1、

(d) there exist an underly pattern about user future purchase. User may decide to purchase if they face some condition that they run out of some resource. So, if we have user history data about purchase, we can predict user future purchase because it may have some relation between history purchase and future purchase. And we can implement it by ML trick.

2 $y_n(t)\, w_{t+1}^T\, x_n(t) = y_n(t)\, x_n(t)^T\, w_{T+1}$

1e) (a) $y_n(t)\, x_n(t)^T w_{T+1} = \underbrace{y_n(t)\, x_n(t)^T w_T}_{-} + \underbrace{|x_n(t)|^2 \cdot 2^{-t}}_{+}$

不一定

(b) $y_n(t)\, x_n(t)^T w_{T+1} = \underbrace{y_n(t)\, x_n(t)^T w_T}_{-} + \underbrace{0.6211\, |x_n(t)|^2}_{+}$

不一定

(c) $y_n(t)\, x_n(t)^T w_{T+1} = y_n(t)\, x_n(t)^T w_T + (-y_n(t)\, w_t^T x_n(t))$

$= 0$

(d) $y_n(t)\, x_n(t)^T w_{T+1} = \underbrace{y_n(t)\, x_n(t)^T w_T}_{-} + \underbrace{|x_n(t)|^2 \frac{1}{1+t}}_{+}$

(e) $y_n(t)\, x_n(t)^T w_{T+1} = y_n(t)\, x_n(t)^T w_T + (-y_n(t)\, w_t^T x_n(t))$

$+ |x_n(t)|^2$

$= |x_n(t)|^2 > 0$

3 (a) $w_f^T w_{T+1} \geq w_f^T w_T + \min_n y_n w_f^T x_n 2^{-T}$

(c) $w_f^T w_1 \geq w_f^T w_0 + \min_n y_n w_f^T x_n 2^{-0}$

3個

$\vdots$

$w_f^T w_T \geq w_f^T w_{T-1} + \min_n y_n w_f^T x_n 2^{-(T-1)}$

$w_f^T w_T \geq \min_n y_n w_f^T x_n (2^{-0} + \cdots 2^{-(T-1)})$

$= \min_n y_n w_f^T x_n (2 - 2^{-t+1})$

$\|w_{T+1}\|^2 \leq \|w_T\|^2 + \max_n \|x_n\|^2 \cdot 2^{-2t}$

$\|w_1\|^2 \leq \|w_0\|^2 + \max_n \|x_n\|^2 \cdot 2^{-2\cdot 0}$

$\vdots$

$\|w_T\|^2 \leq \|w_{T-1}\|^2 + \max_n \|x_n\|^2 \cdot 2^{-2(T-1)}$

$\|w_T\|^2 \leq (2^{-2\cdot 0} + \cdots 2^{-2(T-1)}) \max_n \|x_n\|^2$

$= \frac{4(1-(\frac{1}{4})^T)}{3} \max_n \|x_n\|^2$

$$1 \geq \frac{W_f^T W_T}{|W_f||W_T|} \geq \frac{3 \min_n y_n W_f^T x_n (2-2^{-t+1})}{\sqrt{4(2(1-(\frac{1}{4})^T)/3) \max_n |x_n|^2}}$$

T的遞增函數

此函數有 upper bound

$\Rightarrow$ T 有 upper bound

(b) $W_f^T W_{T+1} \geq W_f^T W_T + \min_n y_n W_f^T x_n (0.6211)$

$W_f^T W_1 \geq W_f^T W_0 + \min_n y_n W_f^T x_n (0.6211)$

$\vdots$

$W_f^T W_T \geq W_f^T W_{T-1} + \min_n y_n W_f^T x_n (0.6211)$

$W_f^T W_T \geq \min_n y_n W_f^T x_n \, T \cdot (0.6211)$

$|W_{T+1}|^2 \leq |W_T|^2 + \max_n |x_n|^2 \cdot 0.6211$

$|W_1|^2 \leq |W_0|^2 + \max_n |x_n|^2 \cdot 0.6211$

,

$$|W_T|^2 \le |W_{T-1}|^2 + \max_n |x_n|^2 \cdot 0.6211$$

$$|W_T|^2 \le T \cdot (0.6211) \max_n |x_n|^2$$

$$1 \ge \frac{W_f^T W_T}{|W_f|\,|W_T|} \ge \frac{0.6211\, T \cdot \min_n x_n W_f^T x_n}{\sqrt{0.6211\, T}\;\max_n |x_n|}$$

