Q1. ans : [d]

若可從顧客的購物資訊，整理出顧客的消費習慣
則可能可以預測顧客之後的消費商品.
然而影響顧客的購買意願的可能原因(參數)
有很多種，並無法輕易判別，故透過ML找出參數和購買
意願的關係是一個好方法.

Q2: ans: [e]

同乘 W_{t+1}

$$W_{t+1} = W_t + y_{n(t)} X_{n(t)} \cdot \eta_t$$

$$\|W_{t+1}\|^2 = W_{t+1}^T W_t + y_{n(t)} W_{t+1}^T X_{n(t)} \cdot \eta_t$$

$$\Rightarrow \frac{\|W_{t+1}\|^2 - W_{t+1}^T W_t}{\eta_t} = y_{n(t)} W_{t+1}^T X_{n(t)} > 0$$

$$\frac{\|W_{t+1}\|^2 - W_{t+1}^T W_t}{\eta_t} > 0$$，假設 $\eta_t > 0$，則 $\|W_{t+1}\|^2 - W_{t+1}^T W_t > 0$

$$\because \|W_{t+1}\|^2 - W_{t+1}^T W_t > 0$$

$$\|W_{t+1}\|^2 > W_{t+1}^T W_t$$

$$= (W_t + y_n X_n \eta_t) W_t$$

$$\|W_{t+1}\|^2 > \|W_t\|^2 + \underbrace{y_n W_t^T X_n \eta_t}_{<0}.$$

$$\Rightarrow \frac{\|W_{t+1}\|^2 - \|W_t\|^2}{y_n W_t^T X_n} < \eta_t$$

$$\Rightarrow \frac{\|W_t + y_n \cdot X_n \cdot \eta_t\|^2 - \|W_t\|^2}{y_n W_t^T X_n} < \eta_t$$

$$\Rightarrow \frac{\left(\cancel{\|W_t\|^2} + \|y_n x_n \eta_t\|^2 + 2 y_n W_t^T x_n \eta_t\right) - \cancel{\|W_t\|^2}}{y_n W_t^T x_n} < \eta_t$$

$$\Rightarrow \frac{\|y_n x_n \eta_t\|^2 + 2 y_n W_t^T x_n \eta_t}{y_n W_t^T x_n} < \eta_t$$

$$\|y_n x_n \eta_t\|^2 + 2 y_n W_t^T x_n \eta_t > \eta_t \cdot y_n \cdot W_n^T x_n$$

$$\Rightarrow \eta_t^2 \|x_n\|^2 + 2 y_n W_t^T x_n \eta_t > \eta_t \cdot y_n W_t^T x_n$$

同除 η_t

$$\eta_t \|x_n\|^2 + 2 y_n W_t^T x_n > y_n W_t^T x_n$$

$$\eta_t \|x_n\|^2 > - y_n W_t^T x_n$$

$$\boxed{\eta_t > \frac{-y_n W_t^T x_n}{\|x_n\|^2} > 0}$$

由於使 $y_{n(t)}w_{t+1}^T x_{n(t)} > 0$ 的 lower bound 如上，

而 (a),(b)(d) 的 learning rate 並無法保證

会大於 $\frac{-y_n w_t^T x_n}{\|x_n\|^2}$，因此排除，

而 (c) 不滿足 $\eta_t > \frac{-y_n w_t^T x_n}{\|x_n\|^2}$ 的條件，

，(e) 的 learning rate 為 $\left\lfloor \frac{-y_n w_t^T x_n}{\|x_n\|^2} + 1 \right\rfloor$

，由於 $\left\lfloor \frac{-y_n w_t^T x_n}{\|x_n\|^2} + 1 \right\rfloor > \frac{-y_n w_t^T x_n}{\|x_n\|^2}$

故只有 (e) 的 η_t 滿足題意.

Q3 : ans : [a]

$\because$ Data is linear separable

$\therefore \exists W_f$ s.t $y_n = \text{sign}(W_f^T X_n)$ $\forall n$

$\Rightarrow y_n W_f^T X_n > 0$.

$\therefore \exists \min_{\forall n} y_n W_f^T X_n \eta_t^* = K > 0$

其中 假設 η_t 是 decrease func, $\eta_t^* \leq \eta_t$ $\forall t$.

(1) $W_f^T W_{t+1} = W_f^T (W_t + y_{n(t)} X_{n(t)} \eta_t)$

$= W_f^T W_t + y_n W_f^T X_{n(t)} \eta_t$.

