Machine Learning HW2

b06502152, 許書銓

---

1. The answer is (c).

   The problem ask to find which set could be shattered by the hypothesis. From the lecture slide, we have the conclusion that the VC dimension for N-D perceptron is $N+1$. Hence, the VC dimension for 3-D is 4. There is no possiblility for (e) to be shatted, since there are 5 points in that set. For other choice, we are looking for whose rank is exactly 4. (a) rank is 3. (b) rank is 3. (c) rank is 4. (d) rank is 3.

2. The answer should be (d).

   Solution 1

   Since in the scenrio of $N=4$, there are 14 dichotomies, fulfiliing the condition when $4N-2$.

   Solution 2

   We could divide the problem into two direction, which are positive x direction and positive y direction. Now, we have N points on 2D coordinate system. We could find that between each point of them we could find $N-1$ positive X direstion line and $N-1$ positive Y direstion line. Each line could determine the sign value of each point; hence, there would be $2*2*(N-1)$ combinations. Furthermore, we have to consider the all positive points combination and all negative points combination. Thus, the final number of combinations would be,

   $$\begin{aligned} m_H(N) &= 2*2*(N-1)+2 \\ &= 4N-2 \end{aligned}$$

3. The answer shuld be (c).

If we would like to identify the VC dimension of a hypothesis set. If the hypothesis fulfill that some set of d distinct inputs is shatted by H and that any set of $d+1$ distinct inputs is not shatted by H, then we could conclude that the VC dimension of the hypothesis is d.Hence, to find out the VC dimension, we have to divide the problem into to the above-mentioned sub-problems.

Any set of 3 distinct inputs is not shatted by H

From the definition of the hypothesis, we have the hypothesis for 2D perceptron with $w_0 > 0$,

$$\begin{aligned} h_w(x) =& sign(\Sigma_{k=0}^{2} w_i x_i) \\ =& sign(w_0 x_0 + w_1 x_1 + w_2 x_2), (w_0 > 0) \end{aligned}$$

Furthermore, if we try to use matrix to find the output of the hypothesis, assuming $x_0 = c$ we can arrange the upper equation into,

$$\begin{aligned} h_w(x) =& sign(\begin{bmatrix} c & x_{01} & x_{02} \\ c & x_{11} & x_{12} \\ c & x_{21} & x_{22} \end{bmatrix} * \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}) \qquad (3.0) \\ =& sign(\begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix}) \end{aligned}$$

By spanning the upper equation, we have

$$\begin{aligned} y_0 &= cw_0 + x_{01}w_1 + x_{02}w_2 \qquad (3.1) \\ y_1 &= cw_0 + x_{11}w_1 + x_{12}w_2 \qquad (3.2) \\ y_2 &= cw_0 + x_{21}w_1 + x_{22}w_2 \qquad (3.3) \end{aligned}$$

By rearranging the equation 3.1 and 3.2, we could use y_0, y_1 and w_0 to assemble w_1 and w_2, and we swap w_1, w_2 with that outcome. Then, we could rearrange the equation 3.3 into,

$$y_2 = my_0 + ny_1 + kw_0, (w_0 > 0), \qquad (3.4)$$

Since what we care about is the sign value of each y, here, we try to figure out that whether y_0, y_1, y_2 can be generated with 8 distinct dichotomies. However, by considering the sign value of (m,n,k) , we realize that for each (m,n,k) composition, there is one dichotomy that we cannot generate. As the following table shows that the left side is each composition of the sign value of (m,n,k), and the right side is the dichotomy that the corrosponding composition (m,n,k) connot generate since w_0 must larger than 0.

m	**n**	**k**	**l**	y_0	y_1	y_2
+	+	+	l	+	+	-
+	+	-	l	-	-	+
+	-	+	l	+	-	-
+	-	-	l	-	+	+
-	+	+	l	-	+	-
-	+	-	l	+	-	+
-	-	+	l	-	-	-
-	-	-	l	+	+	+

That is, if we look into the first row, whose composition (m, n, k) is (+, +, +). we can never generate the dichotomy of (+, +, +). From the equation 3.4, we have that y_2 is determined by three term. However, with that the first and second term are positive and that the third is positive since k and w_0 are also positive, y_2 must be a positive value. Hence, we sill never generate the dichotomy of (+, +, -) when (m, n, k) is (+, + , +) . The other row shares the same idea. From the upper table, we can see that for every possible (m, n ,k) value, we cannot generate all dichotomies. Thus, we can conclude that **Any set of 3 distinct inputs is not shatted by H**.

