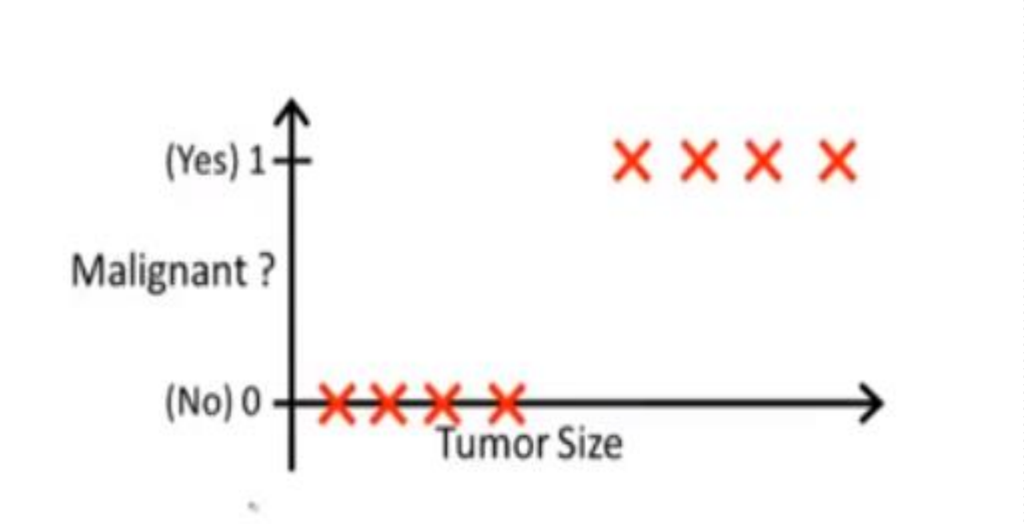
Logistic回归

韩雅妮

XiDian University

将线性回归应用到分类问题上存在的问题：



用直线对数据进行拟合，会出现$h_\theta(x)>1$或者$h_\theta(x)<0$，预测值超出范围。

Logistic回归

Logistic回归是一种分类算法，用于预测值为离散值0或1的情况。

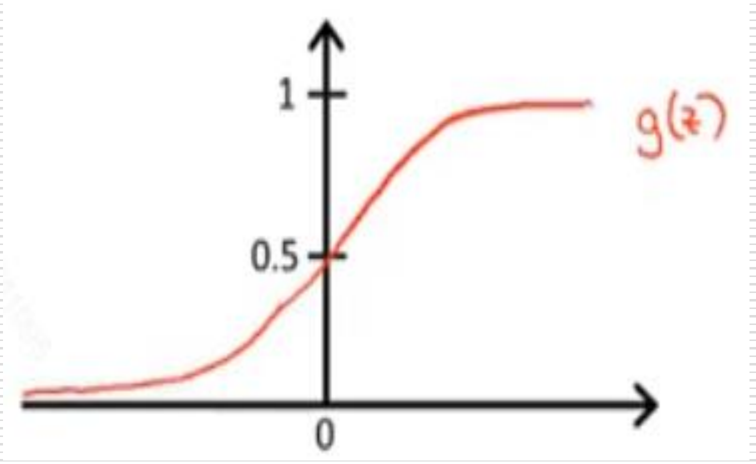
Logistic回归模型：

希望 $0 \le h_\theta(x) \le 1$

定义：$h_\theta(x) = g(\theta^T x)$

$$g(z) = \frac{1}{1+e^{-z}}$$

