Machine Learning **(Due: 30/10/20)**

Homework #2: Referred Answers

Instructor: Lin, Hsuan Tien *Name:* Hao-Cheng Lo, *Id:* D08227104

Problem 1:

[c] is correct. My explanations of each choice are listed below:

[a] Let $a = (7, 8, 9),\ b = (17, 18, 19),\ c = (27, 28, 29)$. $\because\ \overrightarrow{ab} = \overrightarrow{bc},\ \therefore$ it is impossible for us to find a hyperplane that can classify, for example, the dichotomy $a = +1,\ b = -1,\ c = +1$. Hence, these three data cannot be shattered and [a] is incorrect.

[b] In 3D perceptron, the VC dimension is $3 + 1 = 4$. Accordingly, $N = 4$ data set can be possibly shattered. To check whether such data set can be shattered, here let $M = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 7 & 8 & 9 \\ 1 & 15 & 16 & 17 \\ 1 & 21 & 23 & 25 \end{bmatrix}$. $\because\ \det(M) = 0,\ \therefore\ M$ is singular matrix. Due to the singularity of M (i.e., M^{-1} does not exist), we cannot find w such that $w = M^{-1}y \to Mw = y \to \text{sign}(Mw) = y,\ \forall\ y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$. Hence, these four data cannot be shattered and [b] is incorrect.

[c] Similar to the proof of [b], to check whether [c]'s data set can be shattered, here let $M = \begin{bmatrix} 1 & 1 & 1 & 3 \\ 1 & 7 & 8 & 9 \\ 1 & 15 & 16 & 17 \\ 1 & 21 & 23 & 25 \end{bmatrix}$. $\because\ \det(M) \neq 0,\ \therefore\ M$ is non-singular matrix. Due to the non-singularity of M (i.e., M^{-1} does exist), we can find w such that $w = M^{-1}y \to Mw = y \to \text{sign}(Mw) = y,\ \forall\ y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$. Hence, these four data can be shattered and [c] is correct.

[d] Similar to the proof of [b], to check whether [d]'s data set can be shattered, here let $M = \begin{bmatrix} 1 & 1 & 3 & 5 \\ 1 & 7 & 8 & 9 \\ 1 & 15 & 16 & 17 \\ 1 & 21 & 23 & 25 \end{bmatrix}$. $\because\ \det(M) = 0,\ \therefore\ M$ is singular matrix. Due to the singularity of M (i.e., M^{-1} does not exist), we cannot find w such that $w = M^{-1}y \to Mw = y \to \text{sign}(Mw) = y,\ \forall\ y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$. Hence, these four data cannot be shattered and [d] is incorrect.

[e] Because the VC dimension is $3 + 1 = 4$ and we cannot shatter any set of $N > 4$ inputs, five data of [e] > 4 cannot be shattered. Hence, [e] is incorrect.

Problem 2:

[d] is correct.

We have $D = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$ where $|D| \geq 4$. After sorting D by xs' values, we are able to find at most $2(N-1)$ dichotomies by placing $2(N-1)$ kinds of vertical perceptrons in between xs of $N-1$. In a similar vein, after sorting D by ys' values, we are able to find at most $2(N-1)$ dichotomies by placing $2(N-1)$ kinds of horizontal perceptrons in between ys of $N-1$. Moreover, there are 2 dichonomies of all $+1$ and all -1. Hence, $m_H(N) = 2 \times 2(N-1) + 2 = 4N - 2$.

Problem 3:

[c] is correct.

The VC dimension of 2D-positively-biased-perceptron is 2, because we can find that 2 inputs can be shattered by H but 3 inputs not. For example, given $a = (4, 2)$, $b = (9, 3)$, when $a \in +1$ and $b \in +1$, it can be correctly classified by $H: \ 4x + 2y + 1 = 0$; when $a \in +1$ and $b \in -1$, it can be correctly classified by $H: \ -x + y + 3 = 0$; when $a \in -1$ and $b \in +1$, it can be correctly classified by $H: \ 3x - 9y + 2 = 0$; when $a \in -1$ and $b \in -1$, it can be correctly classified by $H: \ -4x - 2y + 3 = 0$. In this example, all w_0 are positive. However, when there are 3 inputs, we can at most implement correctly via 7 hypotheses, which is $< 2^3 = 8$. Stated another way, because positively biased restricts the range of w_0, the degree of freedom of $\mathbf{w}$ is reduced to 2 (i.e., w_1 and w_2). Hence, the VC-dimension is 2.

Problem 4:

[b] is correct.

