# Machine Learning Homework 2

資工碩一 r09922055 陳柏妤

1. **(c)**

(a) $(7, 8, 9), (17, 18, 19), (27, 28, 29)$ 三點共線，無法 shatter

(b)(c)(d)(e)

用 $(7, 8, 9), (15, 16, 17), (21, 23, 25)$ 三個點解平面公式

得到 $x_1 - 2x_2 + x_3 = 0$

(b) $(1, 1, 1)$ 在 $x_1 - 2x_2 + x_3 = 0$ 上

(c) $(1, 1, 3)$ 不在 $x_1 - 2x_2 + x_3 = 0$ 上

(d) $(1, 3, 5)$ 在 $x_1 - 2x_2 + x_3 = 0$ 上

(e) $(1, 2, 3)$、$(4, 5, 6)$ 在 $x_1 - 2x_2 + x_3 = 0$ 上

by 2D perceptron 的 $d_{vc} = 3$，(b)(d)(e) 無法 shatter

2. **(d)**

2D perceptron 的 $d_{vc} = 3$，當 N=4 的時候 $m_h(N) < 2^N$，因此 (a)(b)(c) 一定錯

$w_1 w_2 = 0 \Rightarrow w_1 = 0 | w_2 = 0$ （垂直或水平的線）

說明：growth function of axis-aligned perceptrons $\leq 4N - 2$

假設有一組 N 個 input $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$，其中 N 個 x 座標不重複、N 個 y 座標不重複。把這些點投影到 x 軸和 y 軸後，by “ growth function of 1D perceptron”可分別在 x 軸和 y 軸找到 2N 種 perceptron 的組合 ⇒ 共 4N 種

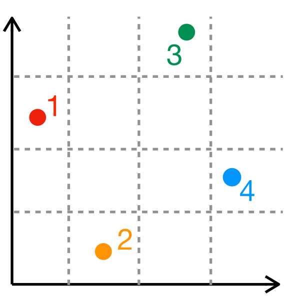
而全部圈和全部叉的組合會被重複計算，扣掉之後會變成 4N-2 種

因此，growth function of axis-aligned perceptrons $\leq 4N - 2$（因為可能有重複）

說明：可以找到一組點使得 growth function of axis-aligned perceptrons $= 4N - 2$

若能找到一組點符合以下條件：「在 x 座標形成的 2N 種 perceptron」與「在 y 座標形成的 2N 種 perceptron」，除了全部圈和全部叉的組合之外，其他的 dichotomy 都不重複。則可以說 growth function of axis-aligned perceptrons $= 4N - 2$

考慮下圖四點的排列，6 條灰色虛線為不同的 hypothesis，可以產生 14 種 dichotomy




垂直線（由左到右）：

OXXX XOOO （正負邊相反）

OOXX XXOO

OOOX XXXO

水平線（由上到下）：

OOXO XXOX

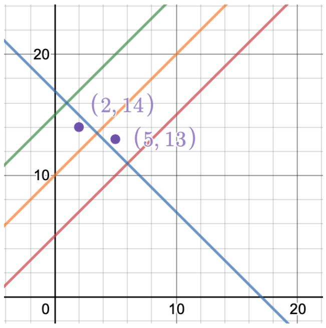
XOXO OXOX

XOXX OXOO

再加上全部都圈、全部都叉兩種組合，共14種（4N-2）

3. **(c)**

(a) 證：$d_{vc} \geq 2$ （There are some 2 inputs we can shatter）




1 $x - y + 5 = 0$ point(2, 14) = $-$, point(5, 13) = $-$

2 $x - y + 10 = 0$ point(2, 14) = $-$, point(5, 13) = $+$

3 $x - y + 15 = 0$ point(2, 14) = $+$, point(5, 13) = $+$

4 $-x - y + 17 = 0$ point(2, 14) = $+$, point(5, 13) = $-$

(b) 證：$d_{vc} \leq 2$（We can not shatter any set of 3 inputs）

假設有三個 input point $(x_1^{(1)}, x_1^{(2)}), (x_2^{(1)}, x_2^{(2)}), (x_3^{(1)}, x_3^{(2)})$

$$X = \begin{bmatrix} 1 & x_1^{(1)} & x_1^{(2)} \\ 1 & x_2^{(1)} & x_2^{(2)} \\ 1 & x_3^{(1)} & x_3^{(2)} \end{bmatrix} , \overrightarrow{x_1} = (1, x_1^{(1)}, x_1^{(2)}), \overrightarrow{x_2} = (1, x_2^{(1)}, x_2^{(2)}), \overrightarrow{x_3} = (1, x_3^{(1)}, x_3^{(2)})$$

