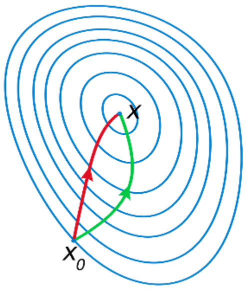
Newton’s method of logistic function

上一堂課使用 Gradient descent 來找出最佳 logistic regression 的 w 參數，現在我們要用第二堂課使用的 Newton's method，因為 Newton method 較 Gradient descent 快達到收斂。



圖片來源：維基百科

綠線是採用 Gradient descent，紅線是採用牛頓法，可看到牛頓法能夠較快收斂

還記得 Hessian matrix 嗎？回憶一下

$$Hession\ \ function\ of\ \ f(x) = Hf(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial {x_0}^2} & \frac{\partial^2 f}{\partial x_0 \partial x_1} & \dots \\ \frac{\partial^2 f}{\partial x_1 \partial x_0} & \frac{\partial^2 f}{\partial {x_1}^2} & \dots \\ \dots & \dots & \dots \end{bmatrix}$$

還有 Newton’s method

$$x_1 = x_0 - \frac{f'(x_0)}{f''(x_0)} = x_0 - Hf(x)^{-1} \nabla f(x_0)$$

要做 Hessian matrix 之前，要先來推導 $\frac{\partial}{\partial w_k} \frac{\partial}{\partial w_j} J$

$$\frac{\partial}{\partial w_k} \frac{\partial J}{\partial w_j} = \frac{\partial}{\partial w_k} \sum_{i=1}^{n} (x_{ij}(y_i - \frac{1}{1+e^{-x_i \mathbf{w}}})) = \frac{-\partial}{\partial w_k} \sum_{i=1}^{n} \frac{x_{ij}}{1+e^{-x_i \mathbf{w}}}$$

$$= \frac{-\partial}{\partial w_k} \sum_{i=1}^{n} \frac{x_{ij}}{1+e^{-(w_1 x_{i1} + \dots + w_k x_{ik} + \dots + w_d x_{id})}}$$

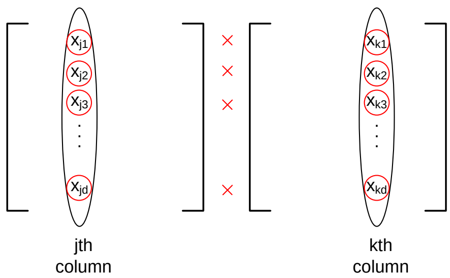
$$= \sum_{i=1}^{n} \frac{x_{ij} x_{ik} e^{-(w_1 x_{i1} + \dots + w_k x_{ik} + \dots + w_d x_{id})}}{(1+e^{-(w_1 x_{i1} + \dots + w_k x_{ik} + \dots + w_d x_{id})})^2}$$

$$= \sum_{i=1}^{n} \frac{x_{ij} x_{ik} e^{-X_i \mathbf{w}}}{(1+e^{-X_i \mathbf{w}})^2}$$

寫成 matrix form

令 $A=\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2d} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nn} \end{bmatrix}$

我們仔細觀察 $x_{ij}x_{ik}$ 這項



而 $\dfrac{e^{-X_i\mathbf{w}}}{(1+e^{-X_i\mathbf{w}})^2}$ 為一常數，故

$$H(J)=A^T DA$$

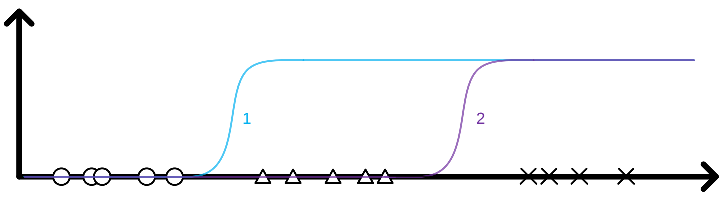
$$\text{而 } D=\begin{bmatrix} \dfrac{e^{-X_1\mathbf{w}}}{(1+e^{-X_1\mathbf{w}})^2} & 0 & \dots & 0 \\ 0 & \dfrac{e^{-X_2\mathbf{w}}}{(1+e^{-X_2\mathbf{w}})^2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \dfrac{e^{-X_d\mathbf{w}}}{(1+e^{-X_d\mathbf{w}})^2} \end{bmatrix}$$

note: $H(J)$ 為一半正定矩陣，並不保證可逆。

Solving problem of extreme data

Logistic regression 介紹到這，最重要的一件事就是，有沒有解決極端值的問題
一樣的我們要將這些 data 分類

我們要分成三類，故我們需要兩條 logistic regression

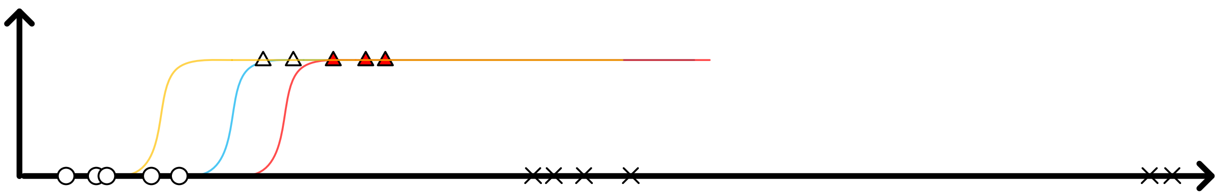


若是 O 群，第 1 條 logistic regression 會分在第 0 群，第 2 條 logistic regression 會被分在第 0 群
其他的以此類推，統整在下表