$\geq W_f^T W_t + K$

$> W_f^T W_t$

(2) $\|W_{t+1}\|^2 = \|W_t + y_n X_n \cdot \eta_t\|^2$

$= \|W_t\|^2 + \|y_n X_n \eta_t\|^2 + \underbrace{2 y_n W_t^T X_n \eta_t}_{< 0}$

$$\le \|W_t\|^2 + \|y_n X_n \eta_t\|^2$$

$$= \|W_t\| + \eta_t^2 \cdot \|X_n\|^2$$

$$\le \|W_t\|^2 + \eta_t^2 \max_{\forall n} \|X_n\|^2$$

$$= \|W_t\|^2 + \eta_t^2 \cdot R^2$$

① $W_f^T W_t \ge W_f^T W_{t-1} + K$

$$\ge W_f^T W_{t-2} + 2K$$

$$\vdots$$

$$\ge W_f^T W_0 + t \cdot K$$ ，其中 $W_0 = 0$.

② $\|W_t\|^2 \le \|W_{t-1}\| + \eta_{t-1}^2 \cdot R^2$

$$\le \|W_{t-2}\| + (\eta_{t-1}^2 + \eta_{t-2}^2) \cdot R^2$$

$$\vdots$$

$$\le \|W_0\| + (\underbrace{\eta_{t-1}^2 + \eta_{t-2}^2 + \dots + \eta_0^2}_{t\text{ 項}}) R^2$$

$$= \|\cancel{W_0}\|^{0} + \sum_{i=1}^{t} (\eta_{t-i}^2) \cdot R^2$$

$$= \sum_{i=1}^{t} (\eta_{t-i}^2) \cdot R^2$$

$$\therefore 1 \geq \cos\theta = \frac{w_f^T w_t}{\|w_t\| \cdot \|w_f\|} \geq \frac{\overset{0}{\cancel{w_f^T w_0}} + t \cdot K}{\|w_f\| \cdot \underbrace{\sqrt{\sum_{i=1}^{t} \eta_{t-i}^2}}_{\text{令为}S.} \cdot R}$$

$$= \frac{t \cdot K}{\|w_f\| \cdot S \cdot R}$$

$$\Rightarrow \frac{\|w_f\| \cdot S \cdot R}{K} \geq t \quad \text{where } K = \exists \min_{\forall n} y_n w_f^T x_n \eta_t^* > 0$$

$$\Rightarrow \frac{\|w_f\| \cdot R \cdot S}{y_n w_f^T x_n \eta_t^*} \geq t$$

$$\Rightarrow \frac{\|w_f\| \cdot R}{y_n w_f^T x_n} \cdot \underbrace{\frac{\sqrt{\sum_{i=1}^{t} \eta_{t-1}^2}}{\eta_t^*}}_{g(t)} \geq \underbrace{t}_{f(t)}.$$

if $\lim_{t \to \infty} \frac{f(t)}{g(t)} \,!= 0$ 则 t 的upper bound会發散,

說明PLA不会停止.

$$\Rightarrow \Omega(g(t)) = \{f(t) : \exists C, t_0 > 0, \text{ s.t. } 0 \leq C g(t) \leq f(t) \ \forall t \geq t_0\}$$

由以上得知，[a] [d] 的 learning rate 所產生的 $g(t)$

隨著 t 成長，$g(t) > f(t)\ \forall t$. (fig 1, fig 3.)

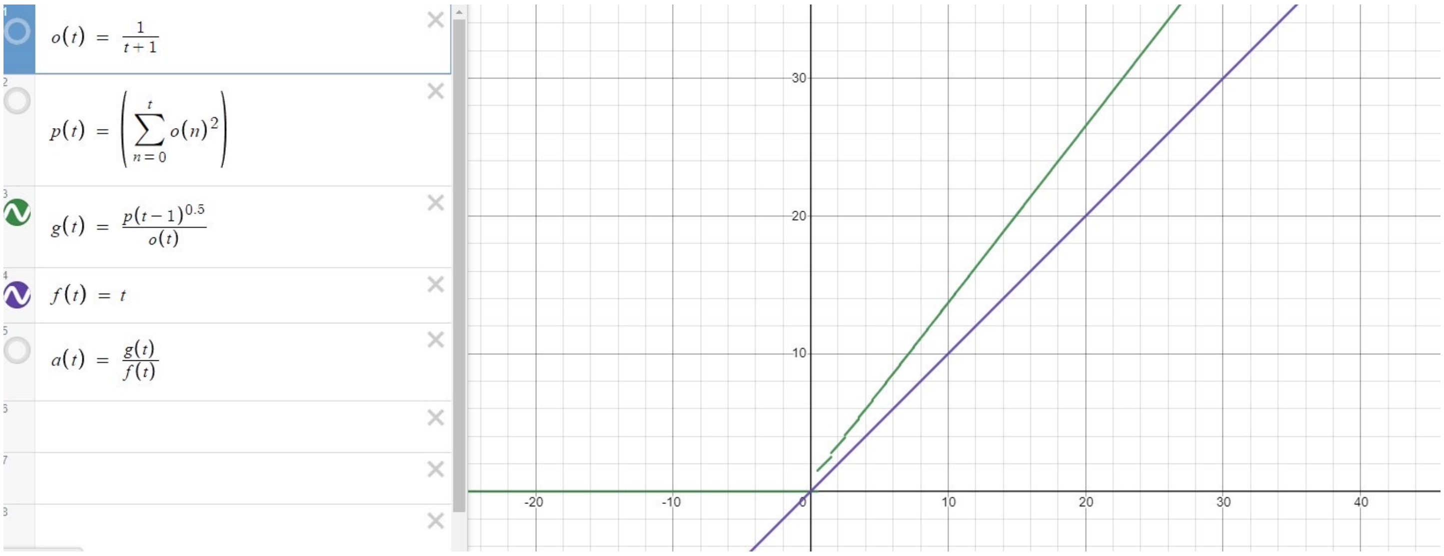
[b] 所代表的 $g(t)$ 成長得比 $t = f(t)$ 慢，故 [b] 可使 PLA 終止.

