Noise and Error

1. ideal mini-target

首先這裡我令

$$\mathrm{sign}(0) = 1$$

從題目給的式子可以知道

$$f_{0/1}(\mathbf{x}) = \underset{y\in\{-1,+1\}}{\mathrm{argmax}}\, P(y|\mathbf{x}) = \mathrm{sign}\left(P(+1|\mathbf{x}) - \frac{1}{2}\right)$$

等式右邊在檢查 +1 的機率有沒有大於或等於 $\frac{1}{2}$。會有這樣的等式我們可以從「期望錯誤」來看看，首先假設$P(+1|\mathbf{x}) = p$，p 介於 0 到 1 之間，此時如果我們得到一個 $\mathbf{x}$ ，去猜他的 y 是 $+1$ 跟 -1 的期望錯誤是：

$$\begin{cases} 1-p & 猜 y = +1 \\ p & 猜 y = -1 \end{cases}$$

所以如果我們希望猜 $y = -1$ 的期望錯誤大於等於猜 $y = +1$ 的期望錯誤，可以得到：

$$p \geq 1 - p \Rightarrow p \geq \frac{1}{2}$$

也就是說，只要$P(+1|\mathbf{x}) \geq \frac{1}{2}$，去猜 $y = +1$ 可以得到較小的期望錯誤，就可以得到上面公式中的：

$$\mathrm{sign}\left(P(+1|\mathbf{x}) - \frac{1}{2}\right)$$

對於 CIA 的 Error Function，題目希望 False Positive 比 False Negative 更重要 1000 倍，所以一樣假設 $P(+1|\mathbf{x}) = p$ ，p 介於 0 到 1 之間，此時如果我們得到一個 $\mathbf{x}$ ，去猜他的y是$+1$跟-1的期望錯誤是：

$$\begin{cases} 1000 \times (1-p) & 猜 y = +1 \\ p & 猜 y = -1 \end{cases}$$

因為 False Positive 比 False Negative 更重要 1000 倍，所以猜$y = +1$的期望錯誤就要多乘上那 1000 倍；最後跟上面一樣，我們希望猜$y = -1$的期望錯誤大於等於猜$y = +1$，可以得到：

$$p \geq 1000 \times (1-p)$$

$$\Rightarrow p \geq \frac{1000}{1001}$$

因此我們可以知道：

$$f_{CIA}(\mathbf{x}) = \mathrm{sign}\left(P(+1|\mathbf{x}) - \frac{1000}{1001}\right)$$

2. a noisy test environment

原本在無干擾(noise)的環境中：

$$E_{out}(g) = \underset{\mathbf{x}\sim P(\mathbf{x})}{\mathcal{E}} [\![g(\mathbf{x}) \neq f(\mathbf{x})]\!]$$

或者可以說：

$$Correct_{out}(g) = \underset{\mathbf{x}\sim P(\mathbf{x})}{\mathcal{E}} [\![g(\mathbf{x}) = f(\mathbf{x})]\!] = 1 - E_{out}(g)$$

此時換到有干擾的環境，有 ϵ 的機率會翻轉原本的結果，因此可以知道其錯誤就會變成：

$$\underbrace{E_{out}(g)(1-\epsilon)}_{\text{原本的部分只剩下}(1-\epsilon)\text{保持錯誤}} + \underbrace{\big(1 - E_{out}(g)\big)\epsilon}_{\text{原本對的有 }\epsilon\text{ 變成錯的}}$$

$$= E_{out}(g) - E_{out}(g)\epsilon + \epsilon - E_{out}(g)\epsilon = E_{out}(g) + \epsilon - 2\epsilon E_{out}(g)$$

● Linear Regression

3. $h(x) = wx$

$$\frac{1}{N}\sum_{n=1}^{N}(h(x_n) - y_n)^2 = \frac{1}{N}\sum_{n=1}^{N}(wx_n - y_n)^2$$

接著對 w 微分並找到等於 0 的值：

$$\frac{d}{dw}\left(\frac{1}{N}\sum_{n=1}^{N}(wx_n - y_n)^2\right) = \frac{1}{N}\sum_{n=1}^{N}2(wx_n - y_n)x_n = 0$$

$$\Rightarrow \sum_{n=1}^{N}wx_n^2 - \sum_{n=1}^{N}x_n y_n = 0 \Rightarrow w\sum_{n=1}^{N}x_n^2 = \sum_{n=1}^{N}x_n y_n$$

$$\Rightarrow w = \frac{\sum_{n=1}^{N}x_n y_n}{\sum_{n=1}^{N}x_n^2}$$

此時再對 w 微分：

$$\frac{d}{dw}\left(\frac{1}{N}\sum_{n=1}^{N}2(wx_n - y_n)x_n\right) = \frac{d}{dw}\left(\frac{1}{N}\sum_{n=1}^{N}2wx_n^2 - 2x_n y_n\right)$$

$$= \frac{1}{N}\sum_{n=1}^{N}2x_n^2$$

因為題目有確保 $\sum_{n=1}^{N}x_n^2$ 不等於 0，所以可以知道二階微分恆大於 0，

也就是說 $w = \frac{\sum_{n=1}^{N}x_n y_n}{\sum_{n=1}^{N}x_n^2}$ 確實是極小值。

4. optimal approximation

首先列出 Squared Error：

$$E^{SE} = (w_1 x + w_0 - a x^2 - b)^2$$
$$= w_1^2 x^2 + w_0^2 + a^2 x^4 + b^2$$
$$+2w_1 w_0 x - 2a w_1 x^3 - 2b w_1 x - 2a w_0 x^2 - 2b w_0 + 2ab x^2$$

接著取期望值：

$$\mathcal{E}[E^{SE}] = w_1^2 \mathcal{E}[x^2] + w_0^2 + a^2 \mathcal{E}[x^4] + b^2$$
$$+2w_1 w_0 \mathcal{E}[x] - 2a w_1 \mathcal{E}[x^3] - 2b w_1 \mathcal{E}[x] - 2a w_0 \mathcal{E}[x^2] - 2b w_0 + 2ab\mathcal{E}[x^2]$$

在偏微分之前，先列出期望值的值：

$$\mathcal{E}[x] = \frac{1}{2}, \mathcal{E}[x^2] = \frac{1}{3}, \mathcal{E}[x^3] = \frac{1}{4}, \mathcal{E}[x^4] = \frac{1}{5}$$

再帶入就可以化簡成：

$$\mathcal{E}[E^{SE}] = \tfrac{1}{3} w_1^2 + w_0^2 + \tfrac{1}{5} a^2 + b^2 + w_1 w_0 - \tfrac{1}{2} a w_1 - b w_1 - \tfrac{2}{3} a w_0 - 2b w_0 + \tfrac{2}{3} ab$$

$$= \frac{1}{5} a^2 + b^2 + \frac{2}{3} ab + \frac{1}{3} w_1^2 + w_0^2 + w_1 w_0 - w_1 \left(\frac{1}{2} a + b\right) - w_0 (\frac{2}{3} a + 2b)$$

接著就可以開始偏微分了：

$$\frac{\partial \mathcal{E}[E^{SE}]}{\partial w_1} = \frac{2}{3} w_1 + w_0 - \frac{1}{2} a - b = 0$$
$$\frac{\partial \mathcal{E}[E^{SE}]}{\partial w_0} = 2w_0 + w_1 - \frac{2}{3} a - 2\mathrm{b} = 0$$

接著開始解聯立方程組：

1. $\frac{2}{3} w_1 + w_0 = \frac{1}{2} a + b$
2. $2w_0 + w_1 = \frac{2}{3} a + 2b$
3. $\frac{2}{3} w_1 + \frac{4}{3} w_0 = \frac{4}{9} a + \frac{4}{3} b$ 第 2 條乘 $\frac{2}{3}$
4. $\frac{1}{3} w_0 = \frac{-1}{18} a + \frac{1}{3} b$ 第 3 條減第一條
5. $w_0 = b - \frac{1}{6} a$
6. $w_1 = a$

