Machine Learning Homework 1

1

My answer is [D]. We can collect the data of each user's information and order history. Then calculate their chances of making a purchase in the next 7 days by machine learning. After all, sorting the prediction to get the result.

2

We have the following equations

$$w_{t+1} = w_t + y_{n(t)}\ x_{n(t)} \cdot \eta_t \tag{1}$$

$$y_{n(t)}\ w_{t+1}\ x_{n(t)} > 0 \tag{2}$$

Substitute (1) into (2)

$$y_{n(t)}\ w_{t+1}\ x_{n(t)} = y_{n(t)}\ w_t\ x_{n(t)} + \|y_{n(t)}x_{n(t)}\|^2 \cdot \eta_t > 0$$

We know that $\|y_{n(t)}\|^2 = 1$, so we can get

$$\|x_{n(t)}\|^2 \cdot \eta_t > -\ y_{n(t)}\ w_t\ x_{n(t)}$$

$$\eta_t > \frac{-\ y_{n(t)}\ w_t\ x_{n(t)}}{\|x_{n(t)}\|^2}$$

Only [e] always satisfy the condition.

3

Assume exists perfect w_f such that $y_n = sign(w_f^T x_n)$, we have

$$w_f^T w_{t+1} = w_f^T(w_t + y_{n(t)}x_{n(t)\eta_t}) \geq \min_n y_n w_f^T x_n \eta_t$$

$$\|w_{x+1}\|^2 = \|w_t + y_{n(t)}x_{n(t)}\eta_t\|^2 \leq \|w_t\|^2 + \max_n \|x_n\|^2 \eta_t^2$$

Implies

$$1 \geq \frac{w_f^T w_T}{\|w_f\|\|w_T\|} \geq \frac{\sum_{t=1}^T \min_n y_n w_f^T x_n \eta_t}{\sqrt{\sum_{t=1}^T \max_n \|x_n^2\|\eta^2}}$$

Let $\min_n y_n w_f^T x_n = \rho$ and $\max_n \|x_n\|^2 = R^2$, we have

$$1 \geq \frac{\rho}{R}\sqrt{\sum_{t=1}^{T} \eta_t}$$

For [b], [c] and [e], we can ensure that η_t always greater than a constant value l. Thus,

$$\frac{R}{\rho} \geq \sqrt{Tl}$$

$$T \leq \frac{1}{l}(\frac{R}{\rho})^2$$

The answer is [c].

4

By $f(x) = sign(z_+(x) - z_-(x) - 0.5)$, we know there exists perfect w_f that satisfy

$$w_f \in \mathbb{R}^{d+1}$$
$$w_f = [-0.5,..., -1, ..., 1, ...]$$

where -0.5 represent the threshold and 1, -1 correspond to the word is whether in z_+ or z_-.
We already know

$$w_f^T w_{t+1} \geq w_f^T w_t + \min_n y_n w_f^T x_n$$
$$\|w_{t+1}\|^2 \leq \|w_t\|^2 + \max_n \|x_n\|^2$$

Then calculate the min and max of the value

$$\min_n y_n w_f^T x_n = \min \{0.5, 1.5, 2.5, ...\}$$
$$\max_n \|x_n\|^2 = (m+1)^2$$

The case $\min_n y_n w_f^T x_n = 0.5$ represent the case has equal number of words in z_+, z_-. And a mail has at most m words plus one for threshold, therefore, $\max_n \|x_n\|^2 = (m+1)^2$.
Start from w_0 and takes T errors

$$1 \geq \frac{w_f^T \, w_T}{|w_f\|\|w_T\|} \geq \frac{0.5T}{(m+1)\sqrt{T)}} \Longrightarrow T \leq \frac{(m+1)^2}{0.25} = 4(m+1)^2$$

So the answer is [a].

5

Let $y(n)$ be the correct classification and $y'(n)$ be the prediction. There will be 4 cases.
Case $y(n) = y'(n) = 1$ and $y(n) = y'(n) = 2$: Nothing to do
Case $y(n) = 1, y'(n) = 2$:

$$w_{PLA} \leftarrow w_{PLA} - x_n$$
$$w_1 \leftarrow w_1 + x_n$$
$$w_2 \leftarrow w_2 - x_n$$

Case $y(n) = 2, y'(n) = 1$:

$$w_{PLA} \leftarrow w_{PLA} + x_n$$
$$w_1 \leftarrow w_1 - x_n$$
$$w_2 \leftarrow w_2 + x_n$$

It's obviously that $w_{PLA} = -w_1 = w_2$. The answer is [b].

6

My answer is [d] self-supervised learning. This machine doesn't need labels. Instead, it was trained by mapping the image with similar time stamps from different angle to similar vector. The way it generate labels itself is similar to the concept of self-supervised learning.

