第 3 周

六、逻辑回归(Logistic Regression)

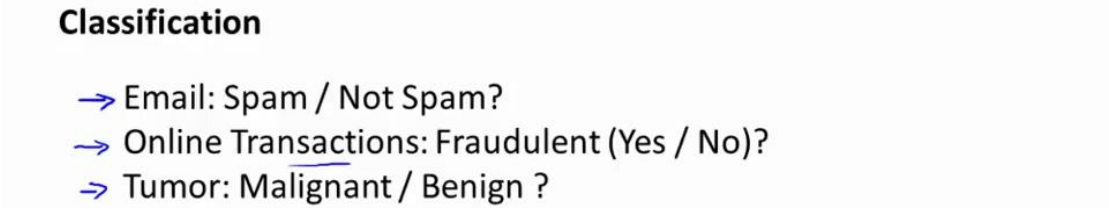
6.1 分类问题

参考文档: 6 - 1 - Classification (8 min).mkv

在这个以及接下来的几个视频中，开始介绍分类问题。

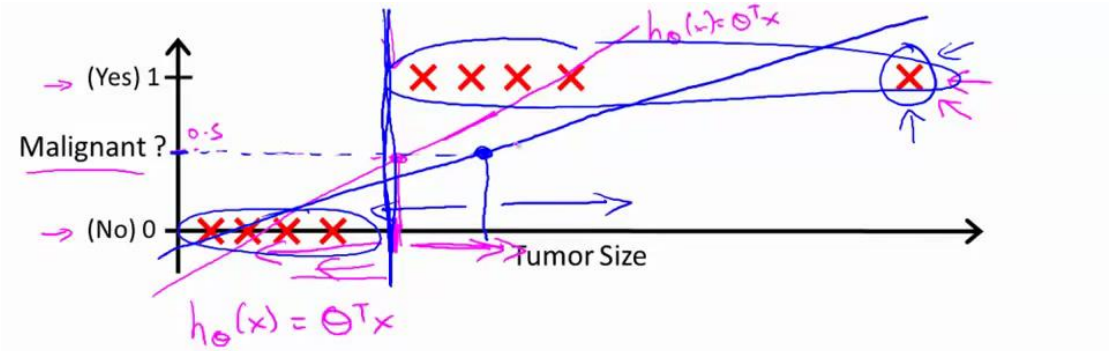
在分类问题中，你要预测的变量 y 是离散的值，我们将学习一种叫做逻辑回归 (Logistic Regression) 的算法，这是目前最流行使用最广泛的一种学习算法。

在分类问题中，我们尝试预测的是结果是否属于某一个类（例如正确或错误）。分类问题的例子有：判断一封电子邮件是否是垃圾邮件；判断一次金融交易是否是欺诈；之前我们也谈到了肿瘤分类问题的例子，区别一个肿瘤是恶性的还是良性的。



我们从二元的分类问题开始讨论。

我们将因变量(dependant variable)可能属于的两个类分别称为负向类（negative class）和正向类（positive class），则因变量 $y \in \{0,1\}$，其中 0 表示负向类，1 表示正向类。



→ Threshold classifier output $h_\theta(x)$ at 0.5:

→ If $h_\theta(x) \geq 0.5$, predict "y = 1"

If $h_\theta(x) < 0.5$, predict "y = 0"

Classification: y = 0 or 1

$h_\theta(x)$ can be > 1 or < 0

Logistic Regression: $0 \leq h_\theta(x) \leq 1$

如果我们要用线性回归算法来解决一个分类问题，对于分类，y 取值为 0 或者 1，但如果你使用的是线性回归，那么假设函数的输出值可能远大于 1，或者远小于 0，即使所有训练样本的标签 y 都等于 0 或 1。尽管我们知道标签应该取值 0 或者 1，但是如果算法得到的值远大于 1 或者远小于 0 的话，就会感觉很奇怪。所以我们在接下来的要研究的算法就叫做逻辑回归算法，这个算法的性质是：它的输出值永远在 0 到 1 之间。

顺便说一下，逻辑回归算法是分类算法，我们将它作为分类算法使用。有时候可能因为这个算法的名字中出现了“回归”使你感到困惑，但逻辑回归算法实际上是一种分类算法，它适用于标签 y 取值离散的情况，如：1 0 0 1。