於是可以知道最佳的權重$w_1^* = a, w_0^* = b - \frac{1}{6} a$

5. output transformation

根據講義的內容：

$$E_{in}(\mathbf{w}_{\mathrm{LIN}}) = \frac{1}{N}\|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 = \frac{1}{N}(\mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{w} - 2\mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{y} + \mathbf{y}^{\mathrm{T}}\mathbf{y})$$

$$\nabla E_{in}(\mathbf{w}_{LIN}) = \frac{2}{N}(\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{w} - \mathbf{X}^{\mathrm{T}}\mathbf{y})$$

$$\mathbf{w}_{LIN} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$$

將 $y_n' = ay_n + b$ 代入式子，可以得到：$\mathbf{w}_{LIN}' = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}(a\mathbf{y} + \mathbf{b})$

此時我們先將$\mathbf{X}^{\mathrm{T}}(a\mathbf{y} + \mathbf{b})$細部放大來看：

$$\mathbf{X}^{\mathrm{T}} = \begin{bmatrix} x_{0,0} & x_{1,0} & \cdots & x_{n-1,0} \\ x_{0,1} & x_{1,1} & \cdots & x_{n-1,1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{0,d} & x_{1,d} & \cdots & x_{n-1,d} \end{bmatrix}, a\mathbf{y} + \mathbf{b} = \begin{bmatrix} ay_0 + b \\ ay_1 + b \\ \vdots \\ ay_{n-1} + b \end{bmatrix}$$

$x_{i,j}$ 代表 $\mathbf{x}_i$ 的第 j 個 element；為了方便觀察，我將 n 改成從 0 開始。

所以可以知道$\mathbf{X}^{\mathrm{T}}(a\mathbf{y} + \mathbf{b})$應該會等於：

$$\begin{bmatrix} x_{0,0} & x_{1,0} & \cdots & x_{n-1,0} \\ x_{0,1} & x_{1,1} & \cdots & x_{n-1,1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{0,d} & x_{1,d} & \cdots & x_{n-1,d} \end{bmatrix} \begin{bmatrix} ay_0 + b \\ ay_1 + b \\ \vdots \\ ay_{n-1} + b \end{bmatrix} = \begin{bmatrix} a\sum_{i=0}^{n-1} x_{i,0}y_i + b\sum_{i=0}^{n-1} x_{i,0} \\ a\sum_{i=0}^{n-1} x_{i,1}y_i + b\sum_{i=0}^{n-1} x_{i,1} \\ \vdots \\ a\sum_{i=0}^{n-1} x_{i,d}y_i + b\sum_{i=0}^{n-1} x_{i,d} \end{bmatrix}$$

此時我們把$(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}$移項到權重向量那邊：

$$\mathbf{X}^{\mathrm{T}}\mathbf{X}\,\mathbf{w}_{LIN}' = \mathbf{X}^{\mathrm{T}}(a\mathbf{y} + \mathbf{b})$$

這時我們再細部查看$\mathbf{X}^{\mathrm{T}}\mathbf{X}$：

$$\mathbf{X}^{\mathrm{T}}\mathbf{X} = \begin{bmatrix} x_{0,0} & x_{1,0} & \cdots & x_{n-1,0} \\ x_{0,1} & x_{1,1} & \cdots & x_{n-1,1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{0,d} & x_{1,d} & \cdots & x_{n-1,d} \end{bmatrix} \begin{bmatrix} x_{0,0} & x_{0,1} & \cdots & x_{0,d} \\ x_{1,0} & x_{1,1} & \cdots & x_{1,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n-1,0} & x_{n-1,1} & \cdots & x_{n-1,d} \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{i=0}^{n-1} x_{i,0}x_{i,0} & \sum_{i=0}^{n-1} x_{i,0}x_{i,1} & \cdots & \sum_{i=0}^{n-1} x_{i,0}x_{i,d} \\ \sum_{i=0}^{n-1} x_{i,1}x_{i,0} & \sum_{i=0}^{n-1} x_{i,1}x_{i,1} & \cdots & \sum_{i=0}^{n-1} x_{i,1}x_{i,d} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{n-1} x_{i,d}x_{i,0} & \sum_{i=0}^{n-1} x_{i,d}x_{i,1} & \cdots & \sum_{i=0}^{n-1} x_{i,d}x_{i,d} \end{bmatrix}$$

這時我們先回顧，在 y 轉換之前，從原本的解可以知道甚麼：

$$\mathbf{w}_{LIN} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} \Rightarrow \mathbf{X}^{\mathrm{T}}\mathbf{y} = \mathbf{X}^{\mathrm{T}}\mathbf{X}\,\mathbf{w}_{LIN}$$

$$\mathbf{X}^{\mathrm{T}}\mathbf{y} = \begin{bmatrix} \sum_{i=0}^{n-1} x_{i,0}y_i \\ \sum_{i=0}^{n-1} x_{i,1}y_i \\ \vdots \\ \sum_{i=0}^{n-1} x_{i,d}y_i \end{bmatrix} \qquad \mathbf{X}^{\mathrm{T}}\mathbf{X}\,\mathbf{w}_{LIN} = \begin{bmatrix} \sum_{i=0}^{n-1} x_{i,0}x_{i,0} & \sum_{i=0}^{n-1} x_{i,0}x_{i,1} & \cdots & \sum_{i=0}^{n-1} x_{i,0}x_{i,d} \\ \sum_{i=0}^{n-1} x_{i,1}x_{i,0} & \sum_{i=0}^{n-1} x_{i,1}x_{i,1} & \cdots & \sum_{i=0}^{n-1} x_{i,1}x_{i,d} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{n-1} x_{i,d}x_{i,0} & \sum_{i=0}^{n-1} x_{i,d}x_{i,1} & \cdots & \sum_{i=0}^{n-1} x_{i,d}x_{i,d} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix}$$

$$\mathbf{X}^{\mathrm{T}}\mathbf{X}\,\mathbf{w}_{LIN} = \begin{bmatrix} \sum_{i=0}^{d} w_i \sum_{j=0}^{n-1} x_{j,0}x_{j,i} \\ \sum_{i=0}^{d} w_i \sum_{j=0}^{n-1} x_{j,1}x_{j,i} \\ \vdots \\ \sum_{i=0}^{d} w_i \sum_{j=0}^{n-1} x_{j,d}x_{j,i} \end{bmatrix} = \mathbf{X}^{\mathrm{T}}\mathbf{y} = \begin{bmatrix} \sum_{i=0}^{n-1} x_{i,0}y_i \\ \sum_{i=0}^{n-1} x_{i,1}y_i \\ \vdots \\ \sum_{i=0}^{n-1} x_{i,d}y_i \end{bmatrix}$$

從上面的等式可以找到對應的項，所以這時再回顧我們需要的內容：

$$\mathbf{X}^{\mathrm{T}}\mathbf{X}\,\mathbf{w}_{LIN}' = \mathbf{X}^{\mathrm{T}}(a\mathbf{y}+\mathbf{b}) = \begin{bmatrix} a\sum_{i=0}^{n-1} x_{i,0}y_i + b\sum_{i=0}^{n-1} x_{i,0} \\ a\sum_{i=0}^{n-1} x_{i,1}y_i + b\sum_{i=0}^{n-1} x_{i,1} \\ \vdots \\ a\sum_{i=0}^{n-1} x_{i,d}y_i + b\sum_{i=0}^{n-1} x_{i,d} \end{bmatrix}$$

可以注意到，粉紅色可以拆成左右兩個部分，其中左邊剛好就是原本的 $\mathbf{X}^{\mathrm{T}}\mathbf{X}\,\mathbf{w}_{LIN}$再多乘上$a$倍，所以我們可以知道：

$$\mathbf{w}_{LIN}' = a\mathbf{w}_{LIN} + ?$$

那接下來就是要看怎麼湊出右半部分：$\begin{bmatrix} b\sum_{i=0}^{n-1} x_{i,0} \\ b\sum_{i=0}^{n-1} x_{i,1} \\ \vdots \\ b\sum_{i=0}^{n-1} x_{i,d} \end{bmatrix}$