T的遞增函數
此函數有 upper bound
⇒ T 有 upper bound

(d) 如果是遞減函數我們可以推得上面的結論

4

$f(x) = \text{sign}(z_+(x) - z_-(x) - 0.5)$

2個相等算負的

(a)

$= \text{sign}(W_f^T X)$

$\Rightarrow W_f^T = [\pm 1, \ldots, \pm 1 - 0.5]$

ith 是 spam +1
不是 spam -1

$$T \leq \left(\frac{\max_n \|x_n\|^2}{\min_n y_n W_f^T x_n} \right)^2 = \left(\frac{m+1}{0.5} \right)^2$$

$$= 4(m+1)^2$$

$x: \begin{bmatrix} \vdots \\ \vdots \\ 1 \end{bmatrix}$ bag of word

at most m word

$\Rightarrow$ at most m+1 個 1

$y_n(t) x_n(t)^T W_{T+1} = y_n(t) x_n(t)^T W_T + (-y_n(t)) W_t^T x_n(t))$

min condition z_+ 和 z_- 一樣

$$\min_n y_n W_f^T x_n = 0.5$$

5

(b) initial

$$\begin{tikzcd} & W_{PLA}^{(0)} = 0 \arrow[dl, "\tilde{y}_n(1) = -1"'] \arrow[dr, "\tilde{y}_n(1) = +1"] & \\ -x_n(0) & & +x_n(0) \\ & & \vdots \end{tikzcd}$$

$$\begin{tikzcd} & W_1^{(0)} = 0 \arrow[dl, "y_n(1) = 1"'] \arrow[dr, "y_n(1) = 2"] & \\ +x_n(0) & & -x_n(0) \end{tikzcd}
\qquad
\begin{tikzcd} & W_2^{(0)} = 0 \arrow[dl, "y_n(1) = 1"'] \arrow[dr, "y_n(1) = 2"] & \\ -x_n(0) & & +x_n(0) \\ & & \vdots \end{tikzcd}$$

run these two algorithm with same order

$\Rightarrow$ W_{PLA} add $x_n(0)$ with same sign as W_2^*

different sign as W_1^*

$$\Rightarrow \quad W_{PLA}^{f} = W_2^{f} = -W_1^{f}$$

6、

(d)

This task is similar to training word embedding which semantic similar word has similar embedding and distant semantic word has different embedding. This task don't have any label. Hence, we should set our label by time stamp relation. So, we can regard it as self-supervised learning.

7

(c)

multilabel: "Each article can belong to several different categories"
we can know by this sentence

semi_supervised: Some training datas have label, some doesn't have. The algorithm is trying to learn from all article.

batch learning: all training data are known before training

raw feature: This article say all training data are stored in utf8 and doesn't do any feature engine