	regression 1	regression 2
O	0	0
Δ	1	0
X	1	1

我們來思考一下 logistic regression 是否能夠做到上列的事情且不受極端值影響，所以我再增加幾個超極端值試試看

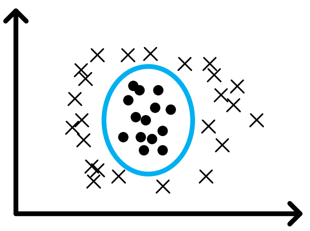
logistic regression1



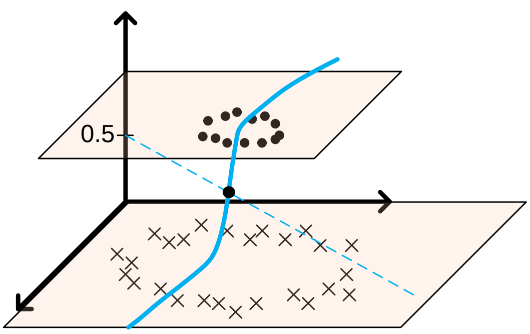
只有藍色的 regression 是最佳的，若結果是黃線，會比藍色 regression 多錯黃色圓圈的 data，同樣的，若是紅線，會比藍色線多錯紅色三角形的 data，故逼近到最後必會是藍色 regression 線

2+ dimension problem of logistic regression

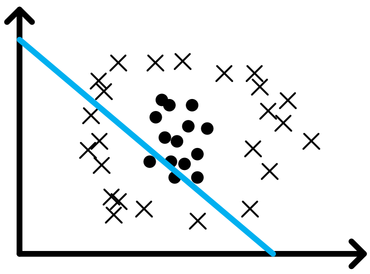
　　但 logistic regression 一旦用在 2 維以上的空間中，就有可能會發生我們肉眼可畫出的 decision boundary，而 logistic regression 無法定義出來。像是下例



但若我們用 logistic regression 解，我們沒辦法畫出該 decision boundary

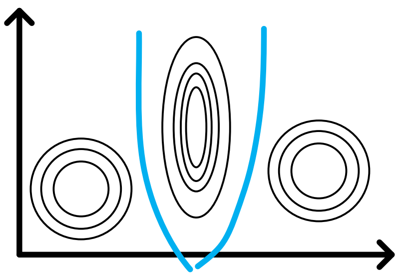


在二維平面顯示其 decision boundary 為



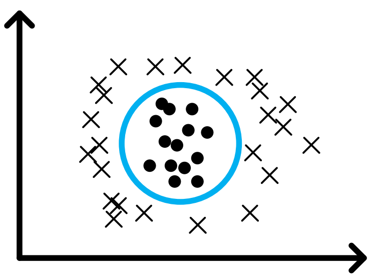
不是一個好的 regression 方法

通常，一般方法都沒有好的解決方式時，Naïve bayes 會是一個不錯的方法，這裡也同樣適用



藍線的 decision boundary 為兩分類機率相等的地方

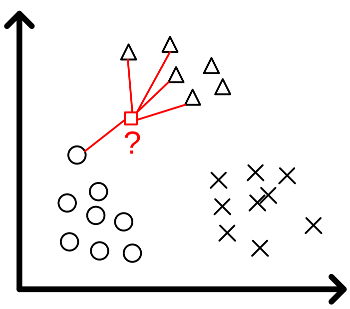
若是上面那個像甜甜圈的 case，Naïve bayes classifier 會形成下圖的 decision boundary



(怪)

k-nn algorithm(k-nearest neighbor)

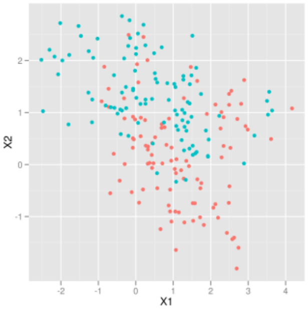
評判分類的方式非常直覺，就是看離自己最近的 k 個 data 中，最多的那種分類。



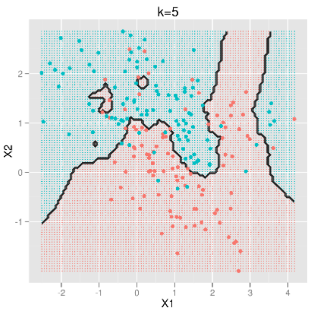
假設 k=5，決定□的分類就是看□最近的 5 個 data，有 4 個 Δ 和一個 O，故□屬於分類 Δ k-nn 沒有 bias，故會有很嚴重的 overfitting 問題。