We have $D = \{(x_1, y_1, z_1), (x_2, y_2, z_3), ..., (x_N, y_N, z_N)\}$ where $|D| = N$. Then, we compute $x^2 + y^2 + z^2 = v$ of each input and get $V = \{v_1, v_2, ..., v_N\}$. It is noteworthy that at most V has N different values. After sorting V by vs' values, we are able to find at most $\binom{N+1}{2}$ dichotomies by placing a and b in $N + 1$ vs' intervals. Plus, there are one dichonomies of all -1. Hence, $m_H(N) = \binom{N+1}{2} + 1$.

Problem 5:

[b] is correct.

Because $m_H(2) = \binom{2+1}{2} + 1 = 4 = 2^2$ and $m_H(3) = \binom{3+1}{2} + 1 = 7 \neq 8 = 2^3$, the break point of previous problem is 3. Hence, the VC dimension of the ring hypothesis set is 2.

Problem 6:

[d] is correct.

Here is my proof:

$$
\begin{aligned}
E_{out}(g) - E_{out}(g) &= E_{out}(g*) - E_{out}(g*) \\
\rightarrow E_{out}(g) &= E_{out}(g*) - E_{out}(g*) + E_{out}(g) \\
&\leq E_{in}(g*) - E_{in}(g) + E_{out}(g*) - E_{out}(g*) + E_{out}(g) \ (\because\ E_{in}(g*) \geq E_{in}(g)) \\
&= [E_{out}(g) - E_{in}(g)] + [E_{in}(g*) - E_{out}(g*)] + E_{out}(g*) \\
&\leq |E_{in}(g) - E_{out}(g)| + |E_{in}(g*) - E_{out}(g*)| + E_{out}(g*) \ (\because\ |\cdot| \geq 0) \\
&\leq 2 \max_{h \in H} |E_{in}(h) - E_{out}(h)| + E_{out}(g*) \\
&\leq 2\epsilon + E_{out}(g*) \\
\rightarrow E_{out}(g) - E_{out}(g*) &\leq 2\sqrt{\frac{8}{N} \ln\left(\frac{4m_H(2N)}{\delta}\right)} \text{ (see Lecture 7 p.21 derivation)}
\end{aligned}
$$

Problem 7:

[d] is correct.

Suppose there are N inputs and all dichotomies of these N inputs can be implemented by H, thus $|H| = M = 2^N \to N = \lfloor lgM \rfloor$. Plus, because these N inputs can be shattered by H, the VC-dimension of H is $N = \lfloor lgM \rfloor$.

Problem 8:

[d] is correct.

All possible inputs of $\{-1, +1\}^k$ (i.e., 2^k inputs) mapping to $\{-1, +1\}$ have at most 2^{2^k} kinds of hypotheses. However, because symmetric boolean functions do not consider permutation of inputs, there are 2^{k+1} kinds of hypotheses. That is, the inputs where the number of $+1 = 0$ are a kind, the inputs where the number of $+1 = 1$ are a kind, ..., the inputs where the number of $+1 = k$ are a kind, and each kind can map to -1 or $+1$. Accordingly, the VC-dimension of the set of all symmetric boolean functions is $k + 1$.

Problem 9:

[c] is correct.

Let the proposition $d_{VC}(H) = d$ be p and 8 conditions in the problem be $\{q_1, q_2, ..., q_8\}$. If we can derive qs from p (i.e., $p \to q$), then qs are p's necessary conditions.

That the VC-dimension of H is d means that (1) $\exists s \in D$ where $|s| = d$, such that H can shatter s and that (2) $\nexists s \in D$ where $|s| = d + 1$, such that H can shatter s.

Accordingly, we know that some set of d distinct inputs is shattered by H, that some set of $d + 1$ distinct inputs is not shattered by H ($\because$ $\nexists$), and that any set of $d+1$ distinct inputs is not shattered by H ($\because$ $\nexists$ implies $\forall$). Hence, q_1, q_6, q_8 are correct.

Problem 10:

[c] is correct.

Consider a set of x that is $(x_1, ..., x_n) = (2^{-1}, ..., 2^{-n})$ and their corresponding y that is $(y_1, ..., y_n) \in \{+1, -1\}^n$ where $n \in \mathbf{N}$. I claim that we can choose the parameter $\alpha = \pi(1 + \sum_{i=1}^{n} 2^i y'_i)$ where $y'_i = (1 - y_i)/2$. Such parameter can always correctly classify the entire sample. First, lets take a look at αx_j where $j \in [1, n]$. We can find that $\alpha x_j = \alpha 2^{-j} = \pi[2^{-j} + (\sum_{i=1}^{j-1} 2^{i-j} y'_i) + y'_j + (\sum_{i=1}^{m-j} 2^i y'_i)]$.

