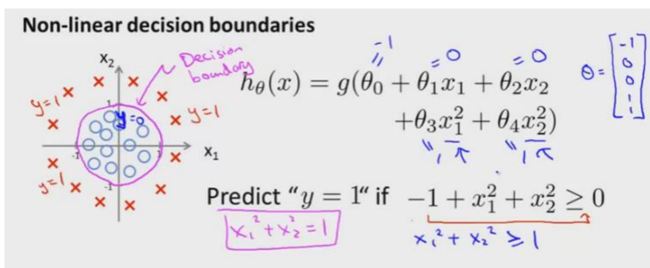
1 Logistic Regression

1.1 What is Logistic Regression?

Logistic regression is a classification algorithm, such as binary classification, multiclass classification, etc. The decision boundary can be linear or nonlinear. **It deals with discrete distributions of results while linear regression deals with continuous distributions**.

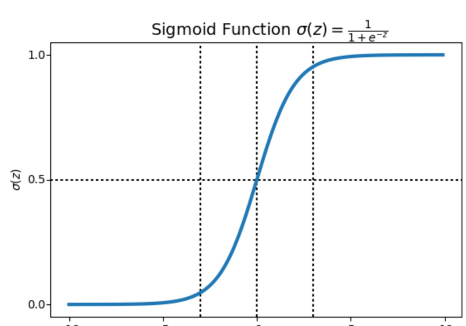


Let's say a binary classification algorithm, it's to solve the probability that y = 1 given x and θ:

$$h_\theta(x) = P(y = 1 \mid x; \theta) \tag{1}$$

1.2 Sigmoid Function

The most common way to do Logistic Regression is to map the output of linear regression **(Value)** to a range (0, 1) **(Probability)** by a sigmoid function.



$$z = \theta^T x \xrightarrow{g(z)=\frac{1}{1+e^{-z}}} h_\theta(x) = \frac{1}{1+e^{-\theta^T x}} \tag{2}$$

1.3 Prediction Model

$$\boldsymbol{h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}} \tag{3}$$

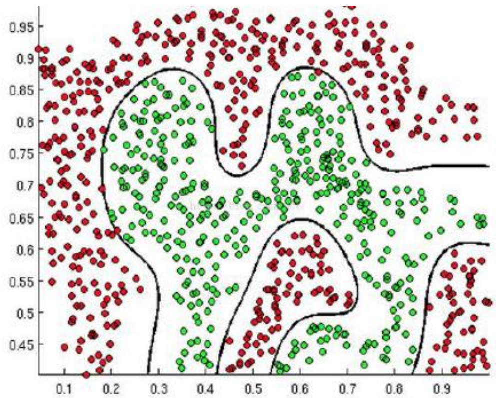
For a binary classification,

$$P(y = 1 \mid x; \theta) = h_\theta(x) \tag{4}$$

$$P(y = 0 \mid x; \theta) = 1 - h_\theta(x) \tag{5}$$

Eq. (4) and Eq. 5 can be merged as Eq. (6):

$$\boldsymbol{P(y \mid x; \theta) = (h_\theta(x))^y \, (1 - h_\theta(x))^{1-y}} \tag{6}$$



1.4 Maximum Likelihood Estimation

$$L(\theta) = \prod_{i=1}^{m} P\left(y^{(i)} \mid x^{(i)}; \theta\right) = \left(h_\theta\left(x^{(i)}\right)\right)^{y^{(i)}} \cdot \left(1 - h_\theta\left(x^{(i)}\right)\right)^{(1-y^{(i)})} \tag{7}$$

$$l(\theta) = log(\theta) = \sum_{i=1}^{m} \left[y^{(i)} log h_\theta\left(x^{(i)}\right) + (1 - y^{(i)}) log\left(1 - h_\theta\left(x^{(i)}\right)\right) \right] \tag{8}$$

Now, we need to get the maximum of $l(\theta)$, so we can use Gradient Descent to solve:

$$J(\theta) = -\frac{1}{m} l(\theta) \tag{9}$$

1.5 Solve Using Minimum Square Method

The derivative for sigmoid function:

$$\begin{aligned}\sigma(x)' &= \left(\frac{1}{1+e^{-x}}\right)' = \frac{-(1+e^{-x})'}{(1+e^{-x})^2} = \frac{-1' - (e^{-x})'}{(1+e^{-x})^2} = \frac{0-(-x)'(e^{-x})}{(1+e^{-x})^2} = \frac{-(-1)(e^{-x})}{(1+e^{-x})^2} = \frac{e^{-x}}{(1+e^{-x})^2} \\ &= \left(\frac{1}{1+e^{-x}}\right)\left(\frac{e^{-x}}{1+e^{-x}}\right) = \sigma(x)\left(\frac{+1-1+e^{-x}}{1+e^{-x}}\right) = \sigma(x)\left(\frac{1+e^{-x}}{1+e^{-x}} - \frac{1}{1+e^{-x}}\right) \\ &= \sigma(x)(1-\sigma(x))\end{aligned}$$

