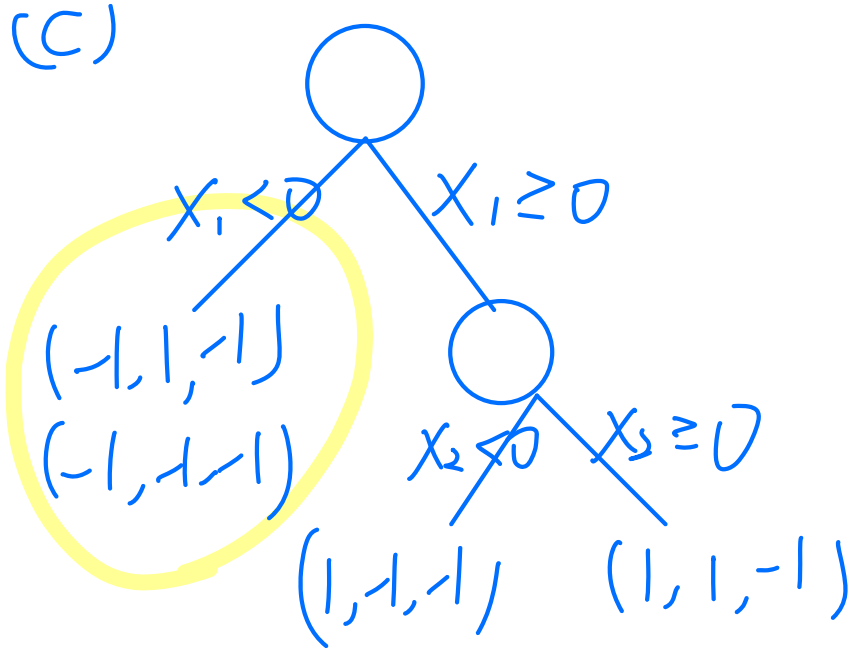
(a)


$X_1 < 0$
$X_1 \geq 0$
(-1, 1, -1)
(-1, -1, -1)
$X_2 < 0$
$X_3 \geq 0$
(1, 1, -1)

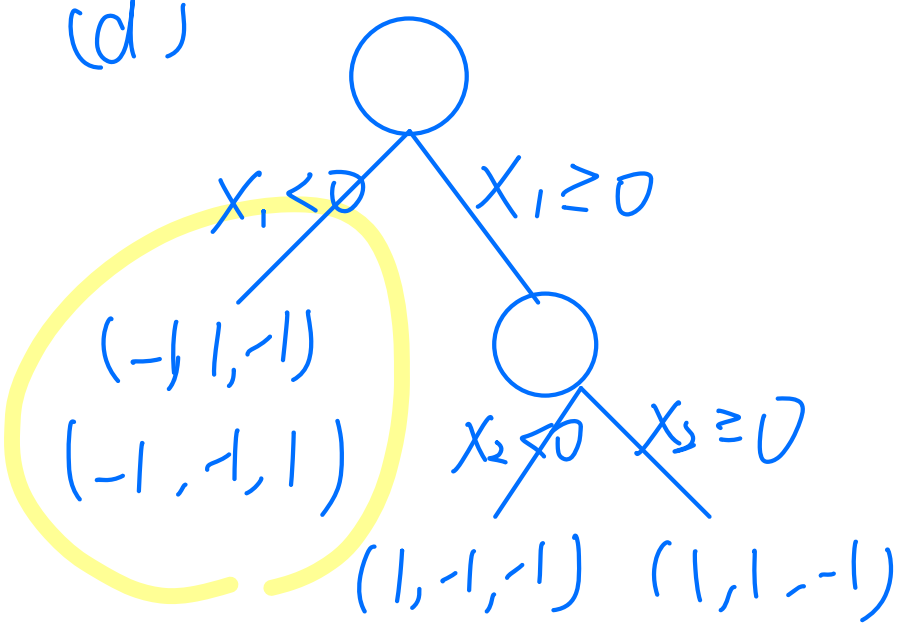
(b)


$X_1 < 0$
$X_1 \geq 0$
(-1, 1, -1)
$X_2 < 0$
$X_3 \geq 0$
(1, -1, -1)
(1, 1, -1)

(c)



(d)



