● Beyond Binary Linear Classification

1. OVO time complexity

總共有 K 個類別，因此 OVO 的組別總共有$C\binom{K}{2}=\frac{K(K-1)}{2}$種，而每種訓練的時間為$a\left(\frac{2N}{K}\right)^3$因此總共的時間為：

$$a\left(\frac{2N}{K}\right)^3\times\frac{K(K-1)}{2}=\frac{4a(K-1)N^3}{K^2}$$

相比於 OVA 的aKN^3，可以看出當$K=10$時 OVA 需要$a10N^3$，但是 OVO 只要$a\frac{9}{25}N^3$。

2. Q-dimensional polynomial transform

如果我們將原本的 1 維資料轉換成 N 維資料，並且形式如同 Vandermonde matrix 的每一項，也就是：

$$\boldsymbol{\Phi}(\boldsymbol{x_i})=\left[\mathbf{1},\boldsymbol{x_i},\boldsymbol{x_i^2},\boldsymbol{x_i^3},\dots,\boldsymbol{x_i^{N-1}}\right]=\mathbf{z_i}$$

$$\mathbf{Z}=\begin{bmatrix}— & \mathbf{z}_1 & —\\ — & \mathbf{z}_2 & —\\ & \vdots & \\ — & \mathbf{z}_N & —\end{bmatrix}$$

根據題目所述，$\mathbf{Z}$ 是個 Vandermonde matrix，其行列式為：

$$\det(\mathbf{Z})=\prod_{1\le n\le m\le N}(x_m-x_n)$$

而且題目有說明每個 x_i 均相異，因此行列式值不為零，也因此可逆。

根據 Linear Regression 的最佳解形式：

$$\tilde{\mathbf{w}}=(\mathbf{Z^T}\mathbf{Z})^{-1}\mathbf{Z^T}\mathbf{y}$$

因為 $\mathbf{Z}$ 可逆，所以可以知道 $\mathbf{Z^T}$ 也可逆，又可以知道 $\mathbf{Z^T}\mathbf{Z}$ 也可逆。

所以 $E_{in}(\tilde{\mathbf{w}})$ 就可以改寫為：

$$E_{in}(\tilde{\mathbf{w}})=\frac{1}{N}\|\mathbf{Z}\tilde{\mathbf{w}}-\mathbf{y}\|^2=\frac{1}{N}\left\|\mathbf{Z}(\mathbf{Z^T}\mathbf{Z})^{-1}\mathbf{Z^T}\mathbf{y}-\mathbf{y}\right\|^2$$

$$=\frac{1}{N}\left\|\mathbf{Z}\mathbf{Z}^{-1}\left(\mathbf{Z^T}\right)^{-1}\mathbf{Z^T}\mathbf{y}-\mathbf{y}\right\|^2$$

$$=\frac{1}{N}\|\mathbf{I}_N\mathbf{y}-\mathbf{y}\|^2=0$$

可知經過 $\boldsymbol{\Phi}(\boldsymbol{x_i})$ 這樣的轉換可以使 E_{in} 等於 0。。

因此確實存一個 Q-dimensional polynomial transform 可以使E_{in}等於 0。

3. peeking

首先我們可以知道經過 $\mathbf{\Phi}$ 轉換後的資料組成的大矩陣就是單位方陣：

$$\mathbf{Z}=\begin{bmatrix}— & \mathbf{z}_1 & — \\ — & \mathbf{z}_2 & — \\ & \vdots & \\ — & \mathbf{z}_N & —\end{bmatrix}=\begin{bmatrix}\mathbf{1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1}\end{bmatrix}$$

所以由 Linear Regression 所得到的最佳權重會是：

$$\widetilde{\mathbf{w}}=(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}\mathbf{y}=\mathbf{I}_N\mathbf{y}=\mathbf{y}$$

因此可以知道：

$$E_{in}(\widetilde{\mathbf{w}})=\frac{1}{N}\|\mathbf{Z}\widetilde{\mathbf{w}}-\mathbf{y}\|^2=\frac{1}{N}\|\mathbf{I}_N\mathbf{y}-\mathbf{y}\|^2=0$$

至於$E_{out}(\widetilde{\mathbf{w}})$的話，將***測試資料***中的 x_m 經過 $\mathbf{\Phi}$ 轉換的 $\mathbf{z}_m$ 會是個只有一個 1，其他都是 0 的$1\times N$矩陣。因為***訓練資料***中每個 x_n 都不一樣，如果***測試資料*** x_m 跟其中一個***訓練資料*** x_n 一樣，那麼就會跟其他***訓練資料***不一樣，導致只有一個值是 1 其他是 0；如果***測試資料*** x_m 跟所有***訓練資料*** x_n 都不一樣，那麼就會是全都是 0 的 0 矩陣。

