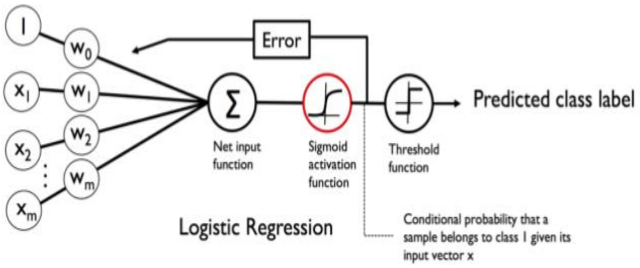
Logistic Regression

原理

為監督式學習，用於預測類別資料與其機率。
將點帶進去回歸線$\hat{y}$，回歸線輸出值若是>=0，是一類(target)，值<0 是另一類(non-target)。而 Logistic Regression 則是一個平滑的曲線，當 w0*x0+w1*x1+…+wn*xn 越大時判斷成 A 類的機率越大，越小時判斷成 A 類的機率越小。



$$h_\theta(x_1, x_2, x_3) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$
$$= \vec{\theta} \cdot \vec{x}$$

Logistic Regression:

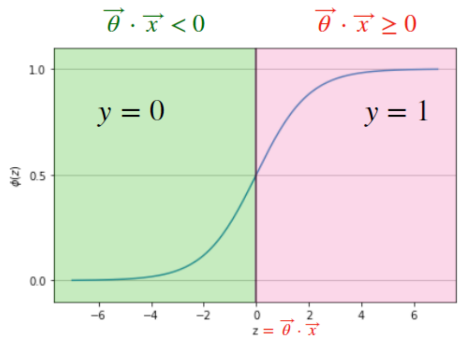
$$\hat{y} = \frac{1}{1 + \exp\left(-\vec{\theta} \cdot \vec{x}\right)}$$

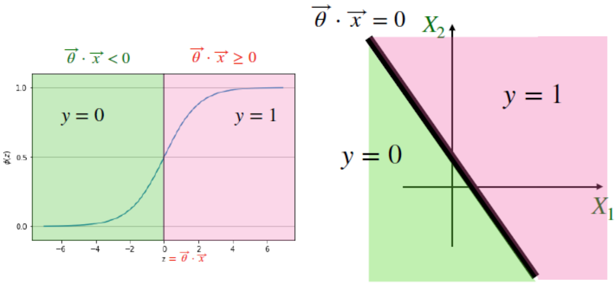
- logistic function

以 Sigmoid 函數為例，這個函數的 y 的值介於 0~1，這樣的分布也符合機率是在 0~1 的範圍中。
因此，將點帶入回歸線$\hat{y}$，當 Z=0 時，經過 Sigmoid 函數判斷成+1 類(A 類)的機率為 0.5；z >0 判斷成 A 類的機率就會>0.5；z <0 判斷成 A 類的機率就會<0.5。而當 A 類判斷機率>0.5 時，則會判斷成 A 類，反之則否。
(也可以改用其他符合 0~1 的函數，因為機率的值是介於 0~1）

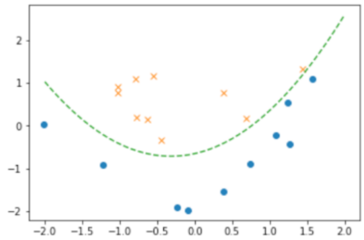
$$\hat{y} = P(y = 1 \mid x) \geq 0.5 \Rightarrow y = 1$$
$$\hat{y} = P(y = 1 \mid x) < 0.5 \Rightarrow y = 0$$

$$z = w^T x$$
$$\phi(z) = \frac{1}{1 + e^{-z}}$$





$$\vec{\theta}\cdot\vec{x} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2$$



- Optimizing Cost Function

若 y = 1，希望 P(y = 1| x) 越大越好；若 y = 0，希望 P(y = 0| x)越大越好，因此，期望極大化所有點經過 logistic function 得出的機率乘積能極大化:

$$\prod_{i=1}^{m} P(y^{(i)} = 1 \,|\, x^{(1)})^{y^{(i)}} P(y^{(i)} = 0 \,|\, x^{(i)})^{1-y^{(i)}}$$

若 $y^{(i)} = 1$

$$P(y^{(i)} = 1 \,|\, x^{(1)})^{y^{(i)}} P(y^{(i)} = 0 \,|\, x^{(i)})^{1-y^{(i)}} = P(y^{(i)} = 1 \,|\, x^{(1)})$$

若 $y^{(i)} = 0$

$$P(y^{(i)} = 1 \,|\, x^{(1)})^{y^{(i)}} P(y^{(i)} = 0 \,|\, x^{(i)})^{1-y^{(i)}} = P(y^{(i)} = 0 \,|\, x^{(1)})$$

因為機率越乘越小，所以目標改為 Maximum:

$$\log\left(\prod_{i=1}^{m} P(y^{(i)} = 1 \,|\, x^{(1)})^{y^{(i)}} P(y^{(i)} = 0 \,|\, x^{(i)})^{1-y^{(i)}}\right)$$

$$= \sum_{i=1}^{m} y^{(i)} \log\left(P(y^{(i)} = 1 \,|\, x^{(1)})\right) + \left(1 - y^{(i)}\right) \log\left(P(y^{(i)} = 0 \,|\, x^{(i)})\right)$$

$$= \sum_{i=1}^{m} y^{(i)} \log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right)$$

利用梯度下降法求得最適參數

Gradient descent (梯度下降法)

$$x := x - \eta \frac{d}{dx} g(x)$$

Algorithm

$$\theta_j := \theta_j + \eta \sum_{i=1}^{m} \left(y^{(i)} - \hat{y}^{(i)} \right) x_j^{(i)}$$