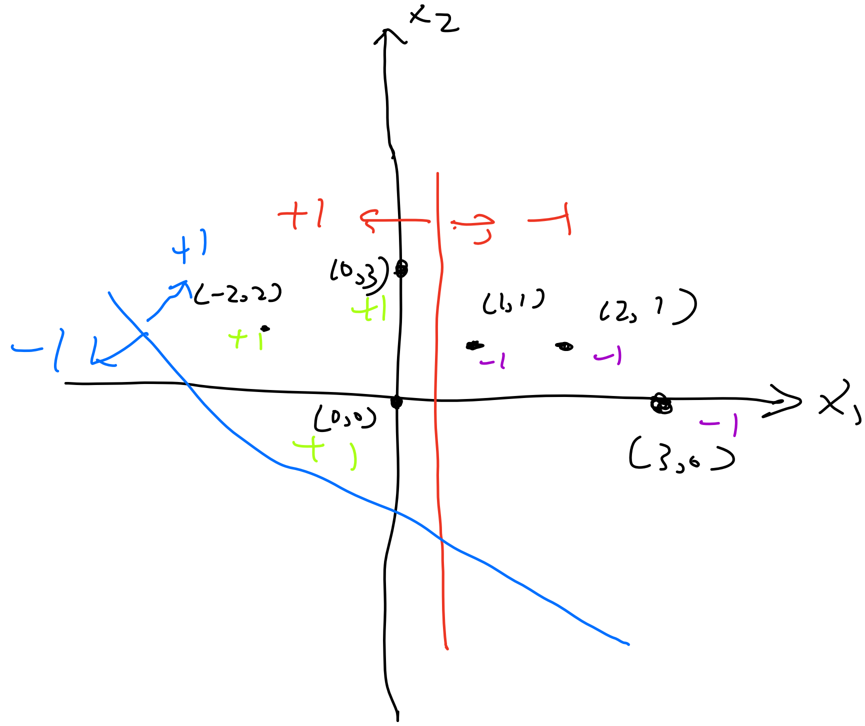
γ, 3 in sample 3 out sample

min

(b) g : $\text{sign}(-x_1 + 0.5) \Rightarrow 0$

max

$\text{sign}(x_1 + x_2 + 0.5) \Rightarrow 1$



9 (a) $E[\hat{\theta}] = \sum_{\{x_1, \dots x_n\} \sim p} \frac{1}{N} \sum_{n=1}^{N} p(x_n) [h(x_n) \neq y_n]$

$= \sum_{x \sim p} p(x) [h(x) \neq f(x)]$

(i) $= \mathop{E}_{x \sim p} [h(x) \neq f(x)]$

(b) $E[\hat{\theta}] = \sum_{\{x_1 \dots x_n\} \sim p} \frac{1}{N} \sum_{n=1}^{N} p(x_n) x_n$

$= \sum_{x \sim p} p(x) x = \sum_{x \sim p} \theta^x (1-\theta)^{(1-x)} x_n$

$= \theta$

(c) $E[\hat{\theta}] = \sum_{\{x_1, \dots x_n\} \sim p} p(x_1 \dots x_n) \max\{x_1, \dots x_n\}$

$= \sum_{k=1}^{M} \left(\left(\frac{1}{k}\right)^n - \left(\frac{1}{k-1}\right)^n \right) k$

$\neq M$

(d) $E[\hat{\theta}] = \sum_{\{x_1, \dots x_n\} \sim p} \frac{1}{N} \sum_{n=1}^{N} p(x_n) x_n^2$

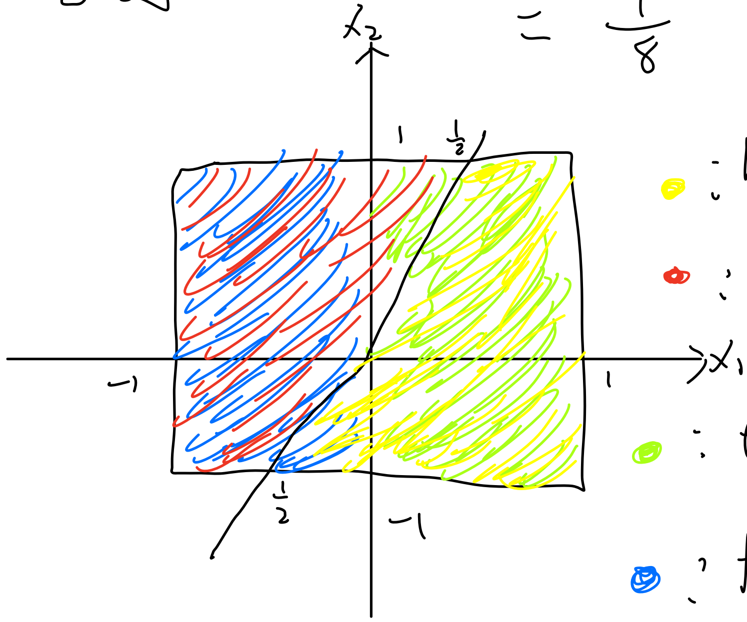
$= \sum_{x \sim p} p(x) x^2 \underset{mean=0}{=} \text{variance} = \theta$

10 $\quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

$E_{out}(h_1) = (\bullet \cap \bullet) \cup (\bullet \cap \bullet)$

$= \frac{1}{8}$

(a)