這時我們可以回顧一個特性，$x_{i,0}$是我們幫$\mathbf{x_i}$補上的常數項，並且值是 1，所以這時候再回顧剛剛的$\mathbf{X}^{\mathrm{T}}\mathbf{X}$矩陣：

$$\begin{bmatrix} \sum_{i=0}^{n-1} x_{i,0}x_{i,0} & \sum_{i=0}^{n-1} x_{i,0}x_{i,1} & \cdots & \sum_{i=0}^{n-1} x_{i,0}x_{i,d} \\ \sum_{i=0}^{n-1} x_{i,1}x_{i,0} & \sum_{i=0}^{n-1} x_{i,1}x_{i,1} & \cdots & \sum_{i=0}^{n-1} x_{i,1}x_{i,d} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{n-1} x_{i,d}x_{i,0} & \sum_{i=0}^{n-1} x_{i,d}x_{i,1} & \cdots & \sum_{i=0}^{n-1} x_{i,d}x_{i,d} \end{bmatrix}$$

可以注意到其實第一個 column 右邊都是 1，所以可以化簡寫成：

$$\begin{bmatrix} \sum_{i=0}^{n-1} x_{i,0} & \sum_{i=0}^{n-1} x_{i,0}x_{i,1} & \cdots & \sum_{i=0}^{n-1} x_{i,0}x_{i,d} \\ \sum_{i=0}^{n-1} x_{i,1} & \sum_{i=0}^{n-1} x_{i,1}x_{i,1} & \cdots & \sum_{i=0}^{n-1} x_{i,1}x_{i,d} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{n-1} x_{i,d} & \sum_{i=0}^{n-1} x_{i,d}x_{i,1} & \cdots & \sum_{i=0}^{n-1} x_{i,d}x_{i,d} \end{bmatrix}$$

所以我們就只取第一個 column，就可以湊出我們想要的$\begin{bmatrix} b\sum_{i=0}^{n-1} x_{i,0} \\ b\sum_{i=0}^{n-1} x_{i,1} \\ \vdots \\ b\sum_{i=0}^{n-1} x_{i,d} \end{bmatrix}$：

$$\begin{bmatrix} \sum_{i=0}^{n-1} x_{i,0} & \sum_{i=0}^{n-1} x_{i,0}x_{i,1} & \cdots & \sum_{i=0}^{n-1} x_{i,0}x_{i,d} \\ \sum_{i=0}^{n-1} x_{i,1} & \sum_{i=0}^{n-1} x_{i,1}x_{i,1} & \cdots & \sum_{i=0}^{n-1} x_{i,1}x_{i,d} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{n-1} x_{i,d} & \sum_{i=0}^{n-1} x_{i,d}x_{i,1} & \cdots & \sum_{i=0}^{n-1} x_{i,d}x_{i,d} \end{bmatrix} \begin{bmatrix} b \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} b\sum_{i=0}^{n-1} x_{i,0} \\ b\sum_{i=0}^{n-1} x_{i,1} \\ \vdots \\ b\sum_{i=0}^{n-1} x_{i,d} \end{bmatrix}$$

也就是第一個值為 b，其他值為 0 的$(d+1)\times 1$的矩陣。

因此最終我們可以知道：

$$\mathbf{w}_{LIN}^{'} = a\mathbf{w}_{LIN} + \begin{bmatrix} b \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

● More on Linear Models

6. Hessian Matrix

$$E_{in}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} \ln\Big(1 + \exp(-y_n \mathbf{w}^{\mathbf{T}}\mathbf{x}_{\mathbf{n}})\Big)$$

如果只求一次微分可以知道：

$$\frac{\partial}{\partial w_i} E_{in}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} \frac{-y_n x_{n,i}}{1 + \exp(y_n \mathbf{w}^{\mathbf{T}}\mathbf{x}_{\mathbf{n}})}$$

接著使用微積分把海森矩陣展開：

$\mathrm{Hessian}\big(E_{in}(\mathbf{w})\big) =$

$$\begin{bmatrix} \frac{1}{N}\sum_{n=1}^{N} \frac{\exp(y_n \mathbf{w}^{\mathbf{T}}\mathbf{x}_{\mathbf{n}})\big(y_n^2 x_{n,0} x_{n,0}\big)}{[1+\exp(y_n \mathbf{w}^{\mathbf{T}}\mathbf{x}_{\mathbf{n}})]^2} & \cdots & \frac{1}{N}\sum_{n=1}^{N} \frac{\exp(y_n \mathbf{w}^{\mathbf{T}}\mathbf{x}_{\mathbf{n}})\big(y_n^2 x_{n,d} x_{n,0}\big)}{[1+\exp(y_n \mathbf{w}^{\mathbf{T}}\mathbf{x}_{\mathbf{n}})]^2} \\ \vdots & \ddots & \vdots \\ \frac{1}{N}\sum_{n=1}^{N} \frac{\exp(y_n \mathbf{w}^{\mathbf{T}}\mathbf{x}_{\mathbf{n}})\big(y_n^2 x_{n,0} x_{n,d}\big)}{[1+\exp(y_n \mathbf{w}^{\mathbf{T}}\mathbf{x}_{\mathbf{n}})]^2} & \cdots & \frac{1}{N}\sum_{n=1}^{N} \frac{\exp(y_n \mathbf{w}^{\mathbf{T}}\mathbf{x}_{\mathbf{n}})\big(y_n^2 x_{n,d} x_{n,d}\big)}{[1+\exp(y_n \mathbf{w}^{\mathbf{T}}\mathbf{x}_{\mathbf{n}})]^2} \end{bmatrix}$$

使用 Logistic Function 的特性、$y_i^2 = 1$，以及將 Σ 的部分改成用矩陣乘法來表示，把矩陣化簡只有對角線有值的對角矩陣 **D**：

$$\begin{bmatrix} \theta(y_1 \mathbf{w}^{\mathbf{T}}\mathbf{x}_{\mathbf{1}})\theta(-y_1 \mathbf{w}^{\mathbf{T}}\mathbf{x}_{\mathbf{1}}) & \cdots & 0 \\ \vdots & \theta(y_i \mathbf{w}^{\mathbf{T}}\mathbf{x}_i)\theta(-y_i \mathbf{w}^{\mathbf{T}}\mathbf{x}_i) & \vdots \\ 0 & \cdots & \theta(y_N \mathbf{w}^{\mathbf{T}}\mathbf{x}_N)\theta(-y_N \mathbf{w}^{\mathbf{T}}\mathbf{x}_N) \end{bmatrix}$$

$\frac{1}{N}$ 可以拿掉的原因是在推導牛頓法的時候，會有一個步驟是同時移項，此時$\frac{1}{N}$就會可以移除掉。可以注意到$\theta(y_i \mathbf{w}^{\mathbf{T}}\mathbf{x}_i)\theta(-y_i \mathbf{w}^{\mathbf{T}}\mathbf{x}_i)$不管 y_i 是$+1$還是-1，$\theta(\mathbf{w}^{\mathbf{T}}\mathbf{x}_i)$ 跟 $\theta(-\mathbf{w}^{\mathbf{T}}\mathbf{x}_i)$ 兩個都一定會出現，所以可以把 y_i 省略掉；接著換成題目所要的形式：

$$\mathbf{D} = \begin{bmatrix} h_t(\mathbf{x}_1)h_t(-\mathbf{x}_1) & \cdots & 0 \\ \vdots & h_t(\mathbf{x}_i)h_t(-\mathbf{x}_i) & \vdots \\ 0 & \cdots & h_t(\mathbf{x}_N)h_t(-\mathbf{x}_N) \end{bmatrix}$$

而位於左右兩邊的 $\mathbf{X}^T$跟 $\mathbf{X}$ ：

$$\mathbf{X}^T = \begin{bmatrix} x_{\mathbf{1,0}} & \cdots & x_{\mathrm{N},\mathbf{0}} \\ \vdots & \ddots & \vdots \\ x_{\mathbf{1,d}} & \cdots & x_{\mathrm{N},\mathbf{d}} \end{bmatrix} = \begin{bmatrix} \vdots & \cdots & \vdots \\ \mathbf{x}_{\mathbf{1}} & \cdots & \mathbf{x}_{\mathbf{N}} \\ \vdots & \cdots & \vdots \end{bmatrix}$$