=) 是解不能 shatter 的原因.

```python
import numpy as np
import math
from tqdm import tqdm


def Theta_feature_label(dataPath):
    All_data = []
    with open(dataPath) as file:
        for data in file:
            data = [float(d) for d in data.split()]
            All_data.append(data)
    All_data = np.array(All_data)
    data,label = All_data[:,:-1],All_data[:,-1:]
    label = label.reshape(-1).tolist()


    feature_bag = []
    feature_theta_bag = []
    theta_bag = []
    for i in range(10):
        theta_i_bag = []

        feature =data[:,i:i+1].reshape(-1)
        feature =feature.tolist()
        feature_bag.append(feature)
        # feature_label = np.concatenate((feature,label),axis=1)
        sorted_feature = sorted(feature)

        for j in range(len(sorted_feature)-1):
            theta = (sorted_feature[j]+sorted_feature[j+1])/2
            theta_i_bag.append(theta)
        theta_bag.append(theta_i_bag)
    return feature_bag,label,theta_bag
def get_all_g(feature_bag,label,theta_bag,iter_num):
    numOfData = len(label)
    init_u = np.array([1 / numOfData for _ in range(numOfData)])
    label =np.array(label)
    total_g_alpha = []
    for _ in tqdm(range(iter_num)):
        Ein =float("inf")
        s_list = [-1,1]
        for i in range(10):
            feature_i = feature_bag[i]
            feature_i = np.array(feature_i)
            theta_i =theta_bag[i]
            for s in s_list:
                for theta in theta_i:
                    feature_i_pred = feature_i-theta
                    feature_i_pred[feature_i_pred>=0] = 1
                    feature_i_pred[feature_i_pred<0] = -1
                    feature_i_pred*=s
                    #以上得到h(X)
                    pred_res = feature_i_pred!=label
                    err =np.dot(init_u,pred_res)/numOfData

                    if err < Ein:
                        Ein = err
                        temp_best_s_i_theta = (s,i,theta)
                        yn_g_bool = pred_res

        epsilon_t = np.dot(init_u,yn_g_bool)/np.sum(init_u)
        block_t = ((1-epsilon_t)/epsilon_t) **0.5
        alpha_t = math.log(block_t)
        total_g_alpha.append([alpha_t,temp_best_s_i_theta])

        #update u_t->u_t+1
        scale_table = np.zeros(numOfData)
        # True 代表不等label的部分
        scale_table[yn_g_bool==True] = block_t
        scale_table[yn_g_bool==False] = 1/block_t
        init_u =np.multiply(init_u,scale_table)

    return total_g_alpha
# numOfT :決定要用幾個小g來算結果
def Big_G(feature_bag,label,total_g_alpha,numOfT,alpha_bool = 1):

    numOfData = len(label)
    vote_arr = np.zeros(numOfData)
    all_iter_err = []
    Q13_index = -1
    for T in range(numOfT):
        alpha_t = total_g_alpha[T][0]
        (s,i,theta) = total_g_alpha[T][1]
        # 如果不想要權重，alpha_t=1
        if alpha_bool!=1:
            alpha_t=1

        feature_i = feature_bag[i]
        feature_i = np.array(feature_i)
        pred_feature = feature_i-theta
        pred_feature[pred_feature>=0] = 1
        pred_feature[pred_feature<0] = -1
        pred_feature*=s

        p = pred_feature!=label
        g_err = np.sum(p)/numOfData
        all_iter_err.append(g_err)

        pred_feature*=alpha_t

        vote_arr+=pred_feature

        temp_pred = np.zeros(numOfData)
        temp_pred[vote_arr>=0] = 1
        temp_pred[vote_arr<0] = -1
        temp_err= np.sum(temp_pred!=label)/numOfData
        if temp_err<=0.05 and Q13_index==-1 :
            Q13_index = T


    vote_arr[vote_arr>=0] = 1
    vote_arr[vote_arr<0] = -1
    acc = np.sum(vote_arr==label)/numOfData
    err = 1-acc

    return acc,err,all_iter_err,Q13_index


def Q11(total_g,train_feature_bag,train_label,iter_num):
    Q11_acc,Q11_err,_,_  =Big_G(train_feature_bag,train_label, total_g, iter_num, 0)
    return Q11_err
def Q12(total_g,train_feature_bag,train_label,iter_num):
    _, _,all_err, _ = Big_G(train_feature_bag, train_label, total_g, iter_num, 0)
    id = all_err.index(max(all_err))
    print(f"id is {id}")
    print(f" alpha is {total_g[id][0]}")
    print(f" s,theta is {total_g[id][1]}")
    return max(all_err)
def Q13(total_g,train_feature_bag,train_label,iter_num):
    _, _, _, Q13_ans = Big_G(train_feature_bag, train_label, total_g, iter_num, 1)
    return Q13_ans
def Q14(total_g,test_feature_bag,test_label,iter_num):
    _, Q14_err, _, _ = Big_G(test_feature_bag, test_label, total_g, iter_num, 1)
    return Q14_err

def Q15(total_g,test_feature_bag,test_label,iter_num):
    _, Q_15_err, _, _ = Big_G(test_feature_bag, test_label, total_g, iter_num, 0)
    return Q_15_err
def Q16(total_g,test_feature_bag,test_label,iter_num):
    _, Q_16_err, _, _ = Big_G(test_feature_bag, test_label, total_g, iter_num, 1)
    return Q_16_err



if __name__ == '__main__':
    tra_dataPath = "hw6_train.dat.txt"
    val_dataPath = "hw6_test.dat.txt"
    train_feature_bag, train_label, theta_bag = Theta_feature_label(tra_dataPath)
    iter_num = 500
    # 等同於訓練在train data
    total_g = get_all_g(train_feature_bag, train_label, theta_bag, iter_num)
    val_feature_bag, val_label, theta_bag = Theta_feature_label(val_dataPath)

    Q11_err = Q11(total_g,train_feature_bag,train_label,1)
    Q12_err = Q12(total_g,train_feature_bag,train_label,500)
    Q13_err = Q13(total_g,train_feature_bag,train_label,500)
    Q14_err = Q14(total_g,val_feature_bag,val_label,1)
    Q15_err = Q15(total_g,val_feature_bag,val_label,500)
    Q16_err = Q16(total_g,val_feature_bag,val_label,500)
    print(Q11_err)
    print(Q12_err)
    print(Q13_err)
    print(Q14_err)
    print(Q15_err)
    print(Q16_err)
```