Some set of 2 distinct inputs are shatted by H

From the upper equation 3.0, we can arrange the equation into 2 input version, assuming that input set is $\{(x_{01}, x_{02}), (x_{11}, x_{12})\}$.

$$\begin{aligned} h_w(x) =& sign(\begin{bmatrix} c & x_{01} & x_{02} \\ c & x_{11} & x_{12} \end{bmatrix} * \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}) \\ =& sign(\begin{bmatrix} y_0 \\ y_1 \end{bmatrix}) \end{aligned} \tag{3.5}$$

By spanning the upper equation, we have

$$y_0 = cw_0 + x_{01}w_1 + x_{02}w_2 \tag{3.6}$$
$$y_1 = cw_0 + x_{11}w_1 + x_{12}w_2 \tag{3.7}$$

y_0 and y_1 is determined by w_1 and w_2, which will not constrained by the conditoin that $w_0 > 0$. We can tune the value of w_1 and w_2 to generate 4 possible dichotomy of (y_0, y_1). Hence, we could easily shatter on 2 inputs.

By the upper two condition fulfilled, we can conclue that the VC dimension of the positice 2D perceptron is 2.

4. The answer should be (b).

   Since the meaning of $x_1^2 + x_2^2 + x_3^2$ is the $distance^2$ between the origin and the point. Hence, the idea of the hypothesis is like a 3D version of the hypothesis of positive interval in 1D, which means we chould choose 2 points from N+1 interval and plus the condition that choose 2 points from the same interval.

5. The answer should be (b).

   Since the growth function of the hypothesis is $\frac{1}{2}N^2 + \frac{1}{2}N + 1$ and the break point is N = 3. Hence, the VC dimension is 2.

6. The answr should be (d).

   From the lecture note, we have that the equation that

$$0 \leq E_{in}(g) - \sqrt{\frac{8}{N} ln(\frac{4m_H(2N)}{\delta})} \leq E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{N} ln(\frac{4m_H(2N)}{\delta})} \quad (6.0)$$

   And, from the definition of the cheating hypothesis's, we change g_* with g in the upper 6.0 eqution. We would have,

$$E_{out}(g_*) \geq 0 \quad (6.1)$$

   since Error shuold never less than 0, and

$$E_{in}(g) \geq \sqrt{\frac{8}{N} ln(\frac{4m_H(2N)}{\delta})} \quad (6.2)$$

   from the left side of the equation 6.0.

   With the relationship in equation 6.2, we would put the relationship into the right side of equation 6.0, which would yield,

$$
\begin{aligned}
E_{out}(g) &\leq E_{in}(g) + \sqrt{\frac{8}{N} ln(\frac{4m_H(2N)}{\delta})} \\
&\leq 2\sqrt{\frac{8}{N} ln(\frac{4m_H(2N)}{\delta})} \qquad (6.3)
\end{aligned}
$$

Now, we are looking for the upper bond of $E_{out}(g) - E_{out}(g_*)$, we could transfer to the meaning of the upper bond of $E_{out}(g)$ minus the lower bond of $E_{out}(g_*)$. Thus, the equation would yield,

$$
\begin{aligned}
E_{out}(g) - E_{out}(g_*) &\leq 2\sqrt{\frac{8}{N} ln(\frac{4m_H(2N)}{\delta})} - 0 \\
&\leq 2\sqrt{\frac{8}{N} ln(\frac{4m_H(2N)}{\delta})} \qquad (6.4)
\end{aligned}
$$

7. The answer should be (d).

Since if we have M hypothesis, we chould have M dichotomies. From the defnition of the VC dimension, we realize that $M \leq 2^N$. Hence, VC dimension should be $\lfloor lg(M) \rfloor$.