7. truncated squared loss

$$\nabla \mathrm{err}(\mathbf{w}^{\mathbf{T}}\mathbf{x}, y) = \begin{cases} 0 & 1 \le y\mathbf{w}^{\boldsymbol{T}}\mathbf{x} \\ 2(y\mathbf{w}^{\mathbf{T}}\mathbf{x} - 1)\mathbf{xy} & 1 > y\mathbf{w}^{\mathbf{T}}\mathbf{x} \end{cases}$$

所以可以知道 truncated squared loss 的 SGD 長的如下：

$$\begin{cases} \mathbf{w}_{\boldsymbol{t+1}} \leftarrow \mathbf{w}_{\boldsymbol{t}} + \eta \times 0 \times (\mathbf{x}_{\boldsymbol{n}}\boldsymbol{y}_{\boldsymbol{n}}) & 1 \le y\mathbf{w}^{\boldsymbol{T}}\mathbf{x} \\ \mathbf{w}_{\boldsymbol{t+1}} \leftarrow \mathbf{w}_{\boldsymbol{t}} + \boldsymbol{\eta} \times\ 2(1 - y_n\mathbf{w}_{\boldsymbol{t}}^{\mathbf{T}}\mathbf{x}_{\boldsymbol{n}})(\mathbf{x}_{\boldsymbol{n}}\boldsymbol{y}_{\boldsymbol{n}}) & 1 > y\mathbf{w}^{\mathbf{T}}\mathbf{x} \end{cases}$$

而原版的 PLA 是：

$$\mathbf{w}_{\boldsymbol{t+1}} \leftarrow \mathbf{w}_{\boldsymbol{t}} + 1 \times\ [\![\boldsymbol{y}_{\boldsymbol{n}} \neq \mathrm{sign}(\mathbf{w}_{\boldsymbol{t}}^{\mathbf{T}}\mathbf{x}_{\boldsymbol{n}})]\!](\mathbf{x}_{\boldsymbol{n}}\boldsymbol{y}_{\boldsymbol{n}})$$

不同之處：

- truncated squared loss 的 SGD 是 $y\mathbf{w}^{\boldsymbol{T}}\mathbf{x} \ge 1$ 才不會更新，但是 PLA 只要 $y\mathbf{w}^{\boldsymbol{T}}\mathbf{x} \ge 0$ 就不會更新。
- truncated squared loss 的 SGD 只要$y\mathbf{w}^{\boldsymbol{T}}\mathbf{x} < 1$就會開始更新但是 PLA 要等到 $y\mathbf{w}^{\boldsymbol{T}}\mathbf{x} < 0$ 才開始更新。
- truncated squared loss 的 SGD 只要$y\mathbf{w}^{\boldsymbol{T}}\mathbf{x}$ 越小，$\boldsymbol{\eta} \times\ 2(1 - y_n\mathbf{w}_{\boldsymbol{t}}^{\mathbf{T}}\mathbf{x}_{\boldsymbol{n}})$的部分就越大，但是 PLA 都固定是 1。

相同之處：

- 只要 $y\mathbf{w}^{\boldsymbol{T}}\mathbf{x} \ge 1$ ，truncated squared loss 的 SGD 跟 PLA 就都不會更新，不像 square error 或 cross entropy 會更新。

● Multinomial Logistic Regression

8. Multinomial Logistic Regression

$$E_{in}(W)=\frac{1}{N}\sum_{n=1}^{N}\left(-\sum_{k=1}^{K}\left([\![y_n=k]\!]\ln\frac{\exp(\mathbf{w}_k^T\mathbf{x}_n)}{\sum_{i=1}^{K}\exp(\mathbf{w}_i^T\mathbf{x}_n)}\right)\right)$$

先將 E_{in} 化簡：

$$E_{in}(W)=\frac{1}{N}\sum_{n=1}^{N}\left(ln\left(\sum_{i=1}^{K}\exp(\mathbf{w}_i^T\mathbf{x}_n)\right)-\mathbf{w}_{y_n}^T\mathbf{x}_n\right)$$

然後我們先只對第 i 個權重做偏微分：

$$\frac{\partial}{\partial\mathbf{w_i}}E_{in}(W)=\frac{1}{N}\sum_{n=1}^{N}\left(\frac{\exp(\mathbf{w}_i^T\mathbf{x}_n)\mathbf{x}_n}{\sum_{i=1}^{K}\exp(\mathbf{w}_i^T\mathbf{x}_n)}-[\![y_n=i]\!]\mathbf{x}_n\right)$$

$$=\frac{1}{N}\sum_{n=1}^{N}(h_i(\mathbf{x}_n)-[\![y_n=i]\!])\,\mathbf{x}_n$$

所以可以知道：

$$\begin{bmatrix}\frac{\partial}{\partial\mathbf{w}_1}\\ \vdots\\ \frac{\partial}{\partial\mathbf{w}_K}\end{bmatrix}=\begin{bmatrix}\frac{1}{N}\sum_{n=1}^{N}(h_1(\mathbf{x}_n)-[\![y_n=1]\!])\,\mathbf{x}_n\\ \vdots\\ \frac{1}{N}\sum_{n=1}^{N}(h_K(\mathbf{x}_n)-[\![y_n=K]\!])\,\mathbf{x}_n\end{bmatrix}$$

上面的矩陣是 K 個 row，$d+1$個 column，由於題目要求的$\nabla E_{in}(W)$是要轉 90 度，所以我們給他轉 90 度：

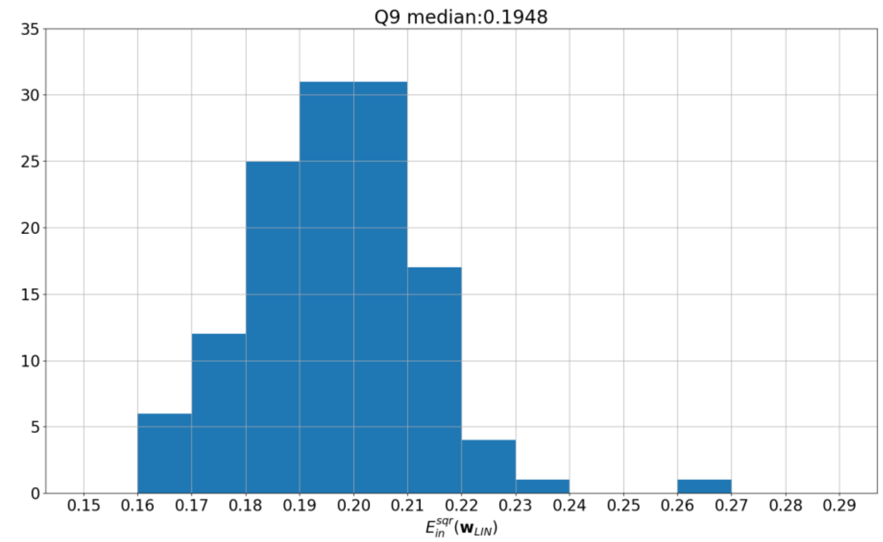
$$\nabla E_{in}(W)=\begin{bmatrix}\vdots & \cdots & \vdots\\ \frac{\partial}{\partial\mathbf{w}_1} & \cdots & \frac{\partial}{\partial\mathbf{w}_K}\\ \vdots & \cdots & \vdots\end{bmatrix}=$$

$$\begin{pmatrix}\vdots & \cdots & \vdots\\ \frac{1}{N}\sum_{n=1}^{N}\left(h_1(\mathbf{x}_n)-[\![y_n=1]\!]\right)\mathbf{x}_n & \cdots & \frac{1}{N}\sum_{n=1}^{N}\left(h_K(\mathbf{x}_n)-[\![y_n=K]\!]\right)\mathbf{x}_n\\ \vdots & \cdots & \vdots\end{pmatrix}$$