Because the last term is mutiples of 2π, it can be dropped out in sin function. Simplifying the remaining term, we can get $\pi[2^{-j} + (\sum_{i=1}^{j-1} 2^{i-j} y'_i) + y'_j] = \pi(2^{-j} + \sum_{i=1}^{j-1} 2^{-i} y'_i + y'_j)$.

Since $y'_i \in \{0, 1\}$, the remaining term can be upper bounded as $\pi(2^{-j} + \sum_{i=1}^{j-1} 2^{-i} y'_i + y'_j) \leq \pi(2^{-j} \times 1 + \sum_{i=1}^{j-1} 2^{-i} \times 1 + y'_j) \leq \pi(\sum_{i=1}^{j} 2^{-i} + y'_j) < \pi(1 + y'_j)$. Such term can be lower bounded as $\pi(2^{-j} + \sum_{i=1}^{j-1} 2^{-i} y'_i + y'_j) > \pi y'_j$.

Hence, if $y_j = 1$ that is $y'_j = 0$, then $0 < \alpha x_j < \pi$ and $\sin(\alpha x_j) > 0$. If $y_j = -1$ that is $y'_j = 1$, then $\pi < \alpha x_j < 2\pi$ and $\sin(\alpha x_j) < 0$.

Accordingly, there exists n data $\in \mathcal{R}$ where $n \in \mathbf{N}$ that a set of sin-family H can shatter. The VC-dimension is ∞.

ref: https://cs.nyu.edu/ mohri/ml16/sol2.pdf

Problem 11:

[d] is correct.

Let $E_{out}(h, 0) = p$ which means under the hypothesis h, there are classification errors with probability p. When mislabels come into materialization with probability τ, the correctly classified ones (with probability $1 - p$) in $E_{out}(h, 0)$ would turn out being wrongly classified in $E_{out}(h, \tau)$. On the contrary, when mislabels do not come into materialization with probability $1 - \tau$, the wrongly classified ones (with probability p) in $E_{out}(h, 0)$ would still be wrongly classified in $E_{out}(h, \tau)$. Accordingly, $E_{out}(h, \tau) = \tau(1 - p) + (1 - \tau)p \to p = \dfrac{E_{out}(h, \tau) - \tau}{1 - 2\tau}$.

Problem 12:

[b] is correct.

Due to the uniform distribution, WLOG $x_i,\ \forall\ i$ is the max one with probability $\frac{1}{3}$.

Using the squared error, here provides $E_{out}(f)_i$ when $f(x) = i,\ \forall\ i$.

$E_{out}(f)_1 = \frac{1}{3}\sum_{y=1}^{3} P(y|x)(y - f(x))^2 = \frac{1}{3}[0.7(1-1)^2 + 0.1(2-1)^2 + 0.2(3-1)^2] = 0.3$
$E_{out}(f)_2 = \frac{1}{3}\sum_{y=1}^{3} P(y|x)(y - f(x))^2 = \frac{1}{3}[0.7(2-2)^2 + 0.1(3-2)^2 + 0.2(1-2)^2] = 0.1$
$E_{out}(f)_3 = \frac{1}{3}\sum_{y=1}^{3} P(y|x)(y - f(x))^2 = \frac{1}{3}[0.7(3-3)^2 + 0.1(1-3)^2 + 0.2(2-3)^2] = 0.2$
$E_{out}(f) = 0.3 + 0.2 + 0.1 = 0.6$

Problem 13:

[b] is correct.

First, we compute $f_*(x)_i,\ \forall\ f(x) = i$.

$f_*(x)_1 = 1(0.7) + 2(0.1) + 3(0.2) = 1.5$

$f_*(x)_2 = 2(0.7) + 3(0.1) + 1(0.2) = 1.9$

$f_*(x)_3 = 3(0.7) + 1(0.1) + 2(0.2) = 2.6$

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

Problem 14:

[d] is correct.

Let $\delta = 4m_H(2N)e^{\frac{-1}{8}\epsilon^2 N} = 4(2(2N))e^{\frac{-1}{8}0.1^2 N} = 16Ne^{-0.00125N}$.

When $N = 6000$, $\delta \approx 53.09$.

When $N = 8000$, $\delta \approx 5.81$.

When $N = 10000$, $\delta \approx 0.59$.

When $N = 12000$, $\delta \approx 0.05$.

When $N = 14000$, $\delta \approx 0.005$.

Accordingly, $N = 12000$ is the smallest number such that its $\delta \leq 0.1$.

Problem 15:

[b] is correct.