但其實所有的 $\mathbf{z}_m$全都是 0 矩陣：由於訓練資料跟測試資料的 x 都是 i.i.d. 的以連續型均勻分佈取樣，因此理論上發生特定值的機率為 0，***測試資料*** 中的 x_m 經過轉換的$\mathbf{z}_m$會全部都是 0 矩陣，因此 square error 就會是 x_m 對應的 y_m 的平方：

$${y_m}^2=(x_m+\epsilon_m)^2$$

所以$E_{out}(\widetilde{\mathbf{w}})$可以表示為：

$$\begin{aligned}&E_{out}(\widetilde{\mathbf{w}})=\mathcal{E}[(x+\epsilon)^2]\\&=\mathcal{E}[x^2]+\mathcal{E}[2x\epsilon]+\mathcal{E}[\epsilon^2]\\&=Var[x]-\mathcal{E}[x]^2+2\mathcal{E}[x]\mathcal{E}[\epsilon]+Var[\epsilon]-\mathcal{E}[\epsilon]^2\\&=\frac{1}{3}-0^2+2\times0\times0+1-0^2=\frac{4}{3}\end{aligned}$$

● Combatting Overfitting

4. virtual examples

首先先列出 $\mathbf{X_h^T X_h}$ ：

$$\mathbf{X_h^T X_h} = \begin{bmatrix} \sum_{i=1}^{2N} x_{i,0}x_{i,0} & \sum_{i=1}^{2N} x_{i,0}x_{i,1} & \cdots & \sum_{i=1}^{2N} x_{i,0}x_{i,d} \\ \sum_{i=1}^{2N} x_{i,1}x_{i,0} & \sum_{i=1}^{2N} x_{i,1}x_{i,1} & \cdots & \sum_{i=1}^{2N} x_{i,1}x_{i,d} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{2N} x_{i,d}x_{i,0} & \sum_{i=1}^{2N} x_{i,d}x_{i,1} & \cdots & \sum_{i=1}^{2N} x_{i,d}x_{i,d} \end{bmatrix}$$

這裡我令x_{N+1}到x_{2N}是加入 noise 後的資料。接著將上面的矩陣拆成：

$$\mathbf{X_h^T X_h} = \begin{bmatrix} \sum_{i=1}^{N} x_{i,0}x_{i,0} & \sum_{i=1}^{N} x_{i,0}x_{i,1} & \cdots & \sum_{i=1}^{N} x_{i,0}x_{i,d} \\ \sum_{i=1}^{N} x_{i,1}x_{i,0} & \sum_{i=1}^{N} x_{i,1}x_{i,1} & \cdots & \sum_{i=1}^{N} x_{i,1}x_{i,d} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{N} x_{i,d}x_{i,0} & \sum_{i=1}^{N} x_{i,d}x_{i,1} & \cdots & \sum_{i=1}^{N} x_{i,d}x_{i,d} \end{bmatrix} + \begin{bmatrix} \sum_{i=N+1}^{2N} x_{i,0}x_{i,0} & \sum_{i=N+1}^{2N} x_{i,0}x_{i,1} & \cdots & \sum_{i=N+1}^{2N} x_{i,0}x_{i,d} \\ \sum_{i=N+1}^{2N} x_{i,1}x_{i,0} & \sum_{i=N+1}^{2N} x_{i,1}x_{i,1} & \cdots & \sum_{i=N+1}^{2N} x_{i,1}x_{i,d} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=N+1}^{2N} x_{i,d}x_{i,0} & \sum_{i=N+1}^{2N} x_{i,d}x_{i,1} & \cdots & \sum_{i=N+1}^{2N} x_{i,d}x_{i,d} \end{bmatrix}$$