這樣一來$\nabla E_{in}(W)$的維度就是$(d+1)\times K$，跟 W 一樣了。

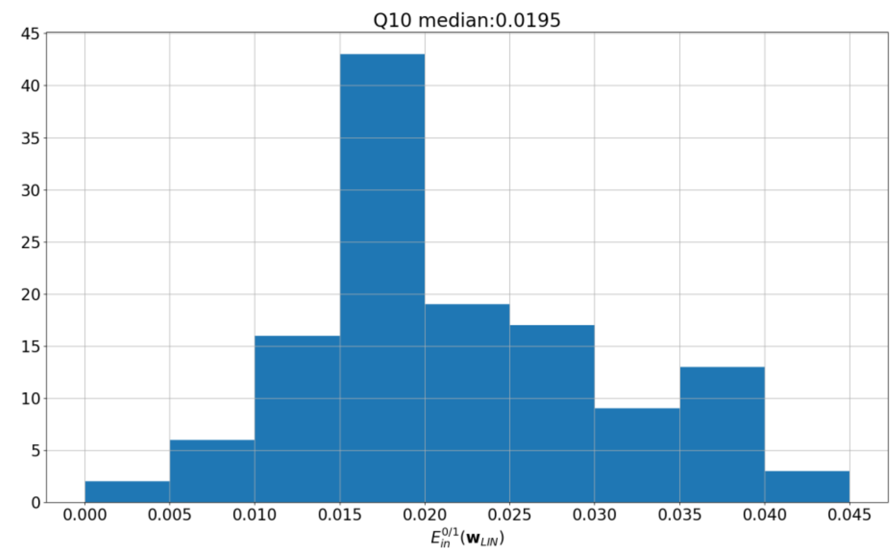
● Multinomial Logistic Regression

9. linear regression with square error



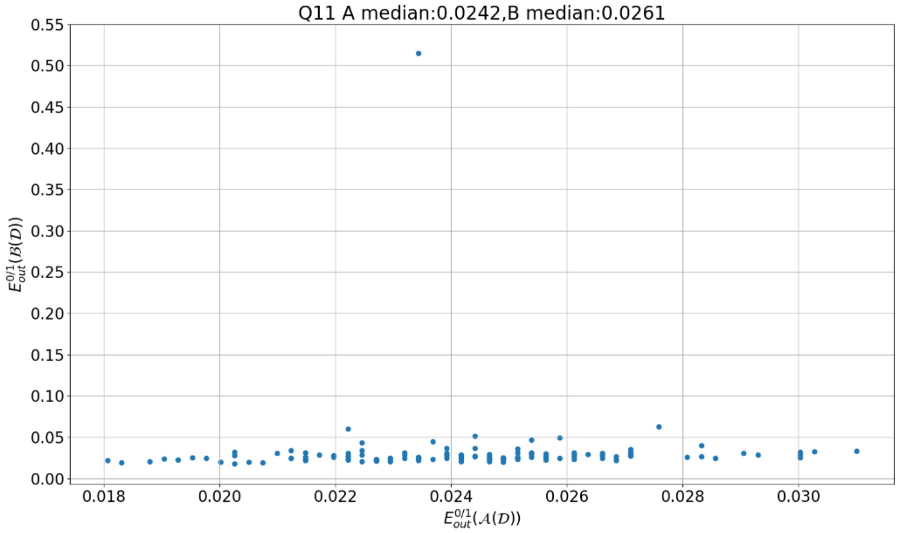
$E_{in}^{sqr}(\mathbf{w}_{LIN})$中位數是 0.1948。

10. linear regression with 0/1 error



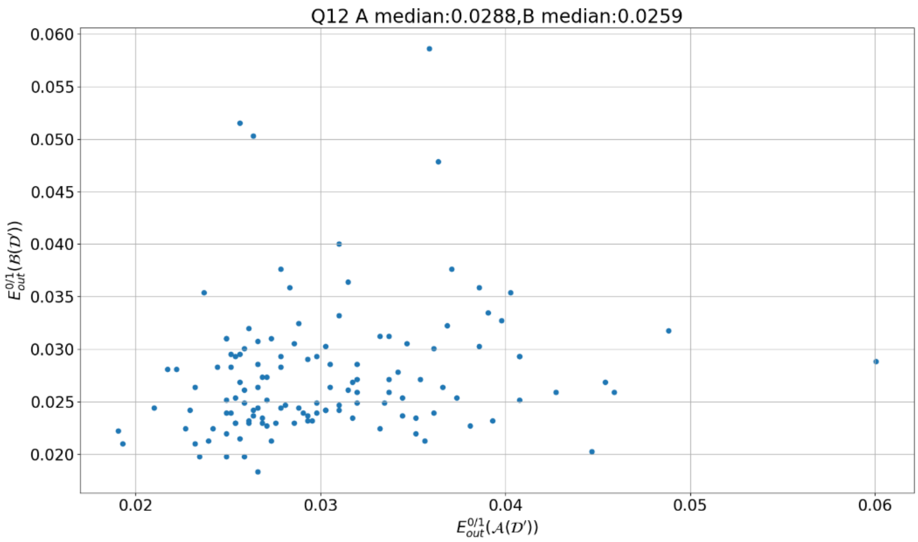
$E_{in}^{0/1}(\mathbf{w}_{LIN})$中位數是 0.0195。

11. compare LIN and LOG



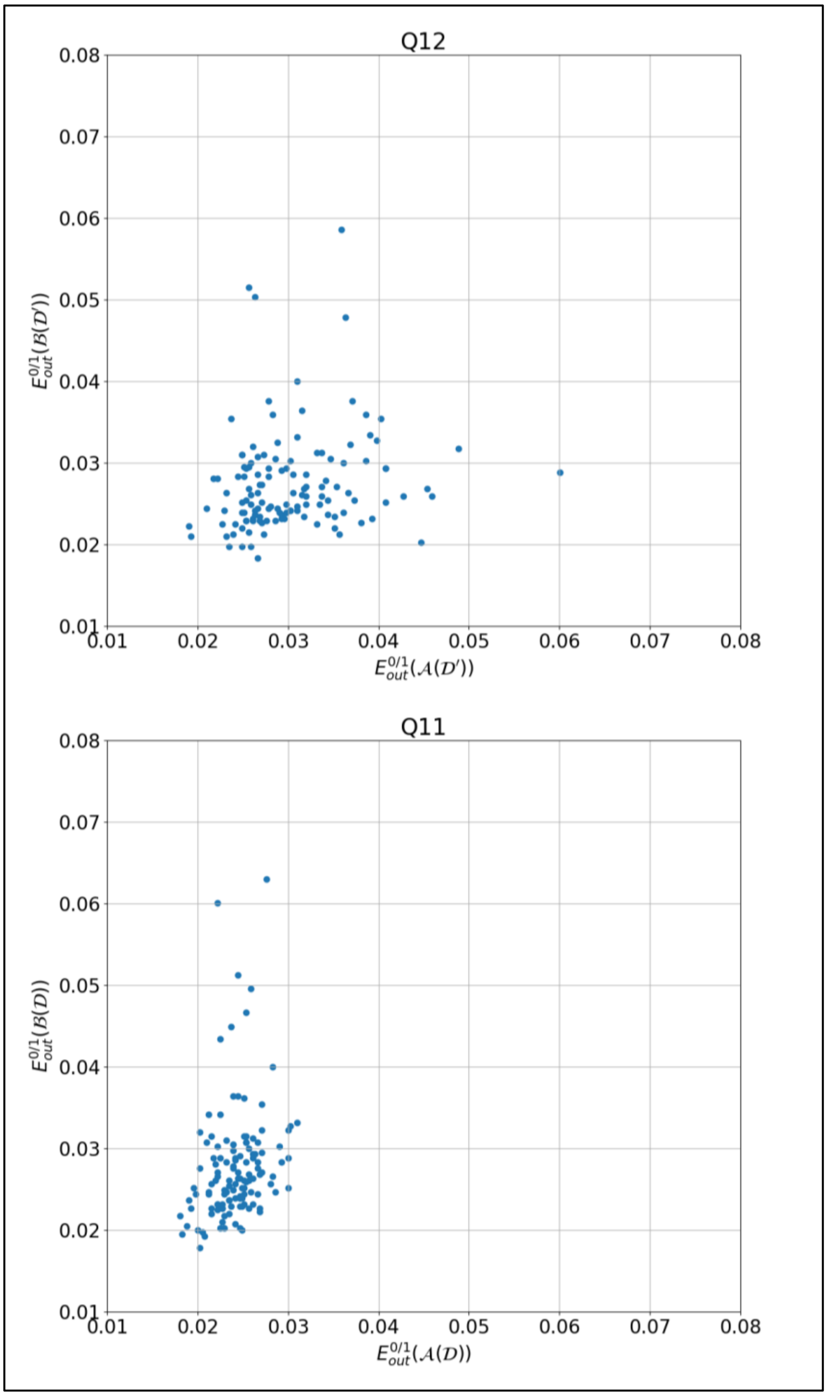
$E_{out}^{0/1}(\mathcal{A}(\mathcal{D}))$跟$E_{out}^{0/1}(\mathcal{B}(\mathcal{D}))$的中位數分別是 0.0242，0.0261