・如果 $\overrightarrow{x_1}, \overrightarrow{x_2}, \overrightarrow{x_3}$ linear dependent $\Rightarrow$ by 上課證明，一定不能 shatter

・如果 $\overrightarrow{x_1}, \overrightarrow{x_2}, \overrightarrow{x_3}$ linear independent（可作為三維空間的 basis）

則存在唯一一組 $(a_1, a_2, a_3)$ 使得三維向量 $\vec{z}$ 被表示為：$\vec{z} = a_1\overrightarrow{x_1} + a_2\overrightarrow{x_2} + a_3\overrightarrow{x_3}$

左右同乘 $w^T \Rightarrow w^T\vec{z} = a_1 w^T\overrightarrow{x_1} + a_2 w^T\overrightarrow{x_2} + a_3 w^T\overrightarrow{x_3}$

此時若 $\vec{z} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$，上式會變 $w_0 = a_1 w^T\overrightarrow{x_1} + a_2 w^T\overrightarrow{x_2} + a_3 w^T\overrightarrow{x_3} = a_1 y_1 + a_2 y_2 + a_3 y_3$

因此，當 $w_0 > 0$ 時，三個點無法 shatter

（如：設 $a_1, a_2, a_3$ 皆大於零，必無法產生 $(y_1, y_2, y_3) = (-, -, -)$ 這種 dichotomy）

4. **(b)**

$x_1^2 + x_2^2 + x_3^2$ 可看作是三維空間中的點到原點距離的平方，而此時 ring hypothesis 就可看作是課堂上 “Growth Function for Positive Intervals” 的問題。即：在數線上畫 N 個點（每個點的值是 $x_n$ 到原點距離的平方），求 Positive Intervals 的 Growth Function。

by 課堂上的推導，$m_H(N) = C_2^{N+1} + 1$，答案為 (b)

5. **(b)**

承上題與課堂講義 “The VC Dimension, P.5”，Positive Intervals 的 VC dimension=2。因此 ring hypothesis set 的 VC dimension=2。

6. **(d)**

(a) 證：$|E_{out}(g)-E_{out}(g^*)| \le |E_{out}(g)-E_{in}(g)|+|E_{in}(g^*)-E_{out}(g^*)|$

$$
\begin{aligned}
|E_{out}(g)-E_{out}(g^*)| &= |E_{out}(g)-E_{in}(g)+E_{in}(g)-E_{out}(g^*)| \\
&\le |E_{out}(g)-E_{in}(g)+E_{in}(g^*)-E_{out}(g^*)| \\
&\le |E_{out}(g)-E_{in}(g)|+|E_{in}(g^*)-E_{out}(g^*)| \text{（三角不等式）}
\end{aligned}
$$

(b)證：For every $g$，with probability more than $1-\delta$，$|E_{in}(g)-E_{out}(g)| \le \sqrt{\frac{8}{N}ln(\frac{4m_H(2N)}{\delta})}$

VC Bound：For any $g = A(D) \in H$ and statistical large $D$，

$$P[|E_{in}(h)-E_{out}(h)| > \epsilon] \le 4m_H(2N)exp(-\frac{1}{8}\epsilon^2N)$$

set $\delta = 4m_H(2N)exp(-\frac{1}{8}\epsilon^2N)$

$\Rightarrow \frac{\delta}{4m_H(2N)} = exp(-\frac{1}{8}\epsilon^2N)$

$\Rightarrow ln(\frac{4m_H(2N)}{\delta}) = \frac{1}{8}\epsilon^2N$

$\Rightarrow \sqrt{\frac{8}{N}ln(\frac{4m_H(2N)}{\delta})} = \epsilon$

by Lecture7 P.21，For every $g$，with probability more than $1-\delta$，$|E_{in}(g)-E_{out}(g)| \le \epsilon$

$\Rightarrow |E_{in}(g)-E_{out}(g)| \le \sqrt{\frac{8}{N}ln(\frac{4m_H(2N)}{\delta})}$