8. The answer should be (d).

The input of the hypothesis can be written in $(e_1, e_2, e_3, \ldots, e_k)$. Based on the one of the hypothesis of the symmetric boolean function, we can detrmine the output by the number of 1's in the input; hence, there are k+1 possible cases, which is zero 1 to k 1's. For convience, we denote the output in $S_0, S_1, S_2, \ldots, S_k$. Obviously, we can shatter on $k+1$ inputs, since every kind of input combination can perfectly catagorize to S_0 to S_k.

However, for $k+2$ inputs, if we try to catogrized every possible combination, we may find that there must exist two of them be put into the same S_i. We cannot shatter on k+2 inputs. Hence, the VC dimension is k+1.

9. The answer should be (b).

The correct condition is that some set of d distinct inputs is shattered by H and that any set of $d+1$ distinct inputs is not shattered by H.

10. The answer should be (c).

To prove the VC dimension is infinite, we have to show that some of the set can be shattered by the H. Hence, if we choose the set to be $(x_1, x_2, x_3, \ldots, x_m)$ and $x_i = 2^{-i}$. Moreover, we would like to choose the parameter ω as $\pi(1 + \Sigma_{i=1}^m 2^i y_i')$, where $y_i' = \frac{1-2_i}{2}$. For any $j \in [1, m]$,

$$\begin{aligned}\omega x = \omega 2^{-j} &= \pi(2^{-j} + \Sigma_{i=1}^{m} 2^{i-j} y_i') \\ &= \pi(2^{-j} + \Sigma_{i=1}^{j-1} 2^{i-j} y_i' + 2^{j-j} y_j' + \Sigma_{i=j+1}^{m} 2^{i-j} y_i') \\ &= \pi(2^{-j} + \Sigma_{i=1}^{j-1} 2^{i-j} y_i' + y_j' + \Sigma_{i=j+1}^{m} 2^{i-j} y_i') \qquad (10.0)\end{aligned}$$

For the last term, we can see that it will always be the mutiples of 2π since $\pi 2^{i-j}$, where $i - j > 0$. Hence, it would not influence the outcome of the hypothesis, and we can simply ignore it. Considering the first two terms, we have,

$$\omega x = \pi(2^{-j} + \Sigma_{i=1}^{j-1} 2^{i-j} y_i' + y_j') \qquad (10.1)$$

As we try to bound ωx, we can show the upper bond that,

$$\omega x = \pi(2^{-j} + \Sigma_{i=1}^{j-1} 2^{i-j} y_i' + y_j') \leq \pi(\Sigma_{i=1}^{j} 2^{-i} + y_j') \leq \pi(1 + y_j') \qquad (10.2)$$

and the lower bond that,

$$\omega x = \pi(2^{-j} + \Sigma_{i=1}^{j-1} 2^{i-j} y_i' + y_j') \geq \pi y_j' \qquad (10.3)$$

Hence, if $y_j = 1$, it turn out that y_j' to be 0. From the upper 10.2 and 10.3 equations, we can bound like $0 \leq \omega x \leq \pi$, where $h(x_i) = 1$ On the other hand, if $y_j = -1$, it turn out that y_j' to be 1. we can bound like $\pi \leq \omega x \leq 2\pi$, where $h(x_i) = -1$.

From the upper scenerio, we could generate all possible dichotomies, and thus shatter the condition. Hence, the VC dimension of sine is infinite.

11. The answer should be (d).

We would like to know the edited out sample error. The possibility of the edited out sample error is that the original out sample error times the possibility not to flip and the possibility of the original correct one times the possibility to flip. Mathematically,

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

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

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

12. The answer should be (b).

    We could convert the probability to the following table,

	f(X) = 1	**f(x) = 2**	**f(x) = 3**
y = 1	0.7	0.2	0.1
y = 2	0.1	0.7	0.2
y = 3	0.2	0.1	0.7

$$E_{out}(f) = \frac{1}{3}\Sigma_{k=1}^{3} E_{out}(k)$$

$$E_{out}(f) = \frac{1}{3}[(1*0.1 + 4*0.2) + (1*0.2 + 1*0.1) + (4*0.1 + 1*0.2)]$$

$$E_{out}(f) = 0.6$$

13. The answer should be (b).

	$f_*(X) = 1$
f(x) = 1	1 * 0.7 + 2 * 0.1 + 3 * 0.2 = 1.5
f(x) = 2	1 * 0.2 + 2 * 0.7 + 3 * 0.1 = 1.9
f(x) = 3	1 * 0.1 + 2 * 0.2 + 3 * 0.7 = 2.6