12. compare LIN and LOG with outlier



$E_{out}^{0/1}(\mathcal{A}(\mathcal{D}'))$跟$E_{out}^{0/1}(\mathcal{B}(\mathcal{D}'))$的中位數分別是 0.0288，0.0259

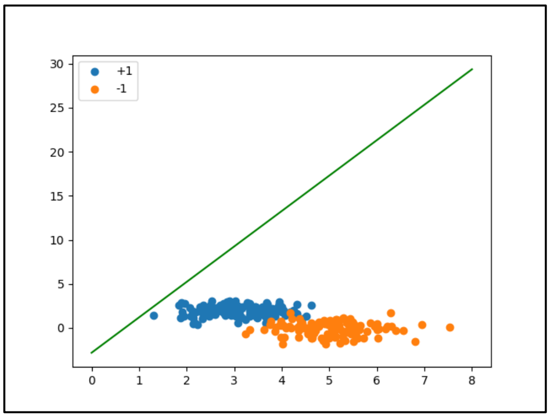
上面的圖因為 outlier 導致圖片變形很嚴重，所以我限定了範圍後得到兩軸比例一樣的圖：

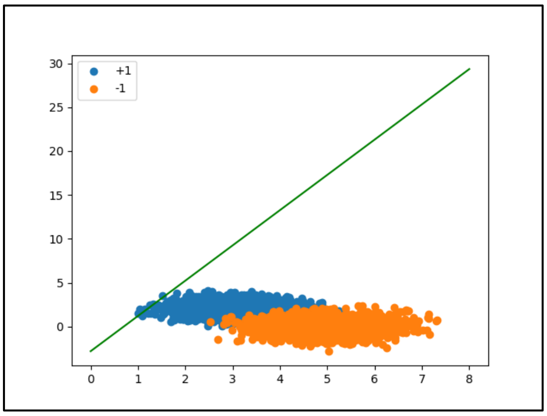


可以看到在一定的範圍內，其實兩者是相當接近的，或者說大致坐落於 $y = x$ 這條線附近，這也跟中位數相近的結果相符合。

- logistic regression 在加入 outlier 前就已經有許多點的錯誤率容易比 linear regression 來的大了，在 11 題的圖中還可以看到有一個點錯誤率比 0.5 還高，個人暫時推測應該是尚未找到最佳權重。
- 可以看到加入 outlier 後，linear regression 的錯誤率也跟著提高，顯示出 outlier 有影響到最佳權重的值。

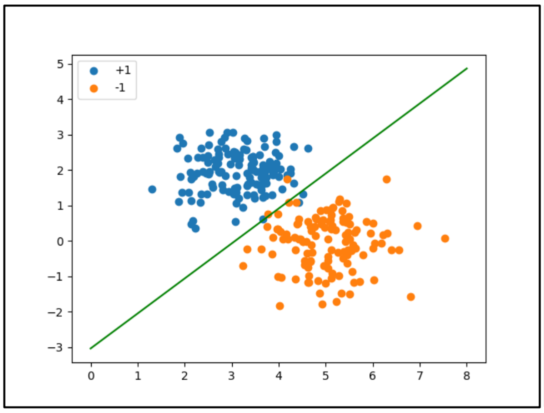
為了印證第一點，我先從尚未加入 outlier 的資料找出那個錯誤率大於 0.5 的資料出來，發現他的圖長的如下：