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



g(z)为sigmoid函数或者logistic函数，也就是logistic回归的名字来源。

输出值的预测

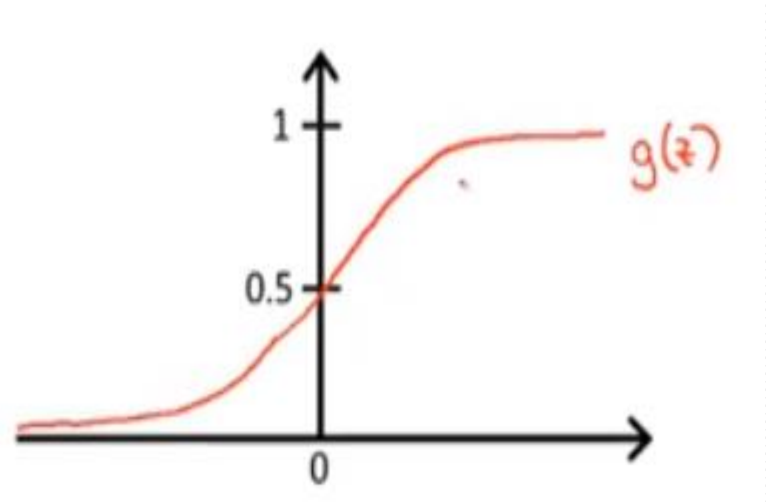
由sigmoid函数图像：

当z≥0时，g(z)≥0.5，预测输出为1。

所以，预测

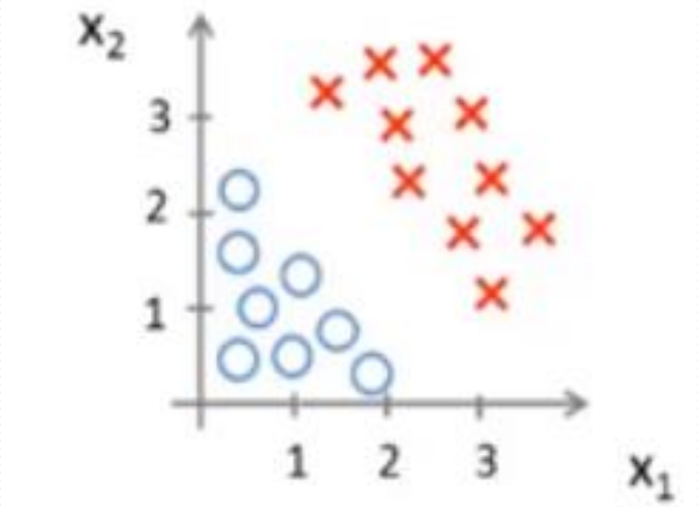
$h_{\theta}(x) \geq 0.5$，y=1;

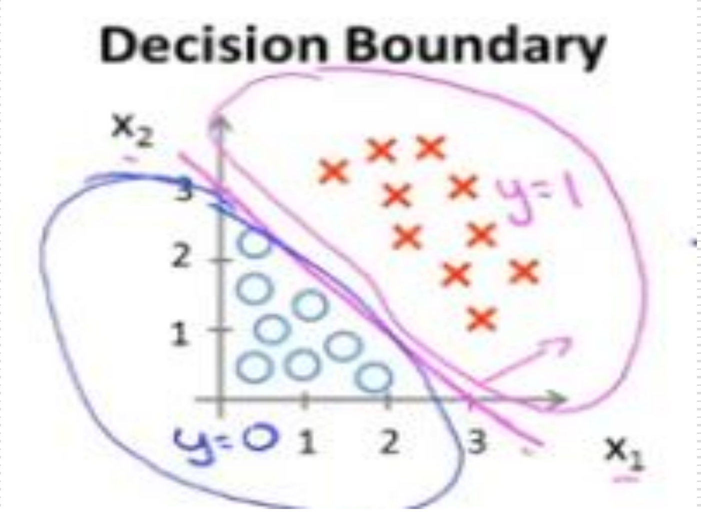
$h_{\theta}(x) < 0.5$，y=0.