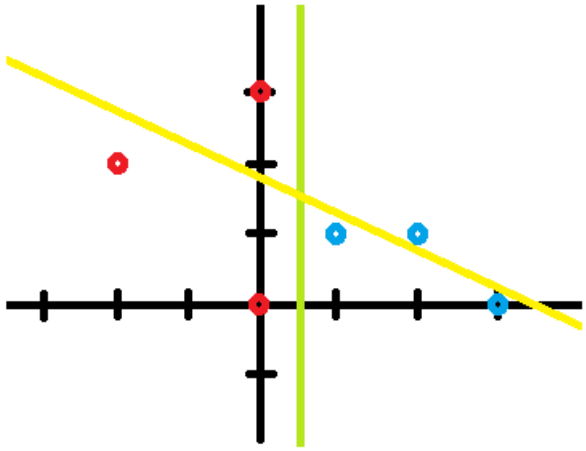
The machine doesn't need to ask the labels of data, so it's not [a] active learning.
The machine doesn't implicit data by goodness, so it's not [b] reinforcement learning.
The machine can do more than classification, so it's not [c] multi-class classification.
Last, the machine doesn't need to ask label from user, so it's not [e] online learning.

7

My answer is [c].
First, the target of the machine is to find out each tag is suitable for the news article or not, so it's multi-label classification.
Second, the machine can use both labeled and unlabeled to learn, thus, it's semi-supervised learning.
Third, the machine can be trained by all known data, so it's batch learning.
Last, the data are news articles, which need to do some NLP to convert it into concrete data, so it's raw features.

8

Let the first variable of x correspond the X-axis and second variable correspond to the Y-axis. We can draw a graph as below. The red point means that y = +1 and blue point means that y = -1.



As we can see, if we choose the point $(0,0),(0,3),(1,1)$, we can construct the boundary indicated by the green line, which lead to $E_{OTS} = 0$.
If we choose the point $(0,3),(1,1),(3,0)$, we can construct the boundary indicated by the yellow line, which lead to $E_{OTS} = 1$.
Thus, the answer is [b].

9

For [e], let

$$\hat{x} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

$$E[\hat{\theta}] = E[\frac{1}{N}\sum_{n=1}^{N}(x_n - \hat{x})^2]$$

So that

$$\begin{aligned}
E[\hat{x}] &= E[\frac{1}{N}\sum_{n=1}^{N} x_n] = \frac{1}{N}E[\sum_{n=1}^{N} x_n]] \\
&= \frac{1}{N}\sum_{n=1}^{N} E[x_n] = \frac{1}{N}\sum_{n=1}^{N}\mu = \frac{1}{N}(N\mu) = \mu = 0 \\
E[\hat{\theta}] &= E[\frac{1}{N}\sum_{n=1}^{N}(x_n - \hat{x})^2] = E[\frac{1}{N}\sum_{n=1}^{N}((x_n - \mu) - (\hat{x} - \mu))^2] \\
&= E[\frac{1}{N}(\sum_{n=1}^{N}(x_n - \mu)^2 - 2\sum_{n=1}^{N}(x_n - \mu)(\hat{x} - \mu) + \sum_{n=1}^{N}(\hat{x} - \mu)^2)] \\
&= \theta - E[(\hat{x} - \mu)^2] = \frac{N-1}{N}\theta \\
E[(\hat{x} - \mu)^2] &= E[(\frac{1}{N}\sum_{n=1}^{N} x_i - \mu)^2] = \frac{1}{N}E[\sum_{n=1}^{N}(x_n - \mu)^2]
\end{aligned}$$

Thus, the unbiased estimator of θ is

$$\hat{\theta} = \frac{1}{N-1}\sum_{n=1}^{N} x_i^2$$

So the answer is [d].

10

$E_{out}(h_1)$:

$$\begin{aligned}
E_{out}(h_1) &= P(0 \le x_1 \le 0.5,\ \ 2 * x_1 \le x_2 \le 1) \\
&\quad + P(-0.5 \le x_1 \le 0,\ \ -1 \le x_2 \le 2 * x_1) \\
&= \frac{1}{1-(-1)} \int_0^{0.5} \frac{1-2x_1}{1-(-1)} dx_1 + \frac{1}{1-(-1)} \int_{-0.5}^{0} \frac{2x_1-1}{1-(-1)} dx_1 \\
&= \frac{1}{16} + \frac{1}{16} = \frac{1}{8}
\end{aligned}$$

$E_{out}(h_2)$:

$$\begin{aligned}
E_{out}(h_2) &= P(0 \le x_1 \le 1,\ \ -1 \le x_2 \le 0) \\
&\quad + P(-1 \le x_1 \le 0,\ \ 0 \le x_2 \le 1) \\
&= \frac{1-0}{1-(-1)} * \frac{0-(-1)}{1-(-1)} + \frac{0-(-1)}{1-(-1)} * \frac{1-0}{1-(-1)} \\
&= \frac{1}{2}
\end{aligned}$$

So the answer is [a].