加入正規化以防止模型 overfitting，確保權重加權值不會過大。

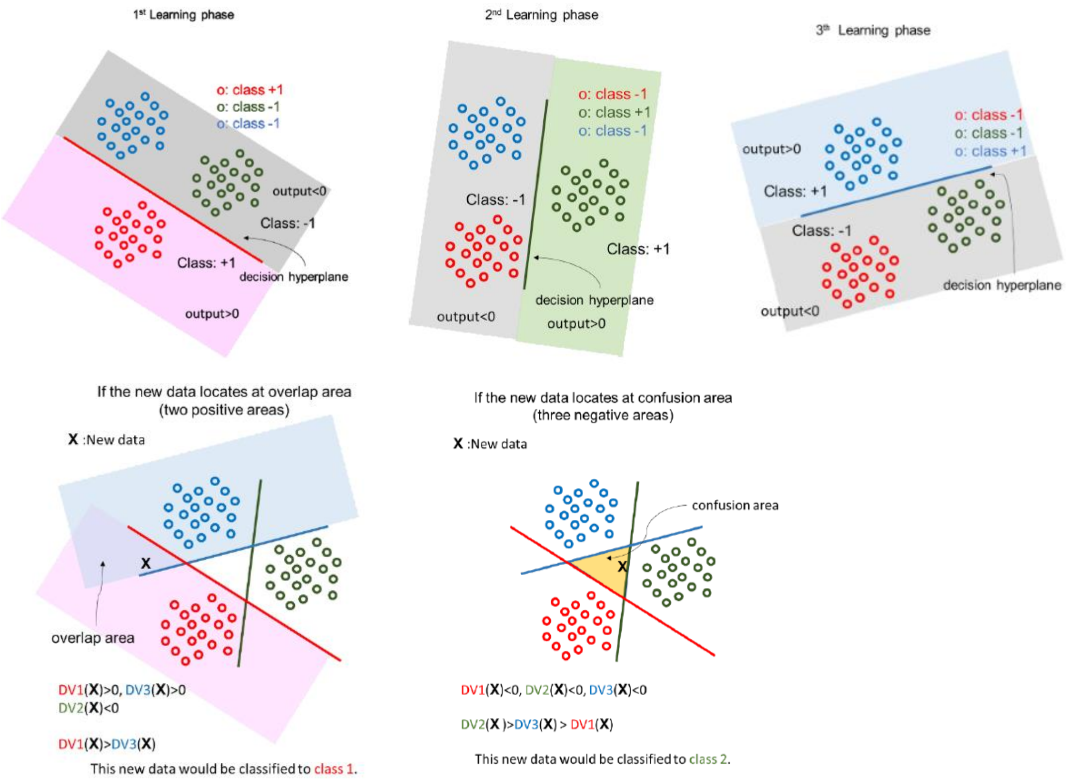
$$\sum_{i=1}^{m} \left[-y^{(i)} \log \left(\hat{y}^{(i)} \right) - \left(1 - y^{(i)} \right) \log \left(1 - \hat{y}^{(i)} \right) \right] + \lambda \sum_{i=1}^{n} \theta_i^2$$

λ : regularization parameter (正規化參數)

在多類別分類問題上使用用二元分類器進行分類

- one-against-all(OAA)

OAA 運作方式是先拿其中一類當作+1 類，剩下的類別當作-1 類，然後用二元分類器學習一次得到一個分類器/decision hyperplane；然後第二次繼續拿下一類當作+1 類，剩下的類別當作-1 類，學習一個分類器/decision hyperplane；直到所以的類別都有當作+1 類為止，最後看 decision value 哪個比較大，資料就判給哪一類。



參考資料

1. 機器學習(5)--邏輯斯迴歸，過度適合與正規化(Logistic regression，overfitting and regularization)
   http://arbu00.blogspot.com/2017/02/5-logistic-regressionoverfitting-and.html
2. 機器/統計學習: 羅吉斯回歸(Logistic regression)
   https://medium.com/@chih.sheng.huang821/%E6%A9%9F%E5%99%A8-%E7%B5%B1%E8%A8%88%E5%AD%B8%E7%BF%92-%E7%BE%85%E5%90%89%E6%96%AF%E5%9B%9E%E6%AD%B8-logistic-regression-aff7a830fb5d
3. [資料分析&機器學習] 第 3.3 講:線性分類-邏輯斯回歸(Logistic Regression)介紹
   https://medium.com/jameslearningnote/%E8%B3%87%E6%96%99%E5%88%86%E6%9E%90-%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-%E7%AC%AC3-3%E8%AC%9B-%E7%B7%9A%E6%80%A7%E5%88%86%E9%A1%9E-%E9%82%8F%E8%BC%AF%E6%96%AF%E5%9B%9E%E6%AD%B8-logistic-regression-%E4%BB%8B%E7%B4%B9-a1a5f47017e5
4. 機器學習:如何在多類別分類問題上使用用二元分類器進行分類(Multiclass Strategy for Binary classifier)
   https://medium.com/@chih.sheng.huang821/%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-%E5%A6%82%E4%BD%95%E5%9C%A8%E5%A4%9A%E9%A1%9E%E5%88%A5%E5%88%86%E9%A1%9E%E5%95%8F%E9%A1%8C%E4%B8%8A%E4%BD%BF%E7%94%A8%E7%94%A8%E4%BA%8C%E5%85%83%E5%88%86%E9%A1%9E%E5%99%A8%E9%80%B2%E8%A1%8C%E5%88%86%E9%A1%9E-multiclass-strategy-for-binary-classifier-b4e5017202ff
5. Class Handout, Lee, Chia Jung professor, MDM64001, School of Big Data Management, Soochow University