$\bullet$: $h_1(x) = +1$

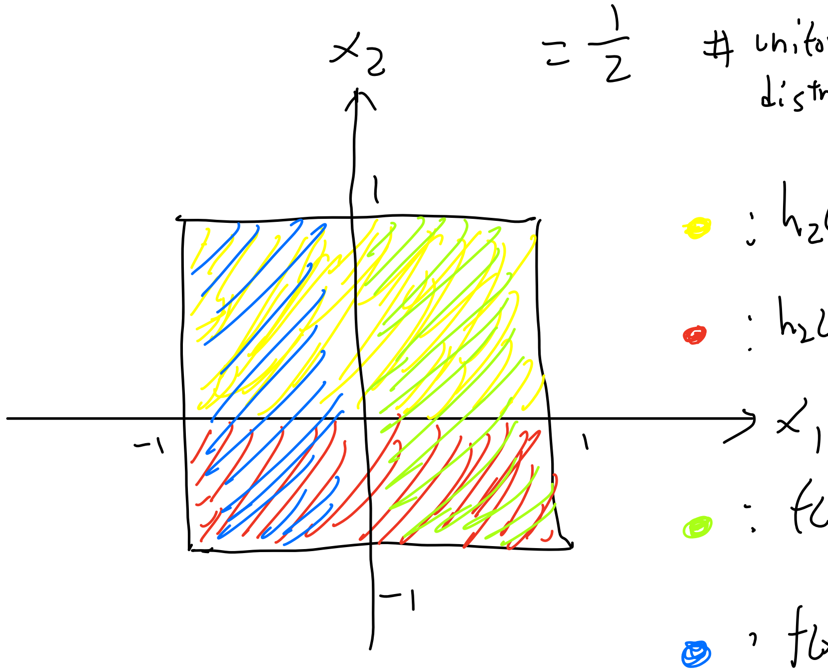
$\bullet$: $h_1(x) = -1$

$\bullet$: $f(x) = +1$

$\bullet$: $f(x) = -1$

$E_{out}(h_2) = (\bullet \cap \bullet) \cup (\bullet \cap \bullet)$

$= \frac{1}{2}$ # uniform distribution



$\bullet$: $h_2(x) = +1$

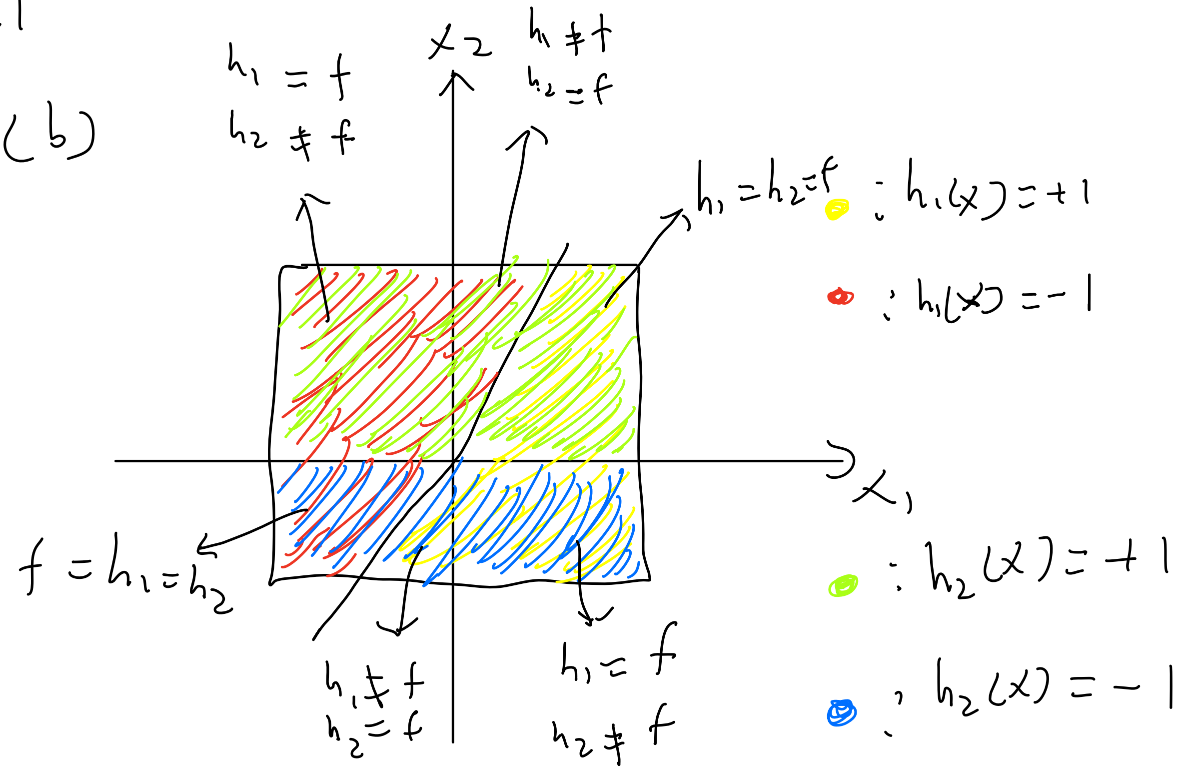
$\bullet$: $h_2(x) = -1$

$\bullet$: $f(x) = +1$

$\bullet$: $f(x) = -1$

11

(b)