加號左半邊就是原本的$\mathbf{X^T X}$；所以我們接著只關注右半邊，右半邊可以改成用帶有 noise 的形式來表示，順便換掉 i 的起始位置：

$$\begin{bmatrix} \sum_{i=1}^{N}(x_{i,0}+\epsilon_{i,0})(x_{i,0}+\epsilon_{i,0}) & \sum_{i=1}^{N}(x_{i,0}+\epsilon_{i,0})(x_{i,1}+\epsilon_{i,1}) & \cdots & \sum_{i=1}^{N}(x_{i,0}+\epsilon_{i,0})(x_{i,d}+\epsilon_{i,d}) \\ \sum_{i=1}^{N}(x_{i,1}+\epsilon_{i,1})(x_{i,0}+\epsilon_{i,0}) & \sum_{i=1}^{N}(x_{i,1}+\epsilon_{i,1})(x_{i,1}+\epsilon_{i,1}) & \cdots & \sum_{i=1}^{N}(x_{i,1}+\epsilon_{i,1})(x_{i,d}+\epsilon_{i,d}) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{N}(x_{i,d}+\epsilon_{i,d})(x_{i,0}+\epsilon_{i,0}) & \sum_{i=1}^{N}(x_{i,d}+\epsilon_{i,d})(x_{i,1}+\epsilon_{i,1}) & \cdots & \sum_{i=1}^{N}(x_{i,d}+\epsilon_{i,d})(x_{i,d}+\epsilon_{i,d}) \end{bmatrix}$$

可以發現每一個 $\sum_{i=1}^{N}(x_{i,j}+\epsilon_{i,j})(x_{i,k}+\epsilon_{i,k})$ 都可以拆成：

$$\sum_{i=1}^{N}(x_{i,j}+\epsilon_{i,j})(x_{i,k}+\epsilon_{i,k}) = \sum_{i=1}^{N} x_{i,j}x_{i,k} + x_{i,k}\epsilon_{i,j} + x_{i,j}\epsilon_{i,k} + \epsilon_{i,j}\epsilon_{i,k}$$

如果取期望值會得到：

$$\mathcal{E}\left[\sum_{i=1}^{N} x_{i,j}x_{i,k} + x_{i,k}\epsilon_{i,j} + x_{i,j}\epsilon_{i,k} + \epsilon_{i,j}\epsilon_{i,k}\right] = \sum_{i=1}^{N} x_{i,j}x_{i,k} + \mathcal{E}[\epsilon_{i,j}\epsilon_{i,k}]$$

因為$\mathcal{E}[\epsilon_{i,j}] = 0$所以中間兩項都消掉了。至於最後的一項要分兩種情形：

$\mathcal{E}[\epsilon_{i,j}\epsilon_{i,k}] = \frac{\delta^2}{3}$ 這是當$j = k$的時候，也就是對角線上。

$\mathcal{E}[\epsilon_{i,j}\epsilon_{i,k}] = 0$ 這是當$j \neq k$的時候，也就是其他地方：因為這兩個ϵ是兩個不同的取樣，所以根據獨立性值 $\mathcal{E}[\epsilon_{i,j}\epsilon_{i,k}] = \mathcal{E}[\epsilon_{i,j}]\mathcal{E}[\epsilon_{i,k}] = 0$。
所以最後加號右半邊的矩陣取期望值就可以表示為：

$$\mathbf{X}^T\mathbf{X} + \begin{bmatrix} \sum_{i=1}^{N} \mathcal{E}[\epsilon_{i,0}\epsilon_{i,0}] & 0 & \dots & 0 \\ 0 & \sum_{i=1}^{N} \mathcal{E}[\epsilon_{i,1}\epsilon_{i,2}] & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sum_{i=1}^{N} \mathcal{E}[\epsilon_{i,\mathrm{d}}\epsilon_{i,\mathrm{d}}] \end{bmatrix}$$

$$= \mathbf{X}^T\mathbf{X} + \frac{N\delta^2}{3}\boldsymbol{I}_N$$

所以最後全部統整起來就會得到：

$$\mathcal{E}[\mathbf{X}_\mathbf{h}^T\mathbf{X}_\mathbf{h}] = \underbrace{\mathbf{X}^T\mathbf{X}}_{\text{左半邊}} + \underbrace{\mathbf{X}^T\mathbf{X} + \frac{N\delta^2}{3}\boldsymbol{I}_N}_{\text{右半邊}} = \mathbf{2}\mathbf{X}^T\mathbf{X} + \frac{N\delta^2}{3}\boldsymbol{I}_N$$

5. augmented error with GD

根據 gradient descent algorithm：

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta\nabla E_{\mathrm{aug}}(\mathbf{w}_\mathrm{t})$$

