17.4.17（逻辑回归）

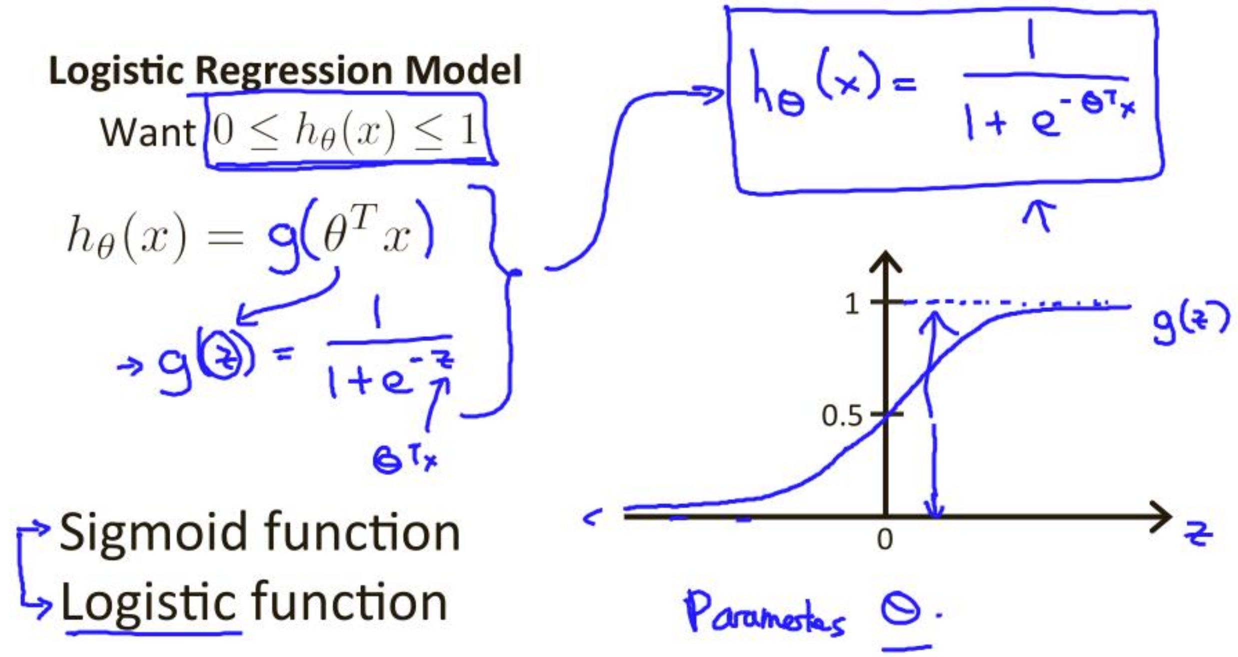
Logistic Regression （逻辑回归）虽然是回归算法，但在实际应用中大部分是用来Classification （分类）。分类两类的，一般是1: "Positive Class" 正类，和0: "Negative Class" 负类。

Threshold classifier output at 0.5:

If Hypothesis Representation大于等于0.5，predict "y = 1" 【Hypothesis Representation：假设model】

If Hypothesis Representation小于0.5 ，predict "y = 0"