(c)證：$|E_{out}(g)-E_{out}(g^*)| \le 2\sqrt{\frac{8}{N}ln(\frac{4m_H(2N)}{\delta})}$

by (a)，
$$
\begin{aligned}
|E_{out}(g)-E_{out}(g^*)| &\le |E_{out}(g)-E_{in}(g)|+|E_{in}(g^*)-E_{out}(g^*)| \\
&\le \sqrt{\frac{8}{N}ln(\frac{4m_H(2N)}{\delta})} + \sqrt{\frac{8}{N}ln(\frac{4m_H(2N)}{\delta})} \\
&\le 2\sqrt{\frac{8}{N}ln(\frac{4m_H(2N)}{\delta})}
\end{aligned}
$$

7. **(d)**

a. 希望 $d_{vc}$ 最大 ⇒ M 個 hypothesis 各產生不同的 M 種 dichotomy 時，$d_{vc}$ 會最大

b. $d_{vc}$ 的意含：最多可 shatter $d_{vc}$ 個點 ⇒ 可對 $d_{vc}$ 個點產生 $2^{d_{vc}}$ 種不同的 dichotomy

綜合以上兩點，可知 $M \ge 2^{d_{vc}} \Rightarrow d_{vc} \le \lfloor log_2 M \rfloor$

8. **(d)**

boolean function 輸出的值只和 input 中 1 的數量有關，而 $\{-1,+1\}^k$ 是 k-dimensional 的資料點，因此 1 的數量只會介於 0~k 之間（k+1 種不同的 input）。此時若這 k+1 個 input 在不同的 *h* 中皆有 -1 和 +1 兩種輸出，則可以對 k+1 個 點產生 $2^{k+1}$ 種 dichotomy ⇒ shattered！

（因為只有 k+1 種不同的 input，若 N>k+1，則必有兩個點的 output 會相同 ⇒ 無法 shatter）

9. **(c)**

$d_{vc} = d \Rightarrow$ 最大的 non-break point = d（d+1 是最小的 break point）$\Rightarrow d_{vc} \geq d$ 且 $d_{vc} \leq d$

・maximum non-break point = d（$d_{vc} \geq d$）：There are ‘some d inputs’ we can shatter

・minimum break point = d+1（$d_{vc} \leq d$）：We can’t not shatter any set of d+1 input

而 “any set of d + 1 distinct inputs is not shattered by H” 也包含了 “some set of d + 1 distinct inputs is not shattered by H”

所以第一點、第六點和最後一點是 necessary condition

10. **(c)**

(c) by https://cs.nyu.edu/~mohri/ml16/sol2.pdf 2(b)

$\{h_\alpha : h_\alpha(x) = sign(sin(\alpha \cdot x))\}$

設有 $m$ ($m \in N$) 個 inputs $(x_1, x_2, \ldots, x_m) = (2^{-1}, 2^{-2}, \ldots, 2^{-m})$

輸出的 labels $(y_1, y_2, \ldots, y_m) \in \{-1, +1\}^m$

此時造一個 $\alpha = \pi(1 + \sum_{i=1}^{m} 2^i y_i'),\ y_i' = \dfrac{1 - y_i}{2}$（即：$y_i' \begin{cases} 1, if\ y_i = -1 \\ 0, if\ y_i = 1 \end{cases}$）

for any $j \in [1, m]$ ($j \in N$)，

$$\alpha \cdot x_j = \alpha \cdot 2^{-j} = \pi(1 + \sum_{i=1}^{m} 2^i y_i') \cdot 2^{-j} = \pi(2^{-j} + \sum_{i=1}^{m} 2^{i-j} y_i')$$

$$= \pi(2^{-j} + \sum_{i=1}^{j-1} 2^{i-j} y_i' + y_j' + \sum_{i=j+1}^{m} 2^{i-j} y_i') = \pi(\sum_{i=1}^{j-1} 2^{-i} y_i' + 2^{-j} + y_j')$$

（橘色區的 i>j ，此時 $\pi \sum_{i=j+1}^{m} 2^{i-j} y_i'$ 為 $2\pi$ 的倍數，對 $sin(\alpha \cdot x)$ 沒影響。省略）

可以得到 $\alpha \cdot x_j = \pi(\sum_{i=1}^{j-1} 2^{-i} y_i' + 2^{-j} + y_j')$ 的 upper 和 lower bound：

・$\pi(\sum_{i=1}^{j-1} 2^{-i} y_i' + 2^{-j} + y_j') \leq \pi(\sum_{i=1}^{j} 2^{-i} + y_j')$（因為 $y_i' = 0\ or\ 1$） $< \pi(1 + y_j')$