(fig 2)

fig 1. [a]

fig2. [b]

fig 3. [d]



$o(t) = \frac{1}{t+1}$
$p(t) = \left(\sum_{n=0}^{t} o(n)^2\right)$
$g(t) = \frac{p(t-1)^{0.5}}{o(t)}$
$f(t) = t$
$a(t) = \frac{g(t)}{f(t)}$
30
20
10
-20
-10
0
10
20
30
40

[C] $w_f^T w_{t+1} = w_f^T\left(w_t + y_n x_n \cdot \frac{-y_n w_t^T x_n}{\|x_n\|^2}\right)$

$$= w_f^T w_t + \cancel{y_n} x_n \frac{-\cancel{y_n} w_f^T w_t^T x_n}{\|x_n\|^2}$$

$$= w_f^T w_t + \left(\frac{-w_f^T w_t \; x_n \cdot x_n}{\|x_n\|^2}\right)$$

$$= w_f^T w_t + \left(-w_f^T w_t \cancelto{1}{\frac{\|x_n\|^2}{\|x_n\|^2}}\right)$$

$$= w_f^T w_t + \left(-w_f^T w_t \cdot 1\right)$$

$$= 0$$

由於 $w_f^T w_{t+1}$ 隨著 t 增加，並不会增加

故 w_t 無法与 w_f 越來越相似.

因此找不出 perfect line.

(e)

$$w_f^T w_{t+1} = w_f^T \left(w_t + y_n x_n \cdot \frac{-y_n w_t^T x_n}{\|x_n\|^2} + 1 \right)$$

$$= w_f^T w_t + \cancel{y_n} x_n \frac{-\cancel{y_n} w_f^T w_t^T x_n}{\|x_n\|^2} + y_{n(t)} w_f^T x_{n(t)}$$

$$= w_f^T w_t + \left(\frac{-w_f^T w_t \; x_n \cdot x_n}{\|x_n\|^2} \right) + y_{n(t)} w_f^T x_{n(t)}$$

$$= w_f^T w_t + \left(-w_f^T w_t \cancelto{1}{\frac{\|x_n\|^2}{\|x_n\|^2}} \right) + y_{n(t)} w_f^T x_{n(t)}$$

$$= w_f^T w_t + \left(-w_f^T w_t \cdot 1 \right) + y_{n(t)} w_f^T x_{n(t)}$$

$$= 0 + y_{n(t)} w_f^T x_{n(t)}$$

由於 $y_n w_f^T x_n$ 只說明 $w_f^T w_{t+1}$ 為正,

並不保證內積会隨著 t 增加, 故 PLA 不会停止.

=> 故答案為[a]

4. ans: [c]

$$t \le \frac{R^2}{\rho^2}, \quad R^2 = \max_{\forall n} \|X_n\|^2$$

$$, \rho^2 = \left(\frac{K}{\|W_f\|}\right)^2, \quad K = \min_{\forall n} y_n W_f^T X_n$$

(1), K為 $\min_{\forall n} y_n W_f^T X_n$, 因在 $\mathcal{J}_+(x) = \mathcal{J}_-(x)$ 時,

$y_n W_f^T X_n$ 為 0.5 為最小值.

(2), W_f^T 可判別 spam-like or less spam-like 的 word, 由於 $W_0 = 0$, 且~~必~~包含某 $W_i = 0.5$ 其餘為 1 or -1 共 d 個,

故 $\|W_f\|^2 = d + 0.25$

(3) $R^2 = \max_{\forall n} \|X_n\|^2$, 由於 at most m distinct words in each email 且 $x_0 = 1$, 因此 $R^2 = m+1$

$$\Rightarrow \text{綜合以上}, \ t \le \frac{m+1}{\left(0.5/\sqrt{d+0.25}\right)^2}$$

$$= (4d+1)(m+1)$$

6. ans : [C]

題意：由5個角度拍攝同一物体，

且希望把同一時間的影像

map至同一vector，

意即把同一時間的5張image

分類為同一類時間，

假設有10秒影片，每秒sample一次

，則10張5角度image，共50張

，接著將50張分類至10個秒(類)

，故此為multi-class classification.