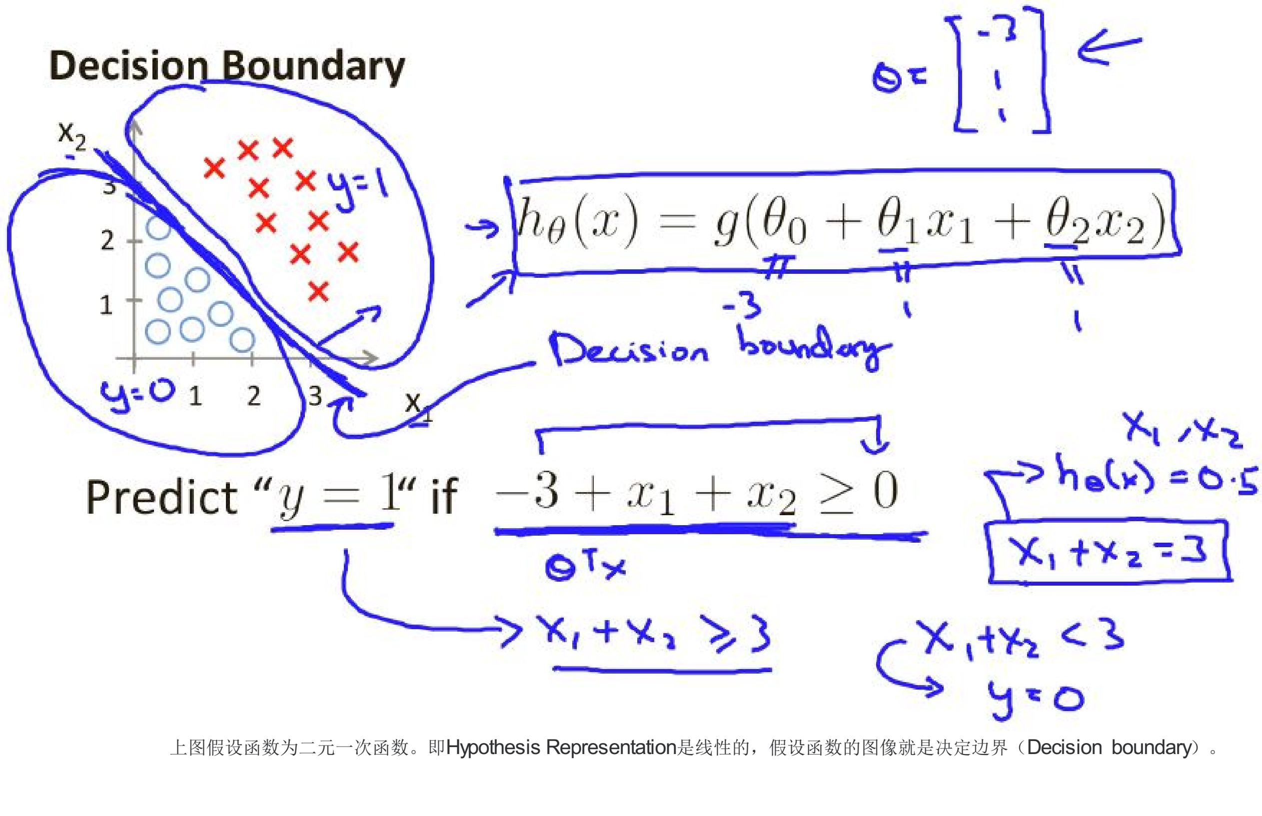
所以，二元的Logistic Regression Algorithm 中的Hypothesis Representation是在0-1 范围内。



图为：逻辑回归模型

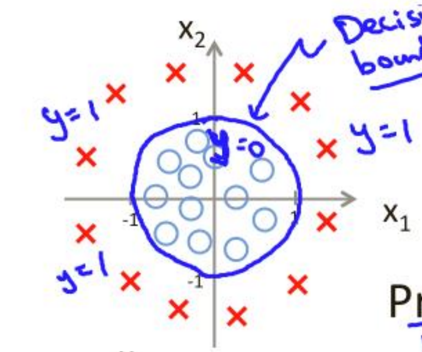
二元Logistic Regression Algorithm 的Hypothesis Representation又叫做Sigmoid function（S型函数）或者Logistic function （逻辑函数）。

Hypothesis Representation输出的是在X条件下达到Y=1的概率。



上图假设函数为二元一次函数。即Hypothesis Representation是线性的，假设函数的图像就是决定边界（Decision boundary）。

Non-linear decision boundaries

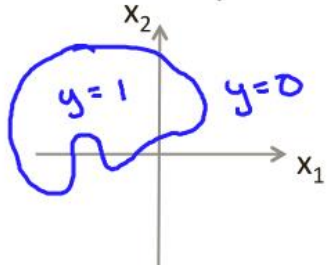


$$h_\theta(x) = g(\overset{=-1}{\theta_0} + \overset{=0}{\theta_1} x_1 + \overset{=0}{\theta_2} x_2 + \underset{=1}{\theta_3} x_1^2 + \underset{=1}{\theta_4} x_2^2)$$

$$\theta = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Predict "$y = 1$" if $\underbrace{-1 + x_1^2 + x_2^2 \geq 0}_{x_1^2 + x_2^2 \geq 1}$

$\boxed{x_1^2 + x_2^2 = 1}$

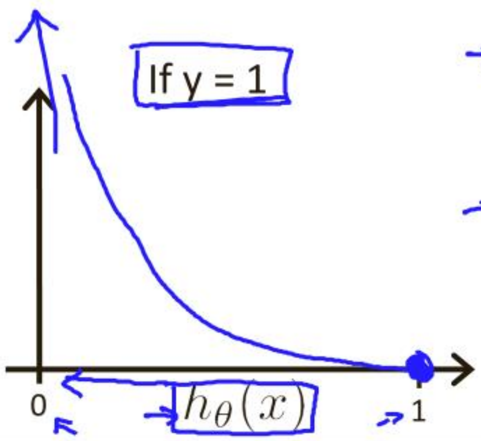


$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^2 x_2^2 + \theta_6 x_1^3 x_2 + \dots)$$