Therefore, we can get the partial derivative:

$$\begin{aligned}\frac{\partial J(\theta)}{\partial \theta_j} &= -\frac{1}{m}\frac{\partial}{\partial \theta_j} l(\theta) \\ &= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\frac{1}{h_\theta(x^{(i)})}\frac{\partial}{\partial \theta_j}h_\theta\left(x^{(i)}\right) - (1-y^{(i)})\frac{1}{1-h_\theta(x^{(i)})}\frac{\partial}{\partial \theta_j}h_\theta\left(x^{(i)}\right)\right] \\ &= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\frac{1}{g(\theta^T x^{(i)})} - (1-y^{(i)})\frac{1}{1-g(\theta^T x^{(i)})}\right]\frac{\partial}{\partial \theta_j}g\left(\theta^T x^{(i)}\right) \\ &= -\frac{1}{m}\sum_{i=1}^{m}\left[\frac{y^{(i)}}{g(\theta^T x^{(i)})} - \frac{1-y^{(i)}}{1-g(\theta^T x^{(i)})}\right]g\left(\theta^T x^{(i)}\right)\left(1-g\left(\theta^T x^{(i)}\right)\right)\frac{\partial}{\partial \theta_j}\theta^T x^{(i)} \\ &= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\left(1-g\left(\theta^T x^{(i)}\right)\right) - (1-y^{(i)})g\left(\theta^T x^{(i)}\right)\right]x_j^{(i)} \\ &= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} - g\left(\theta^T x^{(i)}\right)\right]x_j^{(i)} \\ &= \frac{1}{m}\sum_{i=1}^{m}\left[g\left(\theta^T x^{(i)}\right) - y^{(i)}\right]x_j^{(i)}\end{aligned} \tag{10}$$

You can also write it as a matrix format:

$$\nabla J(\theta) = \frac{1}{m}X^T(g(X\theta) - y) \tag{11}$$

Note: Actually, you don’t have to focus on the position of i and j. For example, It’s totally okay with which one is up, which is down. More importantly, we should understand the meaning of i and j.

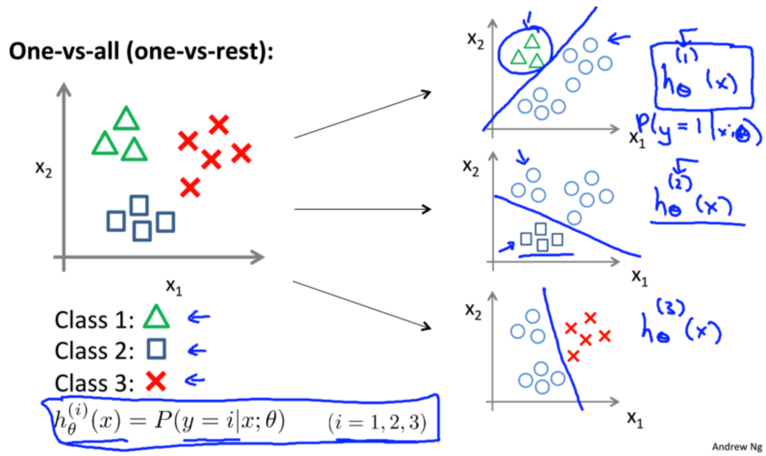
1.6 Solve Using Gradient Descent

After the gradient is solved, the gradient descent method can be used to update θ:

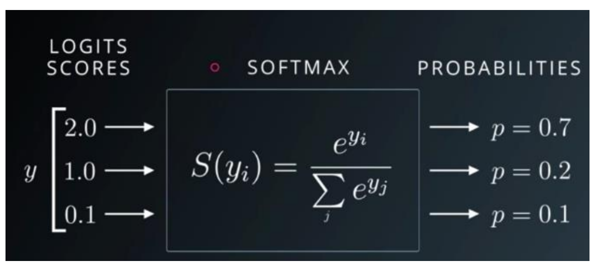
$$\theta_j := \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}\left[h_\theta\left(x^{(i)}\right) - y^{(i)}\right]x_i^j \tag{12}$$

1.7 Multiple Classification

When there are more than two categories, we extend y = {0,1} to y = {0,1...N}.Since y = {0,1...N}, we divide the problem into n+1 binary classification problem; In each case, we predict the probability that 'y' is a member of one of the classes. Finally, input x into n+1 classifier, and then take the maximum probability of n+1 classifier, that is, the probability of y = i.



$$h_\theta\left(x^{(i)}\right) = \begin{bmatrix} P\left(y^{(i)} = 1 \mid x^{(i)}; \theta\right) \\ P\left(y^{(i)} = 2 \mid x^{(i)}; \theta\right) \\ \vdots \\ P\left(y^{(i)} = k \mid x^{(i)}; \theta\right) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix} \tag{13}$$



1.8 Pros and Cons

Pros:

- simple implementation, widely used in industrial issues;
- easy computation, the speed is very fast, and the storage resources are low;
- convenient observation sample probability score.

Cons:

- it is easy to underfit, and the general accuracy is not too high
- can only handle two classification problems (softmax function derived from this can be used for multiple classification) and must be linearly separable;
- for nonlinear features, transformation is required.