所以我們先找出∇E_{aug}，接著再帶入：

$$\nabla E_{aug}(\mathbf{w}_t) = \nabla E_{\mathrm{in}}(\mathbf{w}_t) + \frac{2\lambda}{N}\mathbf{w}_t$$

$$\Rightarrow \mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta\left(\nabla E_{\mathrm{in}}(\mathbf{w}_t) + \frac{2\lambda}{N}\mathbf{w}_t\right)$$

$$\mathbf{w}_{t+1} \leftarrow \left(1 - \frac{2\lambda\eta}{N}\right)\mathbf{w}_t - \eta\nabla E_{\mathrm{in}}(\mathbf{w}_t)$$

$$\Rightarrow \mathbf{w}_{t+1} \leftarrow \left(1 - \frac{2\lambda\eta}{N}\right)\left(\mathbf{w}_t - \frac{\eta}{1 - \frac{2\lambda\eta}{N}}\nabla E_{\mathrm{in}}(\mathbf{w}_t)\right)$$

所以可以知道：

$$\alpha = \left(1 - \frac{2\lambda\eta}{N}\right) < 1, \qquad \beta = \frac{\eta}{\alpha}$$

6. relationship

只要解出最佳的 w 就可以了：

$$\min_{w\in\mathbb{R}} \frac{1}{N}\sum_{n=1}^{N}(w\cdot x_n - y_n)^2 + \frac{\lambda}{N}w^2$$

對 w 取一次微分後找極值：

$$\frac{2}{N}\sum_{n=1}^{N}(w\cdot x_n^2 - x_n y_n) + \frac{2\lambda}{N}w = 0$$

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

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

所以將$w = \sqrt{C}$帶入就可以得到 λ 的關係式：

$$\sqrt{C} = \frac{\sum_{n=1}^{N}x_n y_n}{\sum_{n=1}^{N}x_n^2 + \lambda}$$

$$\Rightarrow \sum_{n=1}^{N}x_n^2 + \lambda = \frac{1}{\sqrt{C}}\sum_{n=1}^{N}x_n y_n$$

$$\Rightarrow \lambda = \frac{1}{\sqrt{C}}\sum_{n=1}^{N}x_n y_n - \sum_{n=1}^{N}x_n^2$$

所以我們可以知道：

$$\alpha = \sum_{n=1}^{N}x_n y_n\,,\beta = -\sum_{n=1}^{N}x_n^2$$

7. Regularizer

先稍微轉換一下：

$$\frac{1}{N}\sum_{n=1}^{N}(\widetilde{\mathbf{w}}^{\mathrm{T}}\boldsymbol{\Phi}(\mathbf{x}_n) - y_n)^2 + \frac{\lambda}{N}\|\widetilde{\mathbf{w}}\|_1 = \frac{1}{N}\sum_{n=1}^{N}(\widetilde{\mathbf{w}}^{\mathrm{T}}\mathbf{V}\mathbf{x}_n - y_n)^2 + \frac{\lambda}{N}\|\widetilde{\mathbf{w}}\|_1$$

由於 $\mathbf{V}$ 矩陣是個都是正數的對角矩陣，所以可以做出下面的更動：

$$= \frac{1}{N}\sum_{n=1}^{N}(\widetilde{\mathbf{w}}^{\mathrm{T}}\mathbf{V}\mathbf{x}_n - y_n)^2 + \frac{\lambda}{N}\left\|\widetilde{\mathbf{w}}^{\mathrm{T}}\mathbf{V}\mathbf{V}^{-1}\right\|_1$$

然後此時令$\mathbf{w} = \widetilde{\mathbf{w}}^{\mathrm{T}}\mathbf{V}$，上面的式子就可以得到：

$$= \frac{1}{N}\sum_{n=1}^{N}(\mathbf{w}\mathbf{x}_n - y_n)^2 + \frac{\lambda}{N}\|\mathbf{w}\mathbf{V}^{-1}\|_1$$

這樣就找到他們之間的關係了，$\boldsymbol{\Omega(\mathbf{w}) = \|\mathbf{w}\mathbf{V}^{-1}\|_1}$。

也就是說對原本的式子找到某個$\widetilde{\mathbf{w}}$使E_{in}最小，就等同在稍微換位子後的E_{in}式子中找到一個$\mathbf{w}$使E_{in}最小。

8. Leave one out

$\mathcal{A}_{minority}$永遠只會回傳數量最少的類別。題目正負兩種資料類別都給了 N 個，所以如果挑出一個 x_n 用來 Loocv 預測，其他做訓練，則：

$$\begin{cases} \mathcal{A}_{minority}(x_n) = 1 & if\ y_n = +1, \text{因為此時} +1 \text{變成 N} - 1 \text{個} \\ \mathcal{A}_{minority}(x_n) = -1 & if\ y_n = -1, \text{因為此時} -1 \text{變成 N} - 1 \text{個} \end{cases}$$