线性决策边界

例：





$h_\theta(x) = g(\theta_0+\theta_1x_1+\theta_2x_2)$

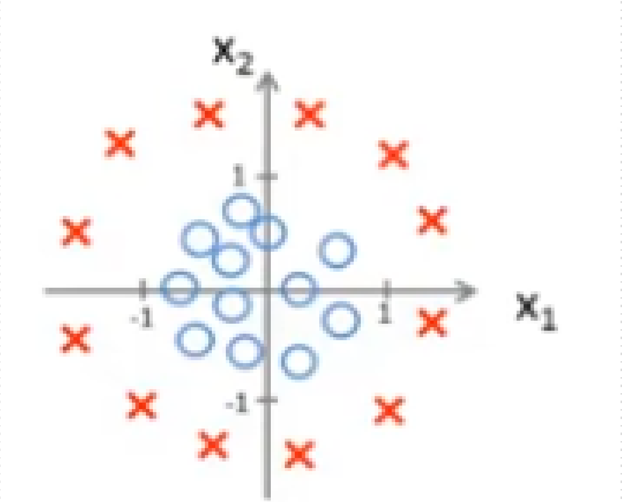
取θ_0=-3，θ_1=1，θ_2=1

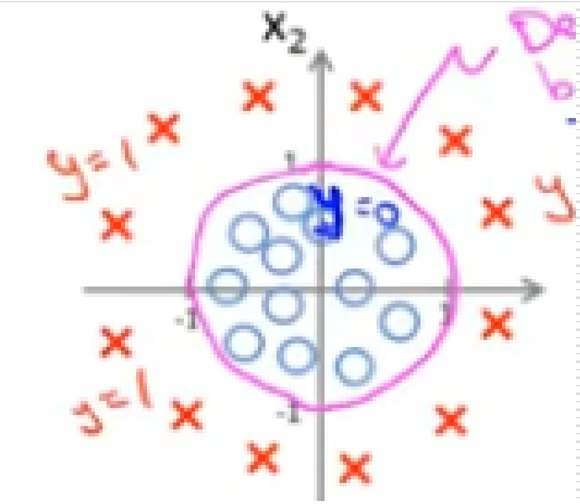
若$-3+x_1+x_2\geq 0$，预测y=1

x_1+x_2=3时，$h_\theta(x) = 0.5$，该函数作为决策边界。

非线性决策边界

例：





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

取$\theta_0=-1$，$\theta_1=0$，$\theta_2=0$，$\theta_3=1$，$\theta_4=1$

若$-1+x_1^2+x_2^2 \geq 0$，预测y=1

$x_1^2+x_2^2=1$时，$h_\theta(x) = 0.5$，y=1，该函数作为决策边界。

Logistic回归中的代价函数

有m个样本：$(x^{(1)}, y^{(1)}),\ldots,(x^{(m)}, y^{(m)})$

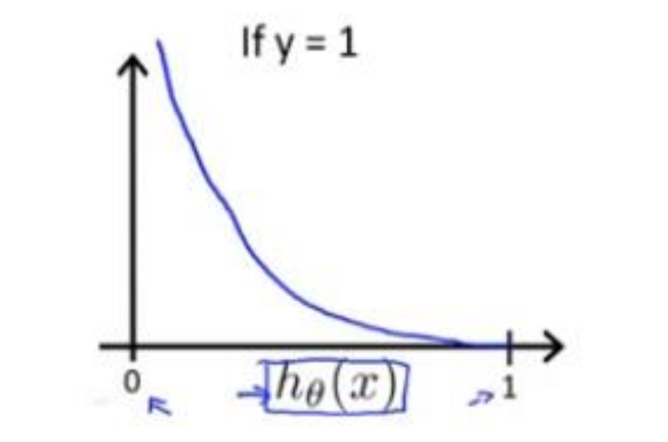
每个样本n+1个特征向量： $x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ \ldots \\ x_n^{(i)} \end{bmatrix}$

线性回归的代价函数：$J(\boldsymbol{\theta}) = \frac{\mathbf{1}}{\boldsymbol{m}} \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{m}} \frac{\mathbf{1}}{2} (\mathrm{h}_{\boldsymbol{\theta}}\mathrm{x}^{(\boldsymbol{i})} - \boldsymbol{y}^{(\boldsymbol{i})})^{\mathbf{2}}$

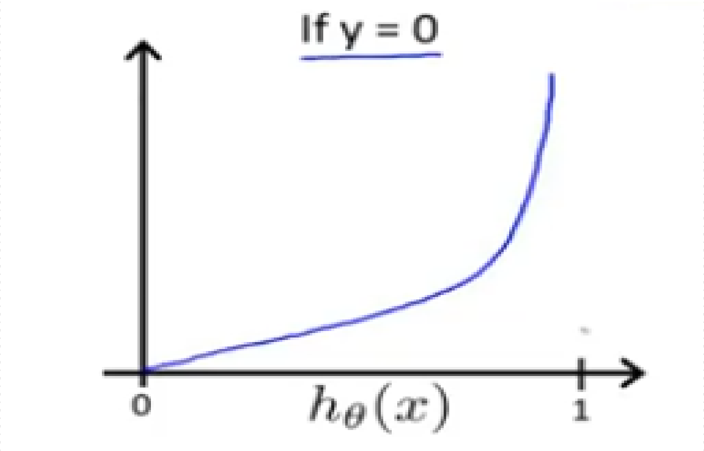
$= \frac{\mathbf{1}}{\boldsymbol{m}} \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{m}} \mathrm{cost}(\mathrm{h}_{\boldsymbol{\theta}}\mathrm{x}^{(\boldsymbol{i})}, \boldsymbol{y}^{(\boldsymbol{i})})$