7. ans: [C]

key word: 1. 標記 1126張，再自动学習剩下的 1126556張，属於有部份 label 的学習 => semi-supervised learning.

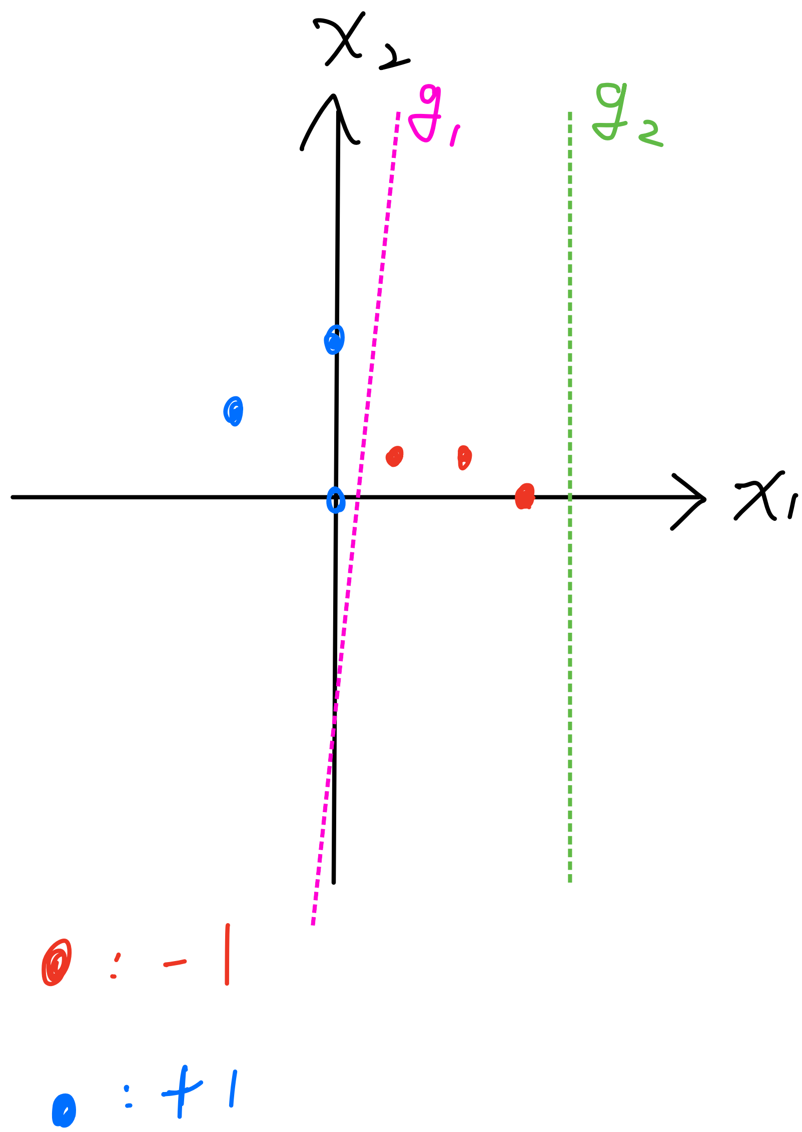
2. 分類成多種類別 => multilabel classification.

3. 一次学習 1126張答案 => batch learning.

4. 由於文章類別的決定應由特殊字的出現次數，
例如：運动：籃球、棒球……
美食：松露、牛排……
等具体描述，而此題只標記分類，故為 raw features.

8. ans: [b]