會發現$\mathcal{A}_{minority}$就可以神奇的完美預測了。

根據E_{loocv}的公式可以知道：

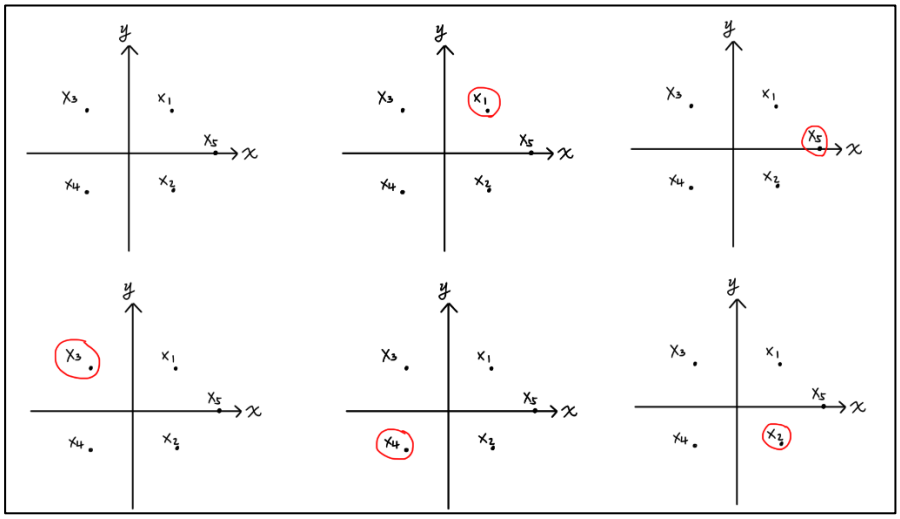
$$\frac{1}{2N}\sum_{n=1}^{2N} err\left(\mathcal{A}_{minority}(x_n), y_n\right) = \frac{1}{2N}\sum_{n=1}^{2N}[\![\mathcal{A}_{minority}(x_n) \neq y_n]\!] = 0$$