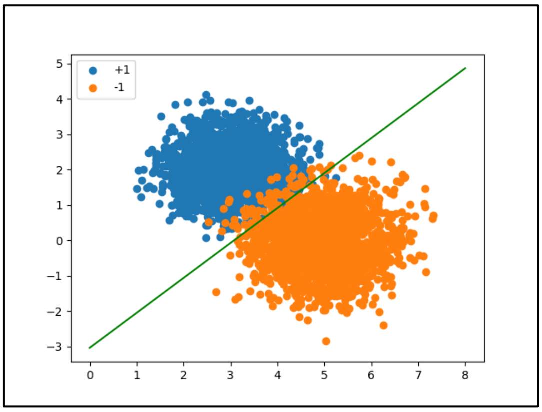




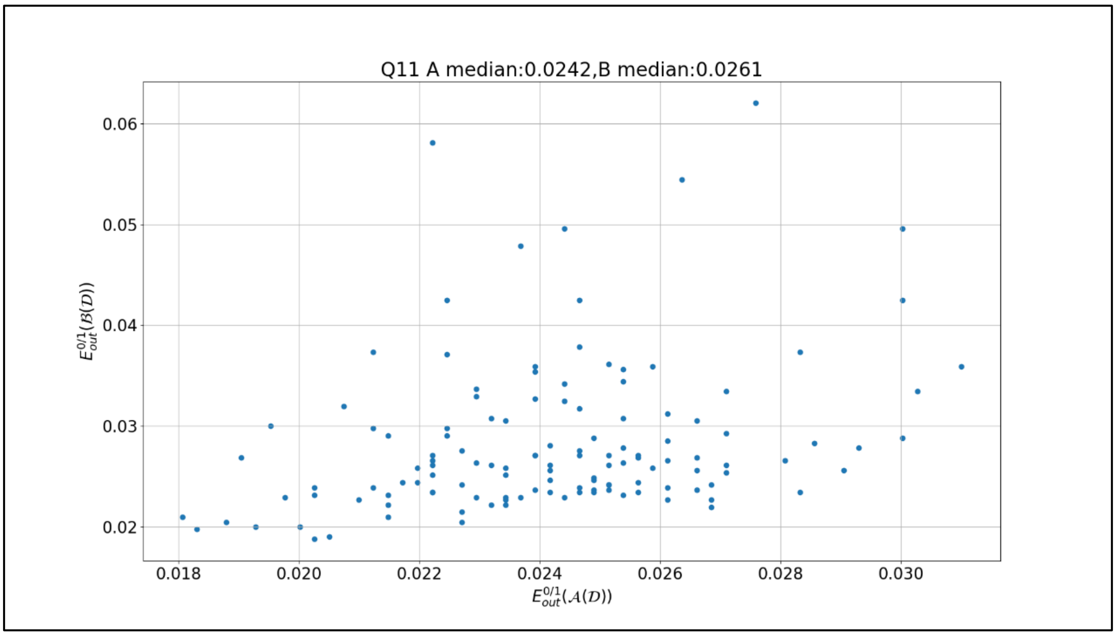
左邊是訓練資料，右邊是測試資料。

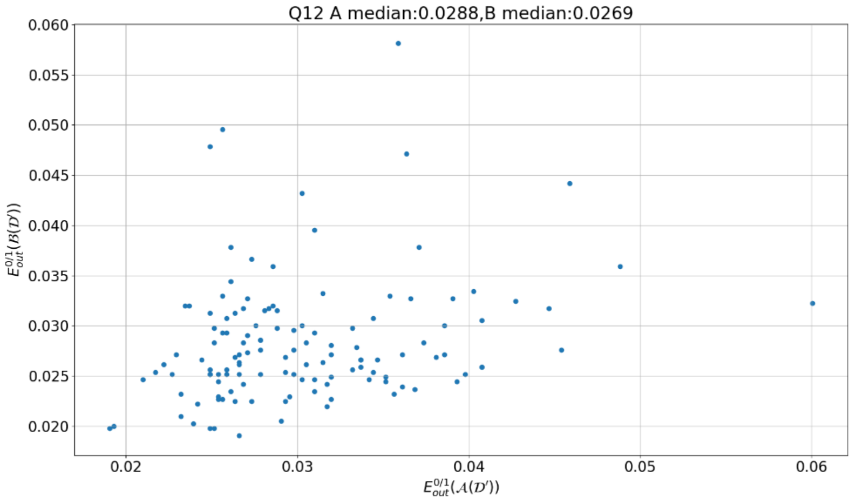
我覺得這應該是尚未找到最佳的權重所導致，於是我將訓練的次數從 500 調整到 5000，$\eta = 0.1$保持不變，訓練跟測試的結果分別如下：



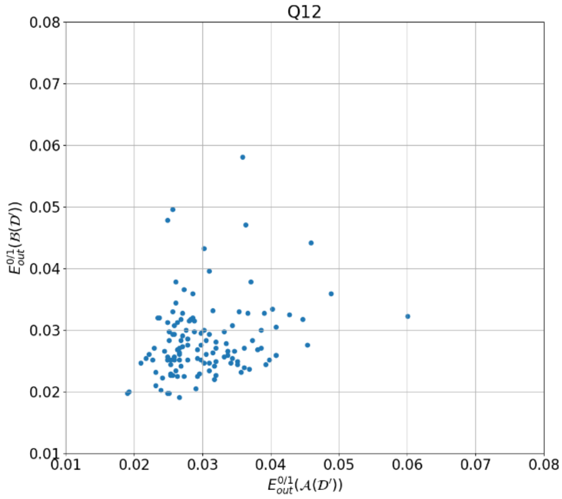


確實印證了我的猜測。所以我重新做了第 11 題跟第 12 題的模擬，將訓練次數改成 5000 次，得到如下的圖形：

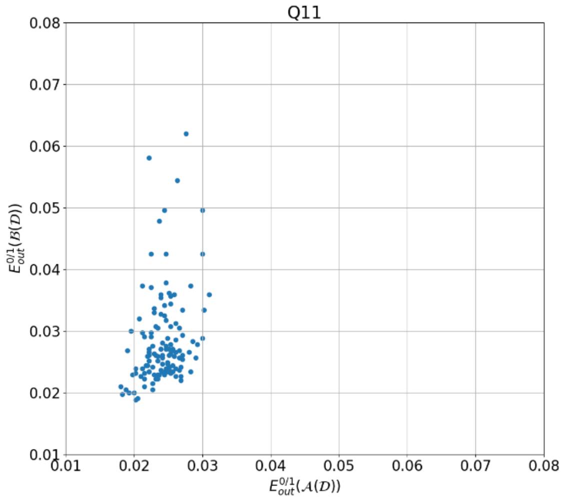




Q12 A median:0.0288,B median:0.0269
0.060
0.055
0.050
0.045
0.040
0.035
0.030
0.025
0.020
$E_{out}^{0/1}(\mathcal{B}(\mathcal{D}'))$
0.02
0.03
0.04
0.05
0.06
$E_{out}^{0/1}(\mathcal{A}(\mathcal{D}'))$

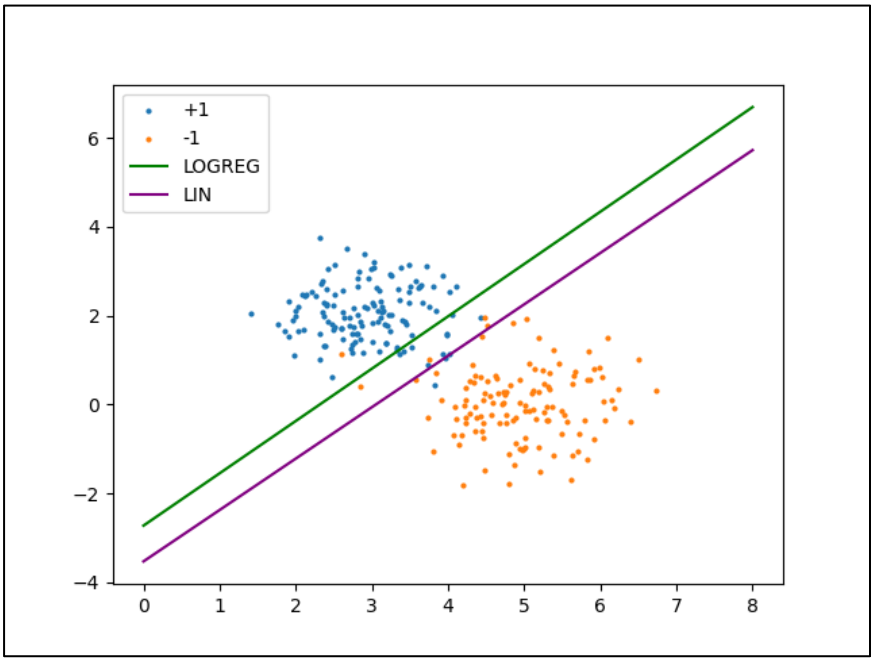


Q12
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
$E_{out}^{0/1}(\mathcal{B}(\mathcal{D}'))$
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
$E_{out}^{0/1}(\mathcal{A}(\mathcal{D}'))$

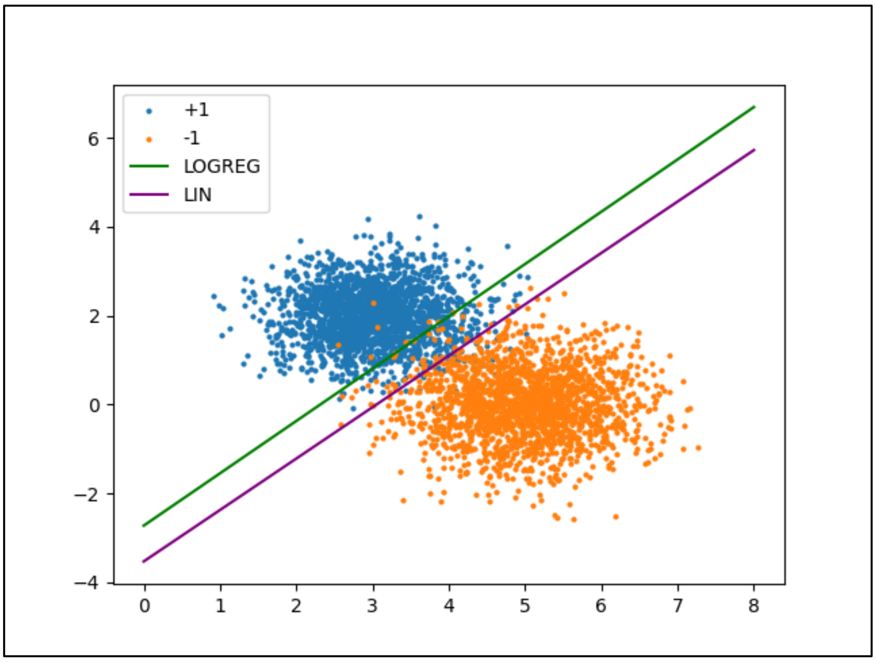


Q11
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
$E_{out}^{0/1}(\mathcal{B}(\mathcal{D}))$
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
$E_{out}^{0/1}(\mathcal{A}(\mathcal{D}))$

但是依舊發現 logistic regression 在加入 outlier 前就已經有許多點的錯誤率容易比 linear regression 來的大了，所以應該只是剛好那個錯誤率大於 0.5 的點沒有學好而已。所以接著我再從尚未加入 outlier 的資料找出一個錯誤率大於 0.05 的資料出來，發現他的圖長的如下：