舉例來源：http://idrisr.com/2012/04/09/knn.html(或課本 p126，只有實體書才有詳細的圖)

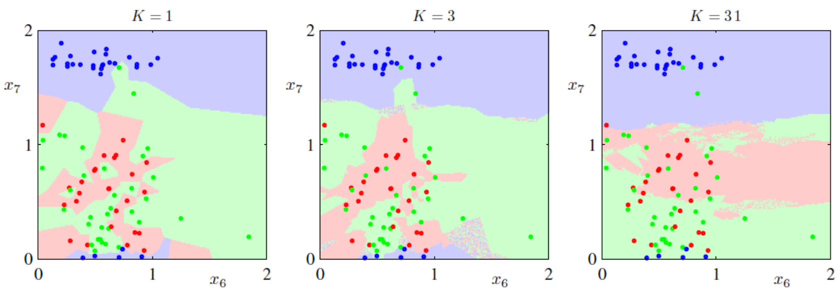
有筆資料需要分成兩類，其分布圖如下



使用 k-nn，k=5 得到下圖



當 k 越小，圖形的坑洞就會越多

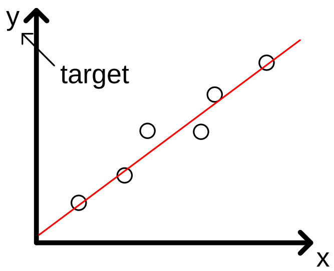


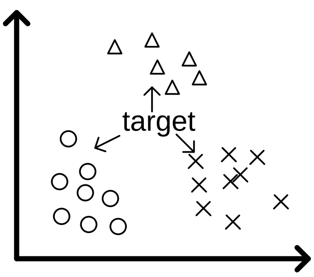
課本 p126

Clustering(unsupervised learning)

supervise learning

每個 train data 都有給定”目標”(target)，表示丟入輸入後應該要得到甚麼輸出才合理





supervise learning 需要 complete data，也就是每個 train data 輸入都需要給 label or target，regression 和 classification 都是屬於此類

unsupervised learning

只有給 incomplete data，如果是 regression 問題，就是只有給輸入沒有給輸入後的輸出。
e.g.

1. EM algorithm
2. K means clustering
3. Gaussian Mixture model(GMM)

直覺的 hierarchical clustering

同一類的 data 通常彼此距離較靠近，hierarchical clustering 就是使用這個性質，一步一步找出現在可以融合為一群的兩個 group or 點

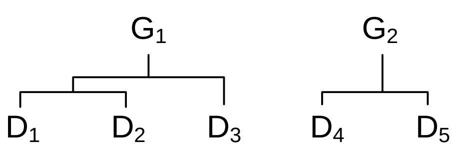
Step1 測量 data 間的距離/相似度，假設有 k 個 data，故需要測 $\binom{k}{2}$ 次

此步做完後可得到一 $k \times k$ 的矩陣

Step2 將最接近的兩點視為同一個 group，並將此兩點視為同一個 group

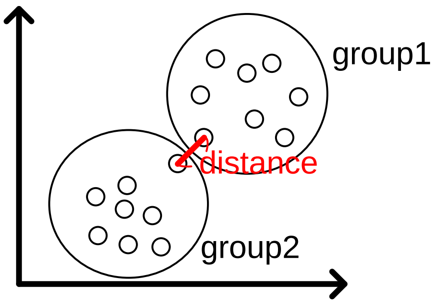
重複這兩個步驟，每次做完會少一個 group，一直重複直到目前的 group 數為我們希望分類的數量為止

e.g. 若要分 2 類

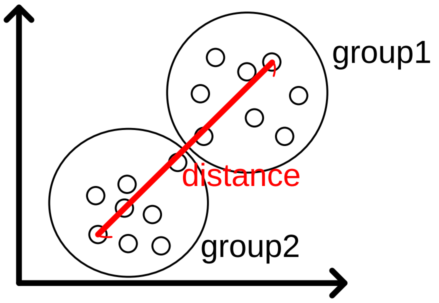


note:
要如何定義兩群組的距離，有多種定義的方式
參考網站：http://mirlab.org/jang/books/dcpr/dcHierClustering.asp?title=3-2%20Hierarchical%20Clustering%20(%B6%A5%BCh%A6%A1%A4%C0%B8s%AAk)&language=chinese

1. 單一連結聚合演算法（single-linkage agglomerative algorithm）
   兩群組間最近的兩點的距離



2. 完整連結聚合演算法（complete-linkage agglomerative algorithm）
   兩群組間最遠兩點的距離



3. 平均連結聚合演算法（average-linkage agglomerative algorithm）
   兩群組間每兩點距離的平均

4. 沃德法（Ward's method）
   兩群組合併後，合併的群中心到各點的距離平方和