X	y
(-2,2)	+1
(0,0)	+1
(0,3)	+1
(1,1)	-1
(2,1)	-1
(3,0)	-1



解釋: 1. 選擇 (0,0),(0,3),(1,1) 進行PLA,

此時能得出 g_1, s.t $E_{in}(g_1)=0$, $E_{ots}(g_1)=0$

2. 選擇 (0,0) (0,3) (-2,2) 進行PLA

此時能得出 g_2, s.t $E_{in}(g_2)=0$, $E_{ots}(g_2)=\frac{3}{3}$

$=1$

故 $(\min E_{ots}(g), \max E_{ots}(g)) = (0, 1)$

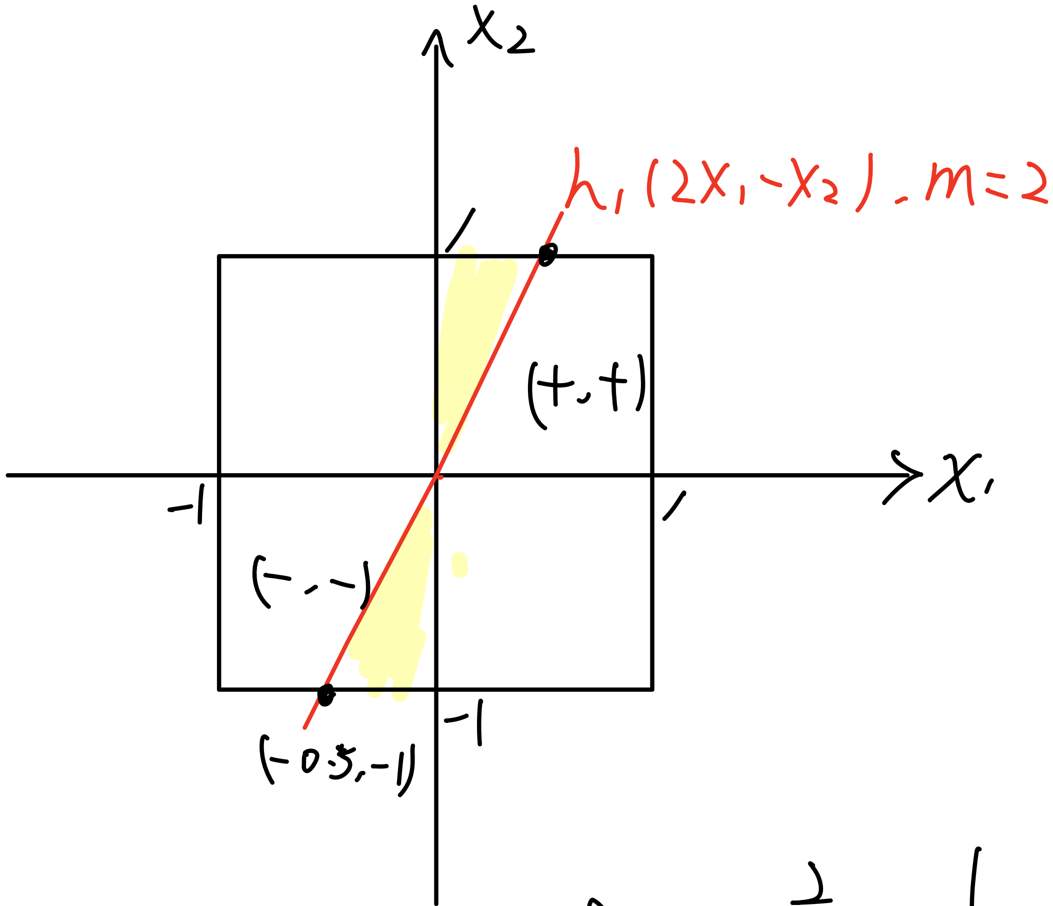
9. ans : [a]

由於由 Ein(g) 推估Eout(g) 的假設前提為

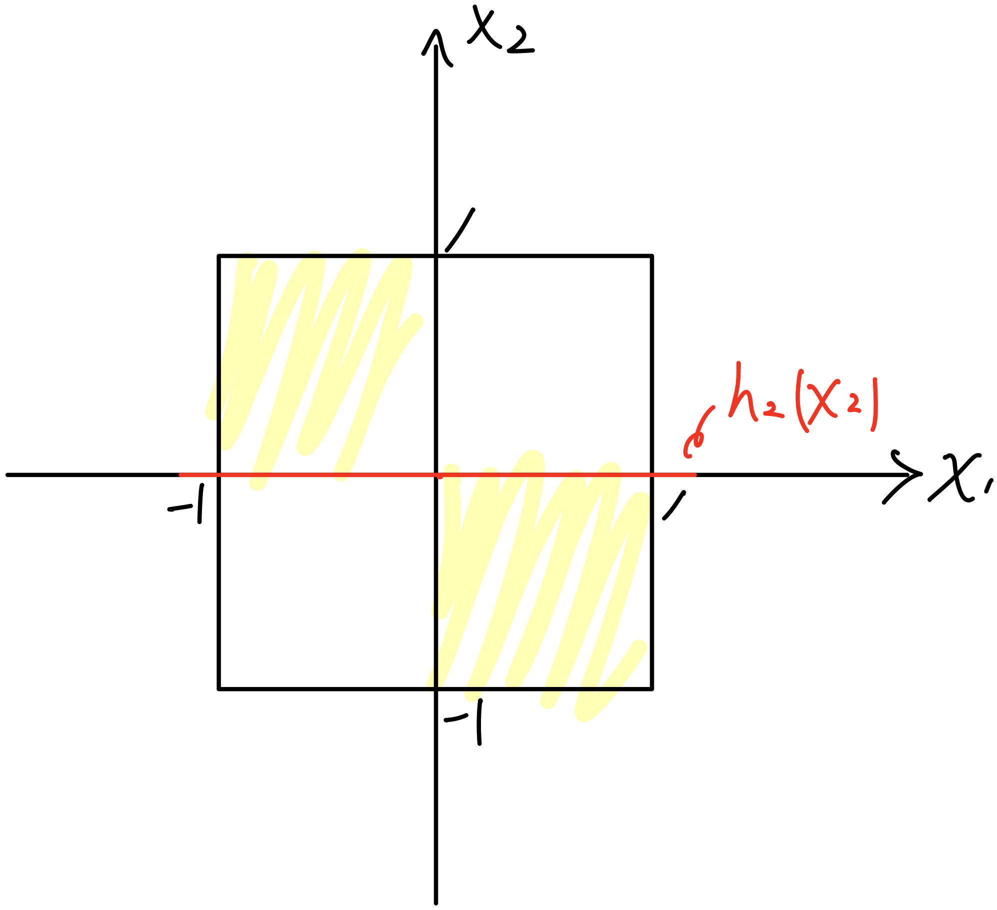
large N and N under iid sampling assumption