因为$\boldsymbol{h_\theta}(\mathrm{x})$是非线性函数，代价函数非凸，使用梯度下降算法并不能使$J(\boldsymbol{\theta})$达到全局最优。因此，需要修改代价函数。

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



若$h_\theta(x)$=1, cost=0, y=1；
当$h_\theta(x)$趋于0时，cost趋于无穷，而实际y=1。



若$\boldsymbol{h_\theta}$(x)=0, cost=0, y=0；
当$\boldsymbol{h_\theta}$(x)趋于1时，cost趋于无穷，而实际y=0。

cost($\boldsymbol{h_\theta}$(x) ，y)= −ylog($\boldsymbol{h_\theta}$(x))−(1−y)log(1−$\boldsymbol{h_\theta}$(x))

代价函数：

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

Logistic回归中的参数

使用梯度下降算法：

repeat{

$$\boldsymbol{\theta}_{\mathrm{j}} := \boldsymbol{\theta}_{\mathrm{j}} - \boldsymbol{\alpha}\frac{\partial}{\partial\boldsymbol{\theta}_{\mathrm{j}}}\mathrm{J}(\boldsymbol{\theta})$$

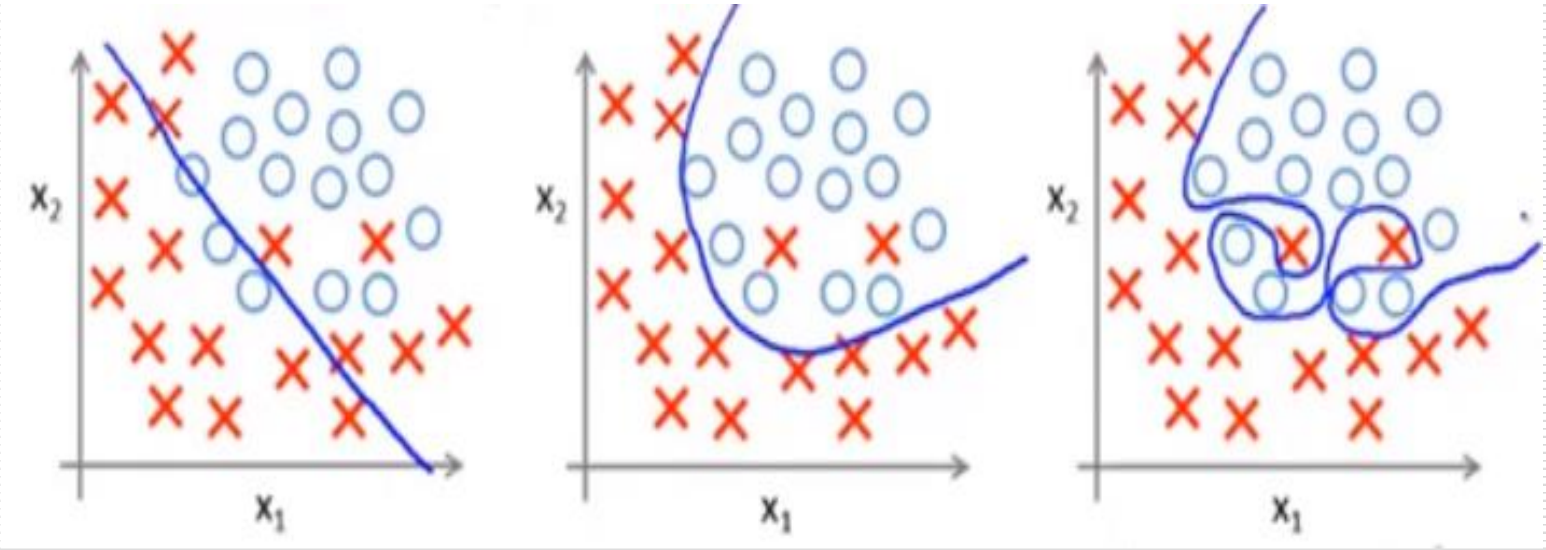
$$= \boldsymbol{\theta}_{\mathrm{j}} - \boldsymbol{\alpha}\frac{1}{\mathrm{m}}\sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{m}}(\mathrm{h}_{\boldsymbol{\theta}}\mathrm{x}^{(\boldsymbol{i})} - \boldsymbol{y}^{(\boldsymbol{i})})\,\mathrm{x}_{\boldsymbol{j}}^{(\boldsymbol{i})}$$