WLOG an expected condition of D_{out} under h with $s = +1$ would be $y_i = -1, \ \forall \ x_i \in [-1, 0]$ and $y_i = +1, \ \forall \ x_i \in [0, +1]$. When $\theta > 0$, classification error would happen in $[0, \theta]$. Thus, $E_{out}(h_{+1,\theta>0}, 0) = |\theta - 0|/2$ where 2 is domain of θ. When $\theta < 0$, classification error would happen in $[\theta, 0]$. Thus, $E_{out}(h_{+1,\theta<0}, 0) = |0 - \theta|/2$ where 2 is domain of θ. Taken together, $E_{out}(h_{+1,\theta}, 0) = |\theta|/2$.

Problem 16:

[d] is correct. The value ≈ 0.30.

Problem 17:

[b] is correct. The value ≈ 0.02.

Problem 18:

[e] is correct. The value ≈ 0.40.

Problem 19:

[c] is correct. The value ≈ 0.05.

Problem 20:

[a] is correct. The value ≈ 0.00.

Decision Stump

In [2]:
```python
# Libraries
import numpy as np
import random
```

In [27]:
```python
class DS():
    def __init__(self, n, tau): #tau is probability of noise
        self.n = n
        self.tau = tau

    def signf(self, val):
        return -1 if val <= 0 else 1

    def data_generator(self, n):
        if n == 1:
            n = self.n
        else:
            n = 100000
        dta = np.zeros((n,2))
        for i in range(n):
            x = random.uniform(-1,1)
            y = self.signf(x)
            y *= -1 if random.random() <= self.tau else 1
            dta[i] = [x,y]
        return dta

    def theta_generator(self, dta):
        xslist = np.sort(dta[:,0])
        thetalist = np.zeros((self.n,1))
        for i in range(self.n-1):
            thetalist[i+1] = (xslist[i]+xslist[i+1])/2
        thetalist[0] = -1
        return thetalist

    def e_out(self, hypothesis): #simulation solution
        eoutdta = self.data_generator(0)
        s = hypothesis[0]
        theta = hypothesis[1]
        delta = eoutdta[:,1]*np.sign(eoutdta[:,0]-theta)
        e_out_error = (100000 - np.sum(s*delta))/(200000)
        return e_out_error

    def e_out_a(self, hypothesis): #analytical solution
        if hypothesis[0] > 0:
            mu = abs(hypothesis[1])/2
        else:
            mu = 1-abs(hypothesis[1])/2
        return mu*(1-self.tau)+(1-mu)*self.tau

    def experiments(self):
        diff_sum = []
        for exp in range(10000):
            dta = self.data_generator(1)
            theta = self.theta_generator(dta)
            e_in_table = np.zeros((2, self.n))
            e_in_min = float("inf")
            hypothesis = [-2,-2]
            for i in range(self.n):
                # if different sign = -1; same sign = +1
                delta = dta[:,1]*np.sign(dta[:,0]-theta[i])
                # 1-delta/2: if different sign = 1; same sign = 0
                e_in_table[0][i] = (self.n - np.sum(-delta)) / (2 * self.n)
                e_in_table[1][i] = (self.n - np.sum(delta)) / (2 * self.n)
                '''
                for dynamic programming, we can add or minus 1 throughout each (x,y)
                '''
                #have been averaged
                if e_in_table[0][i] < e_in_min:
                    e_in_min = e_in_table[0][i]
                    hypothesis[0] = -1
                    hypothesis[1] = theta[i]
                if e_in_table[1][i] < e_in_min:
                    e_in_min = e_in_table[1][i]
                    hypothesis[0] = 1
                    hypothesis[1] = theta[i]
            diff_sum.append(self.e_out_a(hypothesis)-e_in_min)
        print(np.mean(diff_sum))
```

Problem 16

In [43]:
```
Q16 = DS(2,0)
```

In [44]:
```
Q16.experiments()
```

```
0.29252152810820115
```

Problem 17

In [45]:
```
Q17 = DS(20,0)
```

In [46]:
```
Q17.experiments()
```

```
0.023852933994112722
```

Problem 18

In [47]:
```
Q18 = DS(2,0.1)
```

In [48]:
```
Q18.experiments()
```

```
0.3701313008841667
```

Problem 19

In [49]:
```
Q19 = DS(20, 0.1)
```

In [50]:
```
Q19.experiments()
```

```
0.05064484609972458
```

Problem 20

In [40]:
```
Q20 = DS(200,0.1)
```

In [42]:
```
Q20.experiments()
```

```
0.0050604254939029016
```

In [ ]:
```

```