，而[a]並不符合iid的條件.

10. ans: [a]



黃色面積為 $h_1(x) \neq f(x)$, $\frac{2}{16} = \frac{1}{8}$



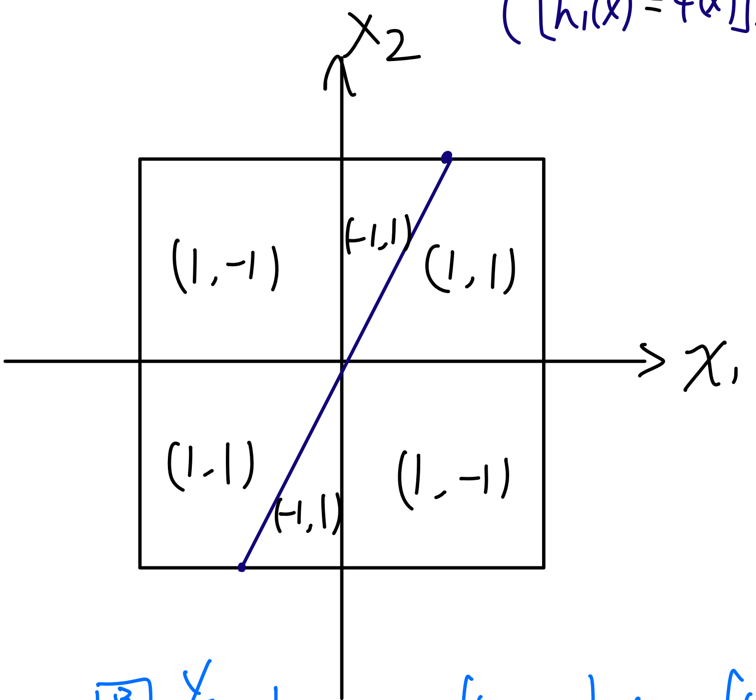
黃色面積為 $h_2(x_2) \neq f(x) = \frac{2}{4} = \frac{1}{2}$

$\Rightarrow$ 故 [a] 為答案.

11. ans: [b]

$$E_{in}(h_1) = E_{in}(h_2) = 0/4,\ 1/4,\ 2/4,\ 3/4,\ 4/4$$

$([h_1(x)=f(x)], [h_2(x)=f(x)])$



圖為 $h_1(x)=f(x)$, $h_2(x)=f(x)$ 的表示圖

$$P(1,1) = \frac{6}{16} = \frac{3}{8}$$

$$P(1,-1) = \frac{1}{2}$$

$$P(-1,1) = \frac{1}{8}$$

$$P(-1,-1) = 0$$

表 1

將 1=O，-1=X 表示

綠4顆 $\begin{matrix} O & O \\ O & O \\ O & O \\ O & O \end{matrix}$ ×1

綠3顆 $\begin{matrix} X & O \\ O & X \\ O & O \\ O & O \end{matrix}$ ×12， $\begin{matrix} X & X \\ O & O \\ O & O \\ O & O \end{matrix}$ ×4

綠2顆 $\begin{matrix} X & X \\ X & X \\ O & O \\ O & O \end{matrix}$ ×6 $\begin{matrix} X & O \\ X & O \\ O & X \\ O & X \end{matrix}$ ×6 $\begin{matrix} X & X \\ O & O \\ X & O \\ O & X \end{matrix}$ ×24

綠1顆 $\begin{matrix} X & X \\ X & X \\ X & X \\ O & O \end{matrix}$ ×4 $\begin{matrix} X & X \\ X & X \\ O & X \\ X & O \end{matrix}$ ×12

綠0顆 $\begin{matrix} X & X \\ X & X \\ X & X \\ X & X \end{matrix}$ ×1

以上為所有 Ein(h₁)=Ein(h₂) 的組合，
接著查表1，將所有机率相加，
因 (X,X) 為 0 故排除出現(X,X)的組合
故剩3組.