}

参数的更新与线性回归问题中的相同，但是线性回归中，$\boldsymbol{h}_{\boldsymbol{\theta}}(\mathrm{x})=\boldsymbol{\theta}^{\boldsymbol{T}}\mathrm{x}$，而**logistic**回归中，$\boldsymbol{h}_{\boldsymbol{\theta}}(\mathrm{x})=\dfrac{\mathbf{1}}{\mathbf{1}+\boldsymbol{e}^{-\boldsymbol{\theta}^{\boldsymbol{T}}\mathrm{x}}}$

正则化

将线性回归或者logistic回归应用到特定的机器学习中，可能会出现过度拟合的问题。



怎样避免过度拟合：

(1)减少变量的数量

a. 人工检查变量清单；

b. 模型选择算法，自动选择特征变量。

(2)正则化

思想：如果参数值较小，意味着假设的模型越简单，得到的函数越平滑，因此不容易出现过拟合问题。

线性回归的正则化

对线性回归的代价函数进行修改：

$$J(\boldsymbol{\theta}) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_{\theta}x^{(i)} - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

其中：$\lambda\sum_{j=1}^{n}\theta_j^2$为正则化项，$\lambda$为正则化参数，控制两个不同目标之间的平衡。

两个目标：(1)更好的拟合训练集；

(2)保持参数尽可能的小。

如果λ的取值过大，对参数的惩罚会更大，这些参数可能会趋于0，$h_\theta(x)=\theta_0$，相当于用一条直线来拟合数据。

1. 梯度下降算法：

$$\theta_0 := \theta_0 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta x^{(i)} - y^{(i)})\, x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha[\frac{1}{m}\sum_{i=1}^{m}(h_\theta x^{(i)} - y^{(i)})\, x_j^{(i)} + \frac{\lambda}{m}\theta_j\,]$$

$$= \theta_j\,(1-\frac{\alpha\lambda}{m}) - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta x^{(i)} - y^{(i)})\, x_j^{(i)}$$

因为α很小，而样本数m很大，因此$1-\frac{\alpha\lambda}{m}$比1略小，接近于1。在每次迭代时，乘以比1略小的数，每次把参数缩一点点，然后进行和之前一样的更新操作。

2. 正规方程

m个样本：$(x^{(1)}, y^{(1)}),\ldots, (x^{(m)}, y^{(m)})$

n个特征向量： $x^{(i)} = \begin{bmatrix} x_1^{(i)} \\ \ldots \\ x_n^{(i)} \end{bmatrix}$

构建 x = $\begin{bmatrix} (x^{(1)})^T \\ \ldots \\ (x^{(m)})^T \end{bmatrix}$, y = $\begin{bmatrix} y^{(1)} \\ \ldots \\ y^{(m)} \end{bmatrix}$

$$\theta = (x^T x + \lambda \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix})^{-1} x^T y$$

逻辑回归的正则化

代价函数：

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\left(y^{(i)}\log\left(h_\theta\left(x^{(i)}\right)\right)+\left(1-y^{(i)}\right)\left(\log\left(1-h_\theta\left(x^{(i)}\right)\right)\right)\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

梯度下降算法：

$$\theta_0 := \theta_0 - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta x^{(i)} - y^{(i)}\right)x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha\left[\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta x^{(i)} - y^{(i)}\right)x_j^{(i)} + \frac{\lambda}{m}\theta_j\right]$$

Thanks