$$\Delta(f, f_*) = \frac{1}{3}[(1 - 1.5)^2 + (2 - 1.9)^2 + (3 - 2.6)^2] = 0.14$$

14. The answer should be (d).

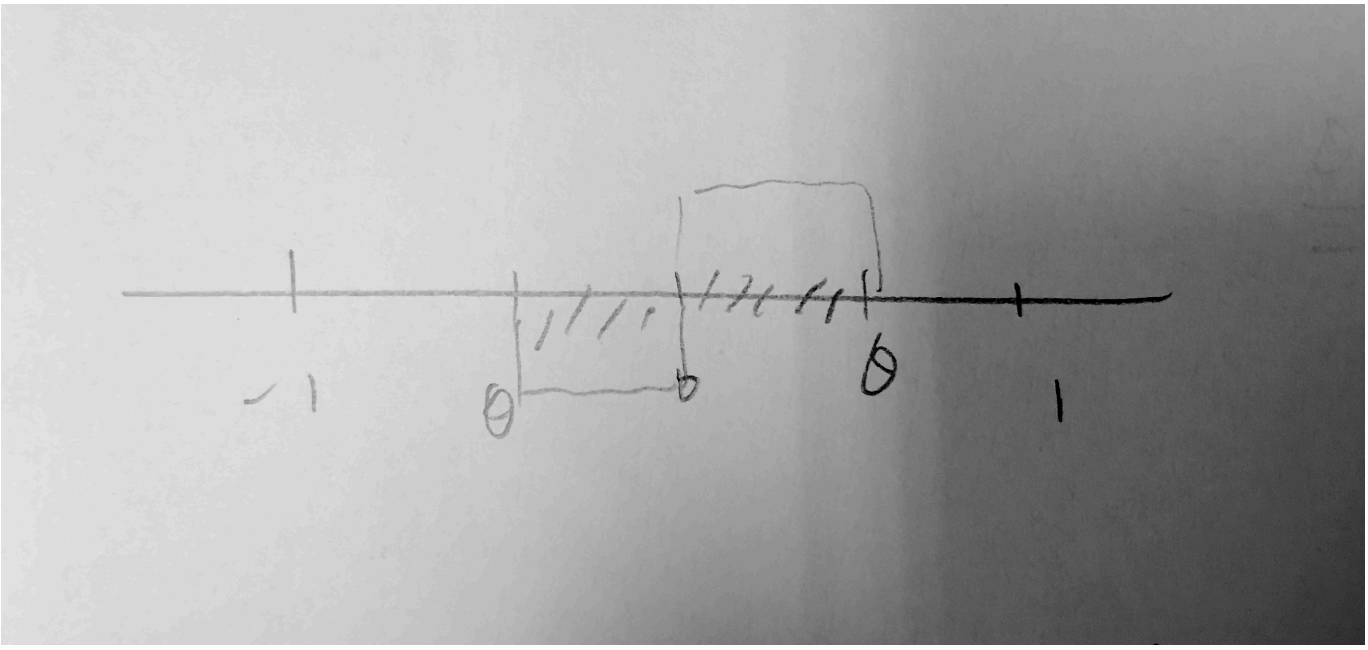
	$\delta = 4(4N) * e^{\frac{-1}{8}\epsilon^2 N}$
N = 6000	53.06
N = 8000	5.811
N = 10000	0.59
N = 12000	0.058
N = 14000	0.005624

Hence, the smallest N for δ to be smaller than 0.1 is 12000.

Or, we try to calculate N in the formula $\delta = 4(4N) * e^{\frac{-1}{8}\epsilon^2 N}$, so that its $\delta \leq 0.1$. From the outcome of wolframalpha, the minimun N is 11543.2. Hence, here we choose the smallest one, which is N = 12000.

15. The answer should be (b)

    We try to figure oyt this problem in 1D line, which looks like,



First of all, we divide the situation of s == 1 into two cases, which $\theta > 0$ and $\theta < 0$.