$$E_{in}(h_2) = E_{in}(h_1)$$

$$0 : \left(\frac{3}{8}\right)^4$$

$$1 : C^4_2 C^2_1 \left(\frac{1}{8}\right)\left(\frac{4}{8}\right)\left(\frac{3}{8}\right)^2$$

$$2 : C^4_2 \left(\frac{1}{8}\right)^2 \left(\frac{4}{8}\right)^2$$

$$\frac{609}{4096}$$

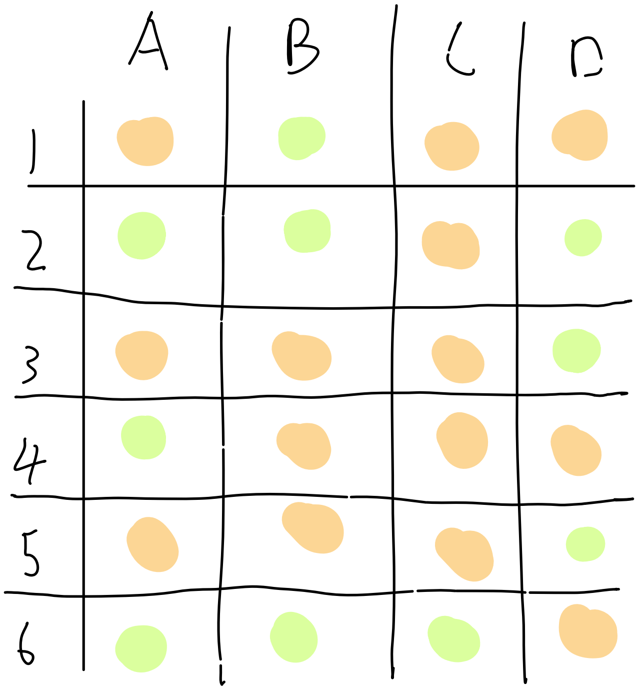
h_1	+	+	+	+
	o	o	o	o
h_2	+	+	+	+

h_1	+	−	+	+
	o	o	o	o
h_2	−	+	+	+

h_1	+	+	−	−
	o	o	o	o
h_2	−	−	+	+

12

(b)



$\{B\}$

$\{A, B, D\}$

$\{D\}$

$\{A\}$

$\{D\}$

(A, B, C)

$$\frac{4+300+150}{4^5}$$

1 : 4

2 : 5(5+10+10+5)

3 : 2(150)

• × × × ×
• • × × ×
• • • × ×
• • • • ×

• × △ × △
• × △ • △
• × △ • ×
• × △ △ △
• × △ × ×
• × △ • •

13 (b) 346.75

14 (e) 1886.36

15 (e) 6.9

16 (a) 533

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

hw1.py

Users > wujunming > Downloads > hw1.py

```python
import numpy as np
import random

def main():
    n = 100
    iter = 1000
    x_data = []
    y_data = []
    length = 0.0

    with open('hw1_train.dat.txt', 'r') as f:
        while True:
            line = f.readline()
            if line == '\n' or len(line) == 0:
                break
            x = [float(i) for i in line.split()[:-1]]     # 14題 x = [float(i)*2 for i in line.split()[:-1]]
            x.append(1.000000)                            # 16題 x.append(0.000000)
            x_data.append(np.array([x]))
            y_data.append(np.array([float(line.split()[-1])]))

    for i in range(iter):
        random.seed(i)
        w =np.zeros((1,11))
        condition = True
        safe_time = 0

        while condition:
            if safe_time >= 5*n:
                break
            ind = random.randint(0,n-1)
            if np.sign(np.dot(x_data[ind], w.T)) != y_data[ind]:    # 15題 if np.sign(np.dot(x_data[ind]/np.linalg.norm(x_data[ind]), w.T)) != y_data[ind]:
                w += y_data[ind]*x_data[ind]                        # 15題 w += y_data[ind]*(x_data[ind]/np.linalg.norm(x_data[ind]))
                safe_time = 0
            else:
                safe_time += 1

        length += np.linalg.norm(w)*np.linalg.norm(w)

    print(length/iter)

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

Restricted Mode ⊗ 0 ⚠ 0 Ln 1, Col 1 Spaces: 4 UTF-8 LF Python