非线性的决定边界（Decision boundary），即Hypothesis Representation是多项式函数。假设theta的值。此时的决定边界构成一个圆。

Logistic regression cost function

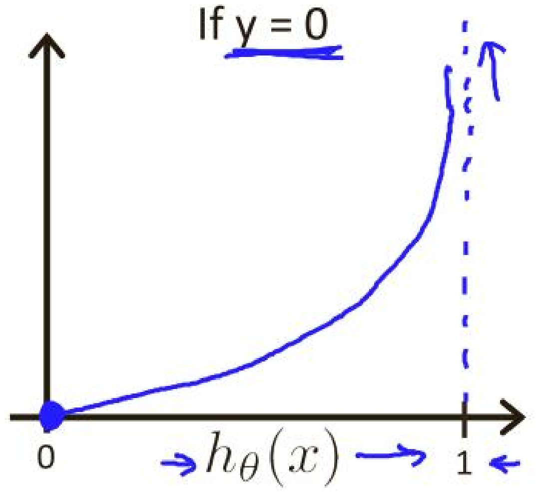
$$\text{Cost}(h_\theta(x), y) = \begin{cases} \boxed{-\log(h_\theta(x))} & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

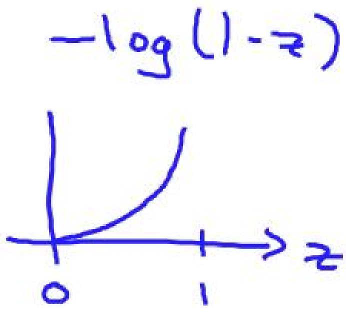


- Cost $= 0$ if $y = 1, h_\theta(x) = 1$
  But as $h_\theta(x) \to 0$
  $Cost \to \infty$
- Captures intuition that if $h_\theta(x) = 0$, (predict $P(y = 1|x; \theta) = 0$), but $\boxed{y = 1}$, we'll penalize learning algorithm by a very large cost.

Logistic regression cost function

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ \boxed{-\log(1 - h_\theta(x))} & \text{if } y = 0 \end{cases}$$





Logistic Regression（逻辑回归）算法的cost function。

前两张图是Logistic Regression（逻辑回归）算法的cost function的具体表现，这样构建的 Cost(h θ (x),y)函数的特点是：当实际的y=1 且 h θ 也为 1 时误差为 0，当 y=1，但 h θ 不为 1 时误差随着 h θ 的变小而变大；当实际的 y=0 且 h θ 也为 0 时代价为 0，当y=0，但 h θ 不为 0 时误差随着 h θ 的变大而变大。

Logistic regression cost function

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

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

To fit parameters θ :

$\boxed{\min_\theta J(\theta) \quad \text{Get } \Theta}$

To make a prediction given new x:

Output $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$ $\quad p(y=1 \mid x;\theta)$

简化的cost function如上，不用额外判断Y的值。

Gradient Descent

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

Want $\min_\theta J(\theta)$:

$$\Theta = \begin{bmatrix} \Theta_0 \\ \Theta_1 \\ \Theta_2 \\ \vdots \\ \Theta_n \end{bmatrix}$$

for i=0 to n

Repeat {

$$\boxed{\theta_j := \theta_j - \alpha \sum_{i=1}^{m} \left(\boxed{h_\theta(x^{(i)})} - y^{(i)}\right) x_j^{(i)}}$$

(simultaneously update all θ_j)

}

$h_\theta(x) = \Theta^T x$

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

Algorithm looks identical to linear regression!

化简的逻辑回归的梯度下降公式和线性回归的表面一样，但是假设函数不一样。

test.m | gradientDescentMulti.m | computeCostMulti.m | normalEqn.m | costFu

```
function [jVal, gradient] = costFunction(theta)
    jVal = (theta(1,1) - 6)^2 + (theta(2,1) - 6)^2;
    gradient(1,1) = 2*(theta(1,1) - 6);
    gradient(2,1) = 2*(theta(2,1) - 6);
end
```

命令行窗口

```
>> options = optimset('GradObj','on','MaxIter',66);
>> initialTheta = zeros(2,1);
>> [optTheta,minjVal,exitFlag] = fminunc(@costFunction,initialTheta,options)
```

高级算法求cost function 的最优值，optimset 是内置的算法结构体，fminunc是优化cost function最小值的函数。高级的，一般情况下可以替代梯度下降。