達到完美的 100%正確率。

● Learning Principles

9. value of the expectation

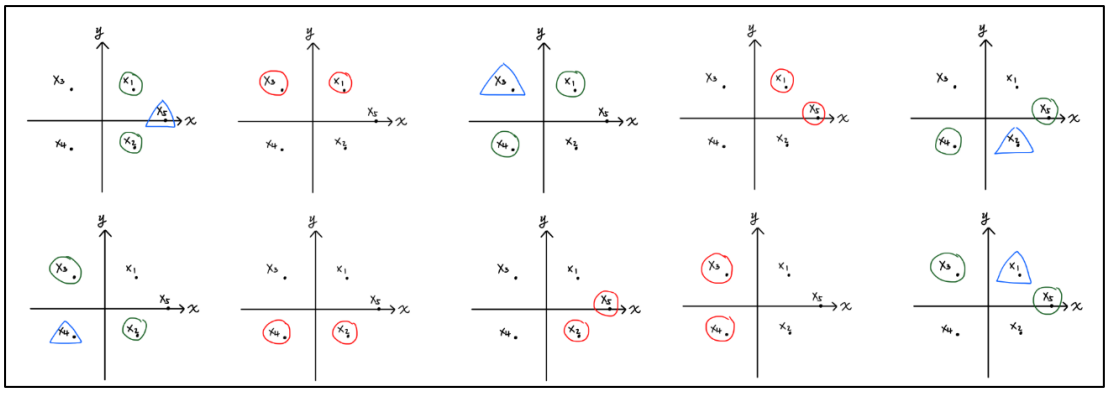
下面列出所有的可能：



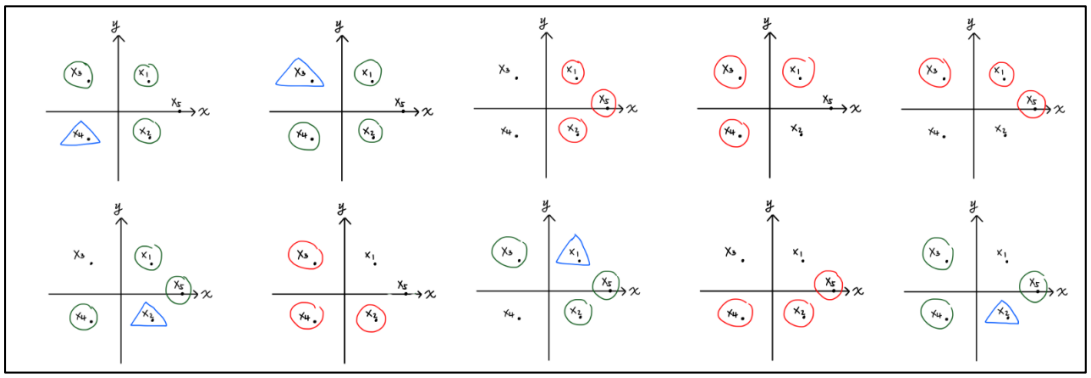
上圖中，圈起來的代表 label 是+1，沒圈起來的代表是−1。

紅色的圈代表該種情形可以用一條線完美分開；左上角的都沒圈情形，也是一種可以完美分開的情形。

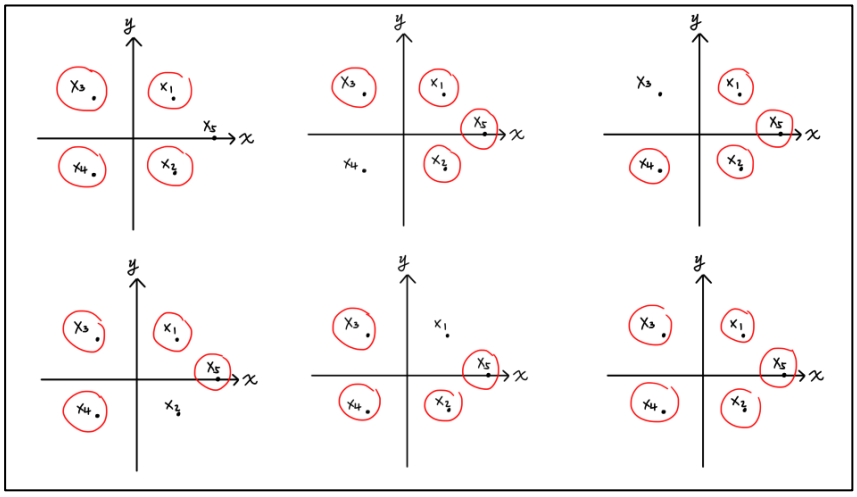
上圖是「都不圈」與「圈一個」的情形。



上圖是「圈兩個」的情形。有「圈」一樣代表該點的 label 是+1。這裡多了綠色圈，代表該種情形無法用一條線完美分開，並且我用「藍色三角形」代表了我們犧牲的那個點，他是預測錯誤的點。
三角形不是圈。



上圖是圈三個的情形，規則一樣。



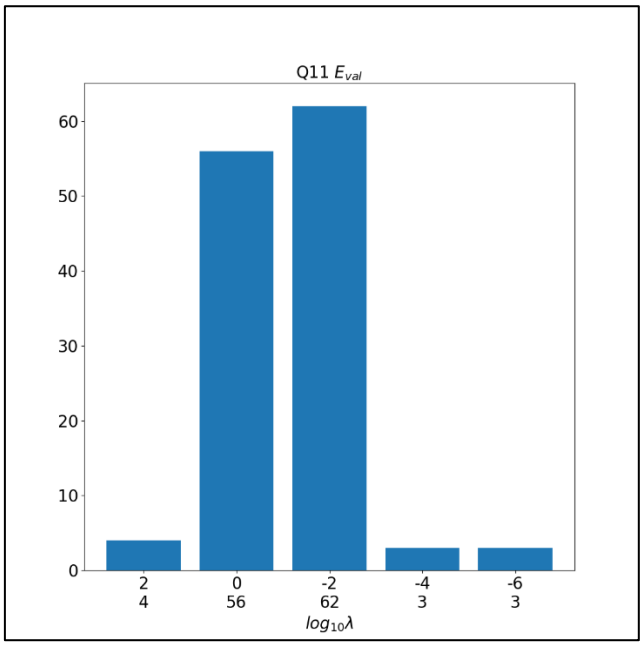
最後是圈四個跟圈五個，可以發現都可以完美預測。
現在我們來算算期望值：
完美預測的錯誤率是 0，所以我們只要考慮有錯的就好。
從上面的圖可以發現，有錯的話，就只有錯一個點，也就是說錯誤率是 $\frac{1}{5}$，而總共有 10 種情形是有預測錯誤的，再乘上發生的機率，就得