11

First, we can calculate the probability below

$$P(h_1(x) = -1, h_2(x) = -1) = \frac{3}{16}$$
$$P(h_1(x) = -1, h_2(x) = +1) = \frac{5}{16}$$
$$P(h_1(x) = +1, h_2(x) = -1) = \frac{5}{16}$$
$$P(h_1(x) = +1, h_2(x) = +1) = \frac{3}{16}$$

Second, list all the cases that satisfy $E_{in}(h_1) = E_{in}(h_2)$
Case 1, 4 (+) 0 (-):

$$\binom{h_1}{h_2} = \binom{+ \quad + \quad + \quad +}{+ \quad + \quad + \quad +} = \frac{4!}{4!}\,(\frac{3}{16})^4 = (\frac{3}{16})^4$$

Case 2, 3 (+) 1 (-):

$$\binom{h_1}{h_2} = \binom{+ \quad + \quad + \quad -}{+ \quad + \quad + \quad -} = \frac{4!}{3!}\,(\frac{3}{16})^4 \qquad = 4\,(\frac{3}{16})^4$$
$$\binom{h_1}{h_2} = \binom{+ \quad + \quad + \quad -}{+ \quad + \quad - \quad +} = \frac{4!}{2!}\,(\frac{3}{16})^2(\frac{5}{16})^2 \quad = 12\,(\frac{3}{16})^2(\frac{5}{16})^2$$

Case 3, 2 (+) 2 (-):

$$\binom{h_1}{h_2} = \binom{+ \quad + \quad - \quad -}{+ \quad + \quad - \quad -} = \frac{4!}{2!2!}\,(\frac{3}{16})^4 \qquad = 6\,(\frac{3}{16})^4$$
$$\binom{h_1}{h_2} = \binom{+ \quad + \quad - \quad -}{+ \quad - \quad + \quad -} = \frac{4!}{1}\,(\frac{3}{16})^2(\frac{5}{16})^2 \quad = 4\,(\frac{3}{16})^2(\frac{5}{16})^2$$
$$\binom{h_1}{h_2} = \binom{+ \quad + \quad - \quad -}{- \quad - \quad + \quad +} = \frac{4!}{2!2!}\,(\frac{5}{16})^4 \qquad = 6\,(\frac{5}{16})^4$$

Case 4, 1 (+) 3 (-): Similar to Case 2.
Case 5, 0 (+) 4 (-): Similar to Case 1.
Sum of 5 cases:

$$16\,(\frac{3}{16})^4 + 48\,(\frac{3}{16})^2(\frac{5}{16})^2 + 6\,(\frac{5}{16})^4 = \frac{7923}{32768}$$

I choose the answer closet to my result, which is [d].

12

We can easily observed that both {A, B, C}'s 6 and {A, B, D}'s 2 are colored green. Thus, just count the combination of these cases and minus the overlapping part {A, B}.

$$\frac{3^5 + 3^5 - 2^5}{4^5} = \frac{454}{1024}$$

So the answer is [b].

13, 14, 15, 16

Code:

```
import numpy as np
import random
import time

def sign(x):
    if x > 0:
        return 1
    else:
        return -1

def PLA(x, y):
    w = np.zeros(x.shape[1])
    t = 0
    random.seed(time.time())

    while True:
        rd = random.randint(0, x.shape[0]-1)
        pred = sign((w.T).dot(x[rd]))
        if pred == y[rd][0]:
            if t > 5 * x.shape[0]:
                break
        else:
            w += x[rd] * y[rd]
        t += 1

    return w

def preprocess(x):
    prob = 13
    print("The result of problem", prob, "is:")

    if prob == 16:
        x = np.hstack((np.zeros((x.shape[0], 1)), x))
    else:
        x = np.hstack((np.ones((x.shape[0], 1)), x))

    if prob == 14:
        x *= 2
    elif prob == 15:
        for i in range(x.shape[0]):
            norm = np.linalg.norm(x[i])
            x[i] /= norm

    #print(x)

    return x
```

```
def main():
    data = np.loadtxt('hw1_train.dat')

    x, y = np.hsplit(data, [-1])
    x = preprocess(x)
    #print(x.shape, y.shape)

    t = 1000
    tot = 0
    for i in range(t):
        w = PLA(x, y)
        tot += w.dot(w)
    print(tot / t)

    return

if __name__ == '__main__':
    main()
}
```

main: Read data, send it to the function below, then calculate the average squared length.

preprocess: Add w_0 to the data and preprocess the data according to the problem. Change the variable "*prob*" to get the result for different problem.

PLA: Do the PLA for at least 500 rounds. return the trained w.

sign: Return the sign value.

The answer nearest to my experiments for each problem is [b], [c], [e], [a].