・$\pi(\sum_{i=1}^{j-1} 2^{-i} y_i' + 2^{-j} + y_j') > \pi(y_j')$

因此，$\begin{cases} \pi < \alpha \cdot x_j < 2\pi, if\ y_j' = 1, y_j = -1 \\ 0 < \alpha \cdot x_j < \pi, if\ y_j' = 0, y_j = 1 \end{cases}$，則 $h_\alpha(x) \begin{cases} -1, if\ y_j = -1 \\ 1, if\ y_j = 1 \end{cases}$

$\forall m \in N$，無論 $(y_1, y_2, \ldots, y_m)$ 長怎樣，都能造出 $\alpha$ s.t $sign(sin(\alpha \cdot x_i)) = y_i,\ i \in [1, m]$

11. **(d)**

$E_{out}(h,\tau) = (1 - E_{out}(h,0)) \cdot \tau + E_{out}(h,0) \cdot (1-\tau)$

- $(1 - E_{out}(h,0)) \cdot \tau$：原本預測正確的資料有 $\tau$ 的機率預測錯誤
- $E_{out}(h,0) \cdot (1-\tau)$：原本預測錯誤的資料有 $1-\tau$ 的機率仍然是錯的

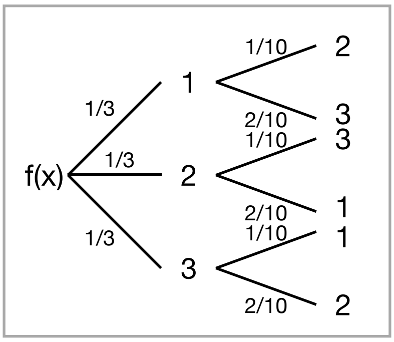
$E_{out}(h,\tau) = \tau - E_{out}(h,0) \cdot \tau + E_{out}(h,0) - E_{out}(h,0) \cdot \tau = \tau + E_{out}(h,0) \cdot (1-2\tau)$

$E_{out}(h,0) = \dfrac{E_{out}(h,\tau) - \tau}{1-2\tau}$

12. **(b)**

如右圖，using squared error 計算 $y \neq f(x)$ 時的 $E_{out}(f)$

$$E_{out}(f) = \frac{1}{3} \cdot (0.1 \cdot (2-1)^2 + 0.2 \cdot (3-1)^2) + \frac{1}{3} \cdot (0.1 \cdot (3-2)^2 + 0.2 \cdot (1-2)^2)$$
$$+\frac{1}{3} \cdot (0.1 \cdot (1-3)^2 + 0.2 \cdot (2-3)^2)$$
$$= \frac{1}{3} \cdot (0.1 + 0.8) + \frac{1}{3} \cdot (0.1 + 0.2) + \frac{1}{3} \cdot (0.4 + 0.2)$$
$$= \frac{1}{3} \cdot (1.8) = 0.6$$




13. **(b)**

- $f(x) = 1$ 時，$f_*(x) = 1 \cdot 0.7 + 2 \cdot 0.1 + 3 \cdot 0.2 = 0.7 + 0.2 + 0.6 = 1.5$
- $f(x) = 2$ 時，$f_*(x) = 2 \cdot 0.7 + 3 \cdot 0.1 + 1 \cdot 0.2 = 1.4 + 0.3 + 0.2 = 1.9$
- $f(x) = 3$ 時，$f_*(x) = 3 \cdot 0.7 + 1 \cdot 0.1 + 2 \cdot 0.2 = 2.1 + 0.1 + 0.4 = 2.6$

$$\Delta(f, f_*) = \frac{1}{3} \cdot (1-1.5)^2 + \frac{1}{3} \cdot (2-1.9)^2 + \frac{1}{3} \cdot (3-2.6)^2 = \frac{0.25 + 0.01 + 0.16}{3} = 0.14$$

14. **(d)**

VC bound：$P[\,|E_{in}(h) - E_{out}(h)| > \epsilon] \leq 4m_H(2N)exp(-\frac{1}{8}\epsilon^2 N)$

by 題目敘述，decision stump model 的 growth function 是 2N、$d_{vc} = 2$

$4m_H(2N)exp(-\frac{1}{8}\epsilon^2 N) = 4(4N)exp(-\frac{1}{8}\epsilon^2 N) = 16Nexp(-\frac{1}{8}\epsilon^2 N)$

帶入 $\epsilon = 0.1,\ \delta = 0.1 \Rightarrow 16Nexp(-\frac{1}{8}\epsilon^2 N) = 16Nexp(-0.00125N)$