到我們要的期望值了：

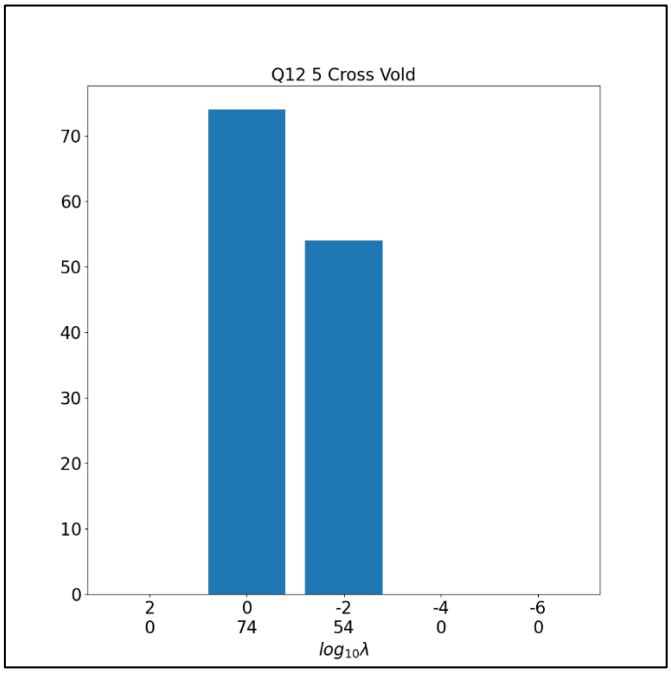
$$\sum_{i=1}^{10} \frac{1}{5} \times \frac{1}{32} = \frac{10}{5 \times 32} = \frac{1}{16}$$

- Experiments with Regularized Logistic Regression

10. best λ among E_{in}
    根據實驗結果，最佳的 λ 是−6次方。
11. best λ among E_{val}



12. best λ among 5 fold Cross Validation

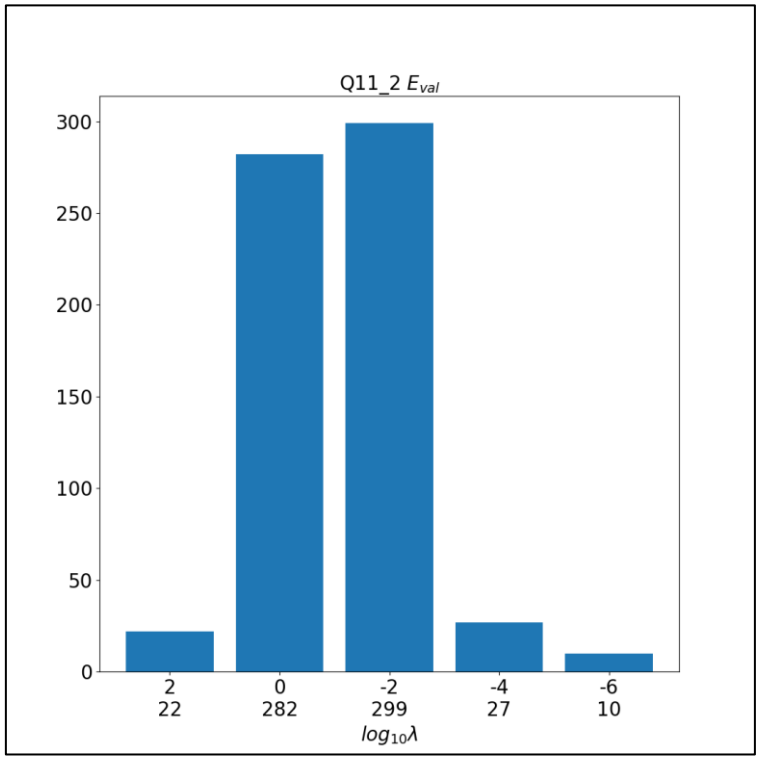


首先可以發現有經過驗證操作的兩種結果得到的λ主要都是 0 或 −2 次方，而使用E_{in}得到的則是−6次方，這裡可以看出使用E_{in}做判斷會造成的 overfitting 現象。

接著是單純使用 1 次驗證以及使用 Cross Validation 所得到的差異，單純使用 1 次驗證會發現有零星的情形選擇了 2 、 −4 跟 −6 次方，Cross Validation 則沒有這樣的情形；兩者都主要選擇 0 或 −2 次方，並且 Cross Validation 更偏好 0 次方，1 次驗證則是稍微偏好 −2 次方。