可以發現在訓練資料中，綠色線雖然跟紫色線有點距離，但是他好像也能不錯的分開兩種點。這時再看測試資料：



會發現綠色線雖然在訓練資料中看起來還不錯，但是到了測試資料就有過多的點判斷錯誤了。

也就是說 logistic regression 在加入 outlier 前的錯誤率比較高，是訓練的結果，而不是訓練不夠。

那現在問題就變成為何 logistic regression 在訓練資料中會得到跟 linear regression 如此不同的權重。我認為主要是 E_{in} 的不同所導致，logistic regression 尋求的是最大化 Likelihood，或者說要最小化：

$$\frac{1}{N}\sum_{n=1}^{N} -ln\Big(\theta(y_n\mathbf{w}^{\mathbf{T}}\mathbf{x}_n)\Big)$$

其中的 $y_n\mathbf{w}^{\mathbf{T}}\mathbf{x}_n$ 是未標準化的與超平面的距離，這個距離會再代入 logistic function 投射到 0 到 1 的範圍，再取 log，全部加起來取負號。此時先來看 linear regression 要最小化的 E_{in}：

$$\frac{1}{N}\sum_{n=1}^{N}(\mathbf{w}^{\mathbf{T}}\mathbf{x}_n - y_n)^2$$

這裡我讓括弧內上下同乘y_n：

$$\frac{1}{N}\sum_{n=1}^{N}\left(\frac{y_n\mathbf{w}^{\mathbf{T}}\mathbf{x}_n - y_n^2}{y_n}\right)^2 = \frac{1}{N}\sum_{n=1}^{N}(y_n\mathbf{w}^{\mathbf{T}}\mathbf{x}_{\mathbf{n}} - 1)^2$$

一樣可以看到「未標準化的與超平面的距離」的身影

- logistic regression 會去最大化$\frac{1}{N}\sum_{n=1}^{N} ln\Big(\theta(y_n\mathbf{w}^{\mathbf{T}}\mathbf{x}_n)\Big)$，也就是說他除了要求點要分對($\theta(y_n\mathbf{w}^{\mathbf{T}}\mathbf{x}_n)$的值大於 0.5)，也會傾向於讓$\theta(y_n\mathbf{w}^{\mathbf{T}}\mathbf{x}_n)$值越大越好，換句話說就是比較喜歡讓分對的點離超平面更遠一些。
- linear regression 則是追求各個點到超平面之間的距離減 1 後的平方總和越小越好。

對於同樣的一組訓練資料，logistic regression 比較偏好讓分對的點離超平面更遠一些的權重，而 linear regression 則是偏好讓分對的點離超平面(到某個位置)更近一些的權重；但是由於今天資料產生的方式是選取兩個中心點，以該中心向外輻射的常態分布，而 logistic regression 的選取方式使他容易受訓練資料的分佈影響，因而偏向其中一個中心，所以才會在測試資料中有較差的表現。

- Bonus

13. internal linear regression

令第 6 題的 $\mathbf{D}$ 矩陣中的$y_i\mathbf{w}^\mathbf{T}\mathbf{x}_i = \boldsymbol{z}_i$，可以得到：

$$\mathbf{D} = \begin{bmatrix} \theta(\boldsymbol{z}_1)\theta(-\boldsymbol{z}_1) & \cdots & 0 \\ \vdots & \theta(\boldsymbol{z}_i)\theta(-\boldsymbol{z}_i) & \vdots \\ 0 & \cdots & \theta(\boldsymbol{z}_N)\theta(-\boldsymbol{z}_N) \end{bmatrix}$$

由於要將 $\mathbf{X}^T\mathbf{D}\mathbf{X}$ 轉成 $\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}}$ 的形式，加上 $\mathbf{D}$ 又是正定對角矩陣，所以我們可以直接給 $\mathbf{D}$ 開根號：

$$\mathbf{D}^{\frac{1}{2}} = \begin{bmatrix} \sqrt{\theta(\boldsymbol{z}_1)\theta(-\boldsymbol{z}_1)} & \cdots & 0 \\ \vdots & \sqrt{\theta(\boldsymbol{z}_i)\theta(-\boldsymbol{z}_i)} & \vdots \\ 0 & \cdots & \sqrt{\theta(\boldsymbol{z}_N)\theta(-\boldsymbol{z}_N)} \end{bmatrix}$$

所以可以知道

$$\widetilde{\mathbf{X}}^T = \mathbf{X}^T\mathbf{D}^{\frac{1}{2}} = \begin{bmatrix} \vdots & \cdots & \vdots \\ \mathbf{x}_1\sqrt{\theta(\boldsymbol{z}_1)\theta(-\boldsymbol{z}_1)} & \cdots & \mathbf{x}_N\sqrt{\theta(\boldsymbol{z}_N)\theta(-\boldsymbol{z}_N)} \\ \vdots & \cdots & \vdots \end{bmatrix}$$

接著處理$-\nabla \mathrm{E}_{\mathrm{in}}(\mathbf{w}_\mathrm{t})$，他應該要長得像下面兩個矩陣乘積後的結果：

$$\begin{bmatrix} \vdots & \cdots & \vdots \\ \mathbf{x}_1 & \cdots & \mathbf{x}_N \\ \vdots & \cdots & \vdots \end{bmatrix}\begin{bmatrix} \theta(-\boldsymbol{z}_1)y_1 \\ \vdots \\ \theta(-\boldsymbol{z}_N)y_N \end{bmatrix}$$

$\frac{1}{N}$ 拿掉的理由一樣是因為移項的時候可以消掉。

而$-\nabla E_{in}(\mathbf{w}_t)$所對應的形式為$\widetilde{\mathbf{X}}^T\tilde{\mathbf{y}}$，所以我們可以知道$\widetilde{\mathbf{X}}^T\tilde{\mathbf{y}}$這兩個矩陣的乘積應該要長的像下面這樣：

$$\begin{bmatrix} \vdots & \cdots & \vdots \\ \mathbf{x}_1\sqrt{\theta(\boldsymbol{z}_1)\theta(-\boldsymbol{z}_1)} & \cdots & \mathbf{x}_N\sqrt{\theta(\boldsymbol{z}_N)\theta(-\boldsymbol{z}_N)} \\ \vdots & \cdots & \vdots \end{bmatrix}\begin{bmatrix} \frac{\theta(-\boldsymbol{z}_1)y_1}{\sqrt{\theta(\boldsymbol{z}_1)\theta(-\boldsymbol{z}_1)}} \\ \vdots \\ \frac{\theta(-\boldsymbol{z}_N)y_N}{\sqrt{\theta(\boldsymbol{z}_N)\theta(-\boldsymbol{z}_N)}} \end{bmatrix} = \begin{bmatrix} \vdots & \cdots & \vdots \\ \mathbf{x}_1 & \cdots & \mathbf{x}_N \\ \vdots & \cdots & \vdots \end{bmatrix}\begin{bmatrix} \theta(-\boldsymbol{z}_1)y_1 \\ \vdots \\ \theta(-\boldsymbol{z}_N)y_N \end{bmatrix}$$

所以可以知道 $\widetilde{\mathbf{y}}$ 的長相：

$$\tilde{\mathbf{y}} = \begin{bmatrix} \frac{\theta(-\boldsymbol{z}_1)y_1}{\sqrt{\theta(\boldsymbol{z}_1)\theta(-\boldsymbol{z}_1)}} \\ \vdots \\ \frac{\theta(-\boldsymbol{z}_N)y_1}{\sqrt{\theta(\boldsymbol{z}_N)\theta(-\boldsymbol{z}_N)}} \end{bmatrix} = \begin{bmatrix} y_1\sqrt{\frac{\theta(-\boldsymbol{z}_1)}{\theta(\boldsymbol{z}_1)}} \\ \vdots \\ y_N\sqrt{\frac{\theta(-\boldsymbol{z}_N)}{\theta(\boldsymbol{z}_N)}} \end{bmatrix}$$