依序帶入 N=6000, 8000, 10000, 12000, 14000

$16Nexp(-0.00125N) = 53.096,\ 5.811,\ 0.596,\ 0.059,\ 0.006$

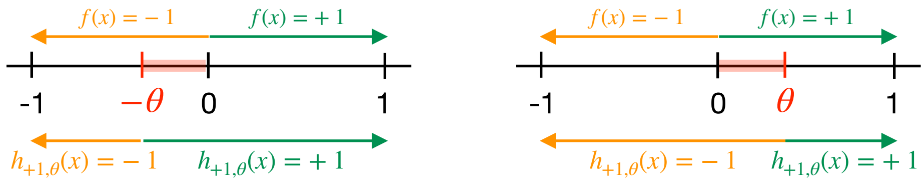
找最小的 N 使得 $16Nexp(-0.00125N) \leq 0.1$，答案為 12000

15. **(b)**

因為 $f(x) = sign(x)$ （即：$f(x \le 0) = -1$、$f(x > 0) = +1$）

$\theta$ 介於 -1 到 1 之間，$h_{+1,\theta}(x \le \theta) = -1$、$h_{+1,\theta}(x > \theta) = +1$

因此會有以下兩種錯誤的情形：




· $\theta < 0$ 時，$-\theta < x \le 0$ 會預測錯誤 $\Rightarrow E_{out}(h_{+1,\theta}, 0) = \dfrac{|\theta|}{2}$

· $\theta \ge 0$ 時，$0 < x \le \theta$ 會預測錯誤 $\Rightarrow E_{out}(h_{+1,\theta}, 0) = \dfrac{|\theta|}{2}$

16. **(d)**
17. **(b)**
18. **(e)**
19. **(c)**
20. **(a)**

In [219]:
```
import numpy as np
def train(times, set_size, tau):
    out_minus_in = 0
    for T in range(times):
        print(T, end='\r')
        ## get x_set, y_set
        x_set = np.sort(np.random.uniform(-1, 1, set_size))
        y_set = np.zeros([set_size])
        num_negative = 0
        for i in range(set_size):
            y_set[i] = np.sign(x_set[i])
            if y_set[i]==0:
                y_set[i] = -1
            temp = np.random.uniform(0, 1)
            if temp<=tau:
                y_set[i] *= -1
            if y_set[i]==-1:
                num_negative += 1
        ##dp_table, E_in
        dp_table = np.zeros([2, set_size])
        dp_table[0][0] = set_size-num_negative ## s=-1 預測錯的個數
        dp_table[1][0] = num_negative ## s=1 預測錯的個數
        for i in range(1, set_size):
            if y_set[i-1]==-1:
                dp_table[0][i] = dp_table[0][i-1] + 1
                dp_table[1][i] = dp_table[1][i-1] -1
            elif y_set[i-1]==1:
                dp_table[0][i] = dp_table[0][i-1] - 1
                dp_table[1][i] = dp_table[1][i-1] + 1
        E_in = np.min(dp_table) / set_size
        ##s, theta
        index = np.unravel_index(np.argmin(dp_table, axis=None), dp
_table.shape)
        if index[0] == 0:
            s = -1
        elif index[0] == 1:
            s = 1
        which_theta = index[1]
        if which_theta == 0:
            theta = -1
        else:
            theta = (x_set[which_theta-1] + x_set[which_theta]) / 2
         ##E_out
        if s == -1:
            E_out = 1 - (0.5 * abs(theta))
        elif s == 1:
            E_out = 0.5 * abs(theta)
        E_out = E_out*(1-(2*tau))+tau

        out_minus_in += (E_out-E_in)
    mean = out_minus_in/times
    return mean
```

In [220]:
```
mean_16 = train(10000, 2, 0)
print("mean_16", mean_16)
```

```
mean_16 0.2907575302757793
```

In [221]:
```
mean_17 = train(10000, 20, 0)
print("mean_17", mean_17)
```

```
mean_17 0.024181985529829673
```

In [222]:
```
mean_18 = train(10000, 2, 0.1)
print("mean_18", mean_18)
```

```
mean_18 0.36722770808211214
```

In [223]:
```
mean_19 = train(10000, 20, 0.1)
print("mean_19", mean_19)
```

```
mean_19 0.05157750106317468
```

In [224]:
```
mean_20 = train(10000, 200, 0.1)
print("mean_20", mean_20)
```

```
mean_20 0.005457546034295806
```