As for $\theta > 0$ part, we can see that our expectation for E_{out} when $\theta > 0$ is that $(x > 0) \wedge (x < \theta)$. It will corrospond to the upper cross-section, and the possibility chosen in that area is $\frac{\theta - 0}{2} = \frac{\theta}{2}$. On the other hand, for $\theta < 0$ part, the expectation for E_{out} is that $(x < 0) \wedge (x > \theta)$. It will corrospond to the left lower cross-section, and the possibility chosen in that area is $\frac{0-\theta}{2} = \frac{|\theta|}{2}$.

Hence, we calculate upper two possible condition, we have,

$$E_{out}(h_{+1,\theta}, 0) = \frac{1}{2}\frac{\theta}{2} + \frac{1}{2}\frac{|\theta|}{2}$$
$$= \frac{|\theta|}{2}$$

Coding

16. ~ 20. (d)(b)(e)(c)(a)

The code is written in python 3.6.10.

```
import numpy as np
import matplotlib.pyplot as plt
import py_compile

def data_X_generator(size):
    X = np.random.uniform(-1, 1, size)
    return X

def data_Y_generator(X, p):
    Y =  np.sign(X)
    prob = np.random.uniform(0, 1, X.shape[0])
    for i in range(X.shape[0]):
        if prob[i] <= p:
            Y[i] *= -1
    return Y

def data_Y_generator(X, p):
    Y =  np.sign(X)
    prob = np.random.uniform(0, 1, X.shape[0])
    for i in range(X.shape[0]):
        if prob[i] <= p:
            Y[i] *= -1
    return Y

def err_out_p_Minus_err_in(size, p):
    delta_err = []
    delta_com = []

    for i in range(10000):
        ##generator in sample X data
```

```python
        X = data_X_generator(size)

        ##generator in sample Y data:
        Y = data_Y_generator(X, p)

        ##generator theta
        X_sort = np.sort(X, axis = -1)
        theta = []
        for i in range(size - 1):
            theta.append((X_sort[i+1] + X_sort[i])/2 )

        theta.insert(0, (-1))

        err_min = len(theta)
        s_min = 1
        theta_min = theta[-1]
        for i in range(len(theta)):
            # s == 1
            y1 = np.sign(X-theta[i])
            err1 = np.sum(y1 != Y)
            # s == -1
            y2 = np.sign((X-theta[i]) *(-1))
            err2 = np.sum(y2 != Y)

            if err1 < err2 and err1 <= err_min:
                if err1 == err_min and (1+theta[i]) >= (s_min + theta_min):
                    continue
                else:
                    err_min = err1
                    s_min = 1
                    theta_min = theta[i]

            elif err2 < err1 and err2 <= err_min:
                if err2 == err_min and (-1+theta[i]) >= (s_min + theta_min):
                    continue
                else:
                    err_min = err2
                    s_min = -1
                    theta_min = theta[i]

        ##generator out sample X data
        X_out = data_X_generator(size)

        ##generator out sample Y data:
        Y_out_p = data_Y_generator(X_out, p)

        temp_y =  np.sign((X_out - theta_min) * s_min)
        err_out_p = np.sum(temp_y != Y_out_p)

        delta_err.append((err_out_p - err_min)/size)

        if s_min == 1:
```

```
            err_out_0 = abs(theta_min) / 2
        elif s_min == -1:
            err_out_0 = 1 - abs(theta_min) / 2

        err_out_com = err_out_0 * (1-2*p) + p
        delta_com.append(err_out_com - err_min/size)

    #plt.hist(delta_com)
    #plt.show()
    print("mean of (E_{out}(g, tau) - E_in) from sampling: ", np.mean(delta_err))
    print("mean of (E_{out}(g, tau) - E_in) from computing: ", np.mean(delta_com))
    print('\n')

#main
print('Given (input size, tau) = (2, 0)')
err_out_p_Minus_err_in(2, 0)

print('Given (input size, tau) = (20, 0)')
err_out_p_Minus_err_in(20, 0)

print('Given (input size, tau) = (2, 0.1)')
err_out_p_Minus_err_in(2, 0.1)

print('Given (input size, tau) = (20, 0.1)')
err_out_p_Minus_err_in(20, 0.1)

print('Given (input size, tau) = (200, 0.1)')
err_out_p_Minus_err_in(200, 0.1)

#py_compile.compile("/Users/leo/Desktop/hello/ML/hw2/ml_hw2.py")
```