對於零星的情形的差異性，我認為是因為 Cross Validation 可以綜合考量 5 個不同位置的等分去進行評估，所以如果某次 random select 產生的資料接近極端情形，讓 1 次驗證的判斷產生誤差，此時 Cross Validation 的特性就可以避免這種情況。

至於 Cross Validation 偏好 0 次方，1 次驗證稍微偏好 −2 次方的現象，原本我以為是實驗上的誤差，但是當我將 1 次驗證的次數拉到 128 的 5 倍後，發現得到一樣的分佈圖：



我才意識到應該是跟剛剛零星情形的現象具有相同的原因：Cross Validation 考慮了五種意見而做出決定；某次 Cross Validation 選擇了 0 次方，1 次驗證因為只考慮一種意見，所以有可能會受到該資料的分佈的影響而選擇了 −2 次方。

● Bonus

13. Scale or Regularize

首先要注意到老師給的條件，$\|\mathbf{w}_{LIN}\|^2 > C$，因為這代表最佳的$\mathbf{w}_{LIN}$長度是大於$\sqrt{C}$，一旦我們加上 L2 Regularizer (或者說 C-constrained linear regression)後得到的最佳權重長度就會在邊界上，也就是說：

$$\|\mathbf{w}_{L2}\|^2 = C \text{。}$$

- If $\mathbf{X}^T\mathbf{X} = \alpha\mathbf{I}$, then $\mathbf{w}_C$ solves the C-constrained linear regression prob.

  根據 ridge regression 的解形式：

$$\mathbf{w}_{L2} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$$

$$\Rightarrow \mathbf{w}_{L2} = (\alpha\mathbf{I} + \boldsymbol{\lambda}\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} = \big((\alpha+\lambda)\mathbf{I}\big)^{-1}\mathbf{X}^T\mathbf{y}$$

$$= \frac{1}{\alpha+\lambda}\mathbf{I}\mathbf{X}^T\mathbf{y} = \frac{1}{\alpha+\lambda}\mathbf{X}^T\mathbf{y}$$

  此時如果再看原本的 linear regression 的解形式：

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

$$\Rightarrow \mathbf{w}_{LIN} = (\alpha\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} = \frac{1}{\alpha}\mathbf{X}^T\mathbf{y}$$

  會發現$\mathbf{w}_{L2}$跟$\mathbf{w}_{LIN}$其實就只差一個常數倍；而我們知道$\|\mathbf{w}_{L2}\| = \sqrt{C}$，因此將$\mathbf{w}_{LIN}$長度伸縮成$\sqrt{C}$而得到的$\mathbf{w}_C$就是$\mathbf{w}_{L2}$：

$$\mathbf{w}_C = \frac{\mathbf{w}_{LIN}}{\|\mathbf{w}_{LIN}\|}\cdot\sqrt{C} = \mathbf{w}_{L2}$$

- If $\mathbf{w}_C$ solves the C-constrained linear regression prob., then $\mathbf{X}^T\mathbf{X} = \alpha\mathbf{I}$

  如果$\mathbf{w}_C$ solves the C-constrained linear regression prob.，我們可以知道$\mathbf{w}_C = \mathbf{w}_{L2}$，也就是說：

$$\underbrace{\frac{(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}}{\|(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}\|}\cdot\sqrt{C}}_{\mathbf{w}_C} = \underbrace{(\mathbf{X}^T\mathbf{X}+\boldsymbol{\lambda}\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}}_{\mathbf{w}_{L2}}$$

$$\frac{\sqrt{C}}{\|(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}\|}\cdot(\mathbf{X}^T\mathbf{X})^{-1} = (\mathbf{X}^T\mathbf{X}+\boldsymbol{\lambda}\mathbf{I})^{-1}$$

$$\Rightarrow \frac{\sqrt{C}}{\|(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}\|}\cdot(\mathbf{X}^T\mathbf{X}+\boldsymbol{\lambda}\mathbf{I}) = (\mathbf{X}^T\mathbf{X})$$

$$\Rightarrow \left(\frac{\sqrt{C}}{\|(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}\|} - 1\right)\mathbf{X}^T\mathbf{X} = -\boldsymbol{\lambda}\mathbf{I}$$

$$\Rightarrow \mathbf{X}^T\mathbf{X} = \frac{-\boldsymbol{\lambda}}{\frac{\sqrt{C}}{\|(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}\|} - 1}\mathbf{I} = \alpha\mathbf{I}$$

注意記得確認$\mathbf{X}^T\mathbf{X}$得到的結果要滿足過程中$\mathbf{X}^T\mathbf{X}$跟$\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I}$可逆的假設。