最後 $\left(\frac{3}{8}\right)^4+\left(\frac{1}{2}\right)\left(\frac{1}{8}\right)\left(\frac{3}{8}\right)^2\times 12+\left(\frac{1}{2}\right)^2\left(\frac{1}{8}\right)^2\times 6$

$=\frac{609}{4096}$，故 ans：[b]

Q.13 ans: [b]

Q14 ans: [c]

Q15 ans: [e]

Q16 ans: [a]

```python
import numpy as np
import random

def PLA(x0 = 1,scale = 1,normalize = 0):
    # 將data 進行初始化動作，包含x0設定 、 x 的scale 、x 的normalize
    data_bag = []
    with open("./hw1_train.dat.txt") as file:
        for data in file:
            data= [float(d) for d in data.split()]
            data.insert(0,x0)
            if scale > 1:
                for _ in range(11):
                    data[_] = data[_] * scale
            if normalize == 1:
                d = np.array(data[:11])
                norm = np.sqrt(d.dot(d))
                for _ in range(11):
                    data[_] = data[_] /norm
            data_bag.append(data)

    # 以下進行1000次的PLA
    w_pla_bag = []
    for _ in range(1000):
        w_pla = [0 for i in range(11)]
        w_pla = np.array(w_pla)
        counter  = 0
        # 需要從0-99中亂數選出500個數字
        random_seed = np.random.randint(0,100,500)
        # 脫離while迴圈的條件為，連續抽出500筆x，並且利用w_pla_i判別正確，則將此次的w_pla_i記錄在w_pla_bag
        while(counter < 500 ):
            # random_seed 為 0-99的亂數列表 共500個
            for seed in random_seed:
                # 找到第一個隨機點
                point = data_bag[seed]
                # 判別此點是否分類正確
                x = np.array(point[:11])
                label = point[-1]
                dotValue = w_pla.dot(x)
                # (若是內積w_pla 和 point_x 與 label 不同號) or (內積為0)， 則判定此點為錯誤點，因此要更新，並將
counter歸0重新計算。
                if dotValue * label <=  0:
                    w_pla = w_pla + (label * x)
                    counter = 0
                    break
                # 若兩者同號則  dotValue * label > 0 ，此時不做更新
                counter += 1

        w_pla_bag.append(w_pla.dot(w_pla))
    # 將1000次的結果取平均並回傳
    w_pla_average = sum(w_pla_bag) / len(w_pla_bag)
    return w_pla_average

# the Q13 w_pla is 382.70403461590075
# the Q14 w_pla is 1583.7527374916785
# the Q15 w_pla is 7.291336425784612
# the Q16 w_pla is 541.1766270706712
```

Q.13 ans:[b]

Q14 ans:[c]

Q15 ans:[e]

Q16 ans:[a]

```python
import numpy as np
import random

def PLA(x0 = 1,scale = 1,normalize = 0):
    # 將data 進行初始化動作，包含x0設定 、 x 的scale 、x 的normalize
    data_bag = []
    with open("./hw1_train.dat.txt") as file:
        for data in file:
            data= [float(d) for d in data.split()]
            data.insert(0,x0)
            if scale > 1:
                for _ in range(11):
                    data[_] = data[_] * scale
            if normalize == 1:
                d = np.array(data[:11])
                norm = np.sqrt(d.dot(d))
                for _ in range(11):
                    data[_] = data[_] /norm
            data_bag.append(data)

    # 以下進行1000次的PLA
    w_pla_bag = []
    for _ in range(1000):
        w_pla = [0 for i in range(11)]
        w_pla = np.array(w_pla)
        counter  = 0
        # 需要從0-99中亂數選出500個數字
        random_seed = np.random.randint(0,100,500)
        # 脫離while迴圈的條件為，連續抽出500筆x，並且利用w_pla_i判別正確，則將此次的w_pla_i記錄在w_pla_bag
        while(counter < 500 ):
            # random_seed 為 0-99的亂數列表 共500個
            for seed in random_seed:
                # 找到第一個隨機點
                point = data_bag[seed]
                # 判別此點是否分類正確
                x = np.array(point[:11])
                label = point[-1]
                dotValue = w_pla.dot(x)
                # (若是內積w_pla 和 point_x 與 label 不同號) or (內積為0)， 則判定此點為錯誤點，因此要更新，並將
counter歸0重新計算。
                if dotValue * label <=  0:
                    w_pla = w_pla + (label * x)
                    counter = 0
                    break
                # 若兩者同號則  dotValue * label > 0 ，此時不做更新
                counter += 1

        w_pla_bag.append(w_pla.dot(w_pla))
    # 將1000次的結果取平均並回傳
    w_pla_average = sum(w_pla_bag) / len(w_pla_bag)
    return w_pla_average

# the Q13 w_pla is 382.70403461590075
# the Q14 w_pla is 1583.7527374916785
# the Q15 w_pla is 7.291336425784612
# the Q16 w_pla is 541.1766270706712
```

Q.13 ans: [b]

Q14 ans: [c]

Q15 ans: [e]

Q16 ans: [a]

```python
import numpy as np
import random

def PLA(x0 = 1,scale = 1,normalize = 0):
    # 將data 進行初始化動作，包含x0設定 、 x 的scale 、x 的normalize
    data_bag = []
    with open("./hw1_train.dat.txt") as file:
        for data in file:
            data= [float(d) for d in data.split()]
            data.insert(0,x0)
            if scale > 1:
                for _ in range(11):
                    data[_] = data[_] * scale
            if normalize == 1:
                d = np.array(data[:11])
                norm = np.sqrt(d.dot(d))
                for _ in range(11):
                    data[_] = data[_] /norm
            data_bag.append(data)

    # 以下進行1000次的PLA
    w_pla_bag = []
    for _ in range(1000):
        w_pla = [0 for i in range(11)]
        w_pla = np.array(w_pla)
        counter  = 0
        # 需要從0-99中亂數選出500個數字
        random_seed = np.random.randint(0,100,500)
        # 脫離while迴圈的條件為，連續抽出500筆x，並且利用w_pla_i判別正確，則將此次的w_pla_i記錄在w_pla_bag
        while(counter < 500 ):
            # random_seed 為 0-99的亂數列表 共500個
            for seed in random_seed:
                # 找到第一個隨機點
                point = data_bag[seed]
                # 判別此點是否分類正確
                x = np.array(point[:11])
                label = point[-1]
                dotValue = w_pla.dot(x)
                # (若是內積w_pla 和 point_x 與 label 不同號) or (內積為0)， 則判定此點為錯誤點，因此要更新，並將
counter歸0重新計算。
                if dotValue * label <=  0:
                    w_pla = w_pla + (label * x)
                    counter = 0
                    break
                # 若兩者同號則  dotValue * label > 0 ，此時不做更新
                counter += 1

        w_pla_bag.append(w_pla.dot(w_pla))
    # 將1000次的結果取平均並回傳
    w_pla_average = sum(w_pla_bag) / len(w_pla_bag)
    return w_pla_average

# the Q13 w_pla is 382.70403461590075
# the Q14 w_pla is 1583.7527374916785
# the Q15 w_pla is 7.291336425784612
# the Q16 w_pla is 541.1766270706712
```

Q.13 ans: [b]

Q.14 ans: [c]

Q.15 ans: [e]

Q.16 ans: [a]

```python
import numpy as np
import random

def PLA(x0 = 1,scale = 1,normalize = 0):
    # 將data 進行初始化動作，包含x0設定 、 x 的scale 、x 的normalize
    data_bag = []
    with open("./hw1_train.dat.txt") as file:
        for data in file:
            data= [float(d) for d in data.split()]
            data.insert(0,x0)
            if scale > 1:
                for _ in range(11):
                    data[_] = data[_] * scale
            if normalize == 1:
                d = np.array(data[:11])
                norm = np.sqrt(d.dot(d))
                for _ in range(11):
                    data[_] = data[_] /norm
            data_bag.append(data)

    # 以下進行1000次的PLA
    w_pla_bag = []
    for _ in range(1000):
        w_pla = [0 for i in range(11)]
        w_pla = np.array(w_pla)
        counter  = 0
        # 需要從0-99中亂數選出500個數字
        random_seed = np.random.randint(0,100,500)
        # 脫離while迴圈的條件為，連續抽出500筆x，並且利用w_pla_i判別正確，則將此次的w_pla_i記錄在w_pla_bag
        while(counter < 500 ):
            # random_seed 為 0-99的亂數列表 共500個
            for seed in random_seed:
                # 找到第一個隨機點
                point = data_bag[seed]
                # 判別此點是否分類正確
                x = np.array(point[:11])
                label = point[-1]
                dotValue = w_pla.dot(x)
                # (若是內積w_pla 和 point_x 與 label 不同號) or (內積為0)， 則判定此點為錯誤點，因此要更新，並將counter歸0重新計算。
                if dotValue * label <=  0:
                    w_pla = w_pla + (label * x)
                    counter = 0
                    break
                # 若兩者同號則  dotValue * label > 0 ，此時不做更新
                counter += 1

        w_pla_bag.append(w_pla.dot(w_pla))
    # 將1000次的結果取平均並回傳
    w_pla_average = sum(w_pla_bag) / len(w_pla_bag)
    return w_pla_average

# the Q13 w_pla is 382.70403461590075
# the Q14 w_pla is 1583.7527374916785
# the Q15 w_pla is 7.291336425784612
# the Q16 w_pla is 541.1766270706712
```