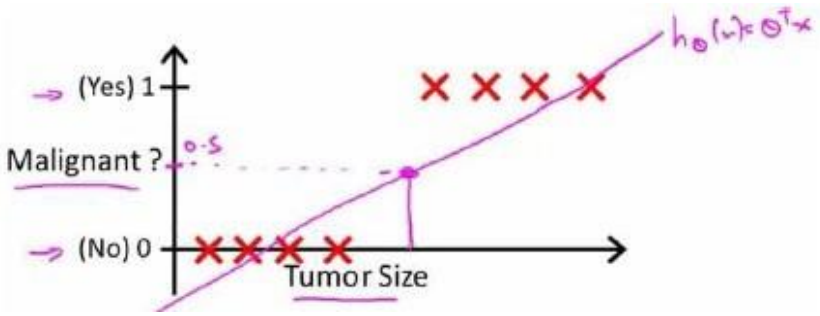
在接下来的视频中，我们将开始学习逻辑回归算法的细节。

6.2 假说表示

参考视频: 6 - 2 - Hypothesis Representation (7 min).mkv

在这段视频中，我要给你展示假设函数的表达式，也就是说，在分类问题中，要用什么样的函数来表示我们的假设。此前我们说过，希望我们的分类器的输出值在 0 和 1 之间，因此，我们希望想出一个满足某个性质的假设函数，这个性质是它的预测值要在 0 和 1 之间。

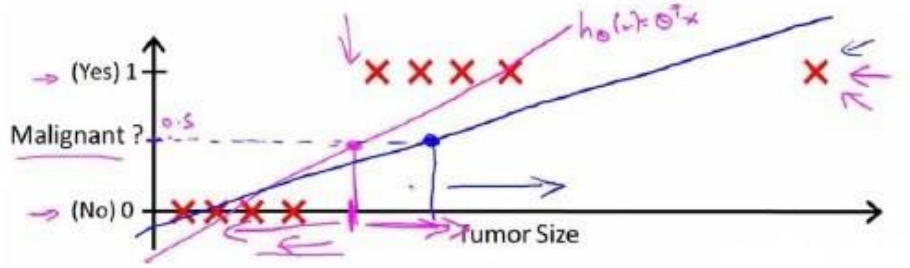
回顾在一开始提到的乳腺癌分类问题，我们可以用线性回归的方法求出适合数据的一条直线：



根据线性回归模型我们只能预测连续的值，然而对于分类问题，我们需要输出 0 或 1，我们可以预测：

当 h_θ 大于等于 0.5 时，预测 y=1。

当 h_θ 小于 0.5 时，预测 y=0 对于上图所示的数据，这样的一个线性模型似乎能很好地完成分类任务。假使我们又观测到一个非常大尺寸的恶性肿瘤，将其作为实例加入到我们的训练集中来，这将使得我们获得一条新的直线。



这时，再使用 0.5 作为阀值来预测肿瘤是良性还是恶性便不合适了。可以看出，线性回归模型，因为其预测的值可以超越[0,1]的范围，并不适合解决这样的问题。

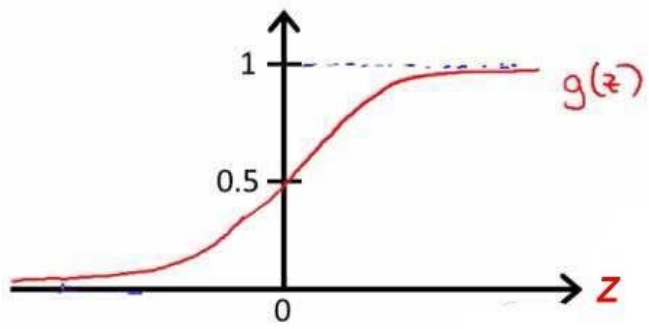
我们引入一个新的模型，逻辑回归，该模型的输出变量范围始终在 0 和 1 之间。 逻辑回归模型的假设是：$h_\theta(x)=g(\theta^T X)$

其中：

X 代表特征向量

g 代表逻辑函数（logistic function）是一个常用的逻辑函数为 S 形函数（Sigmoid function），公式为： $g(z)=\frac{1}{1+e^{-z}}$ 。

该函数的图像为：



合起来，我们得到逻辑回归模型的假设：

对模型的理解： $h_\theta(x)=\frac{1}{1+e^{-\theta^T x}}$

$h_\theta(x)$的作用是，对于给定的输入变量，根据选择的参数计算输出变量=1 的可能性（estimated probablity）即 $h_\theta(x)=P(y=1|x;\theta)$

例如，如果对于给定的 x，通过已经确定的参数计算得出 $h_\theta(x)=0.7$，则表示有 70%的几率 y 为正向类，相应地 y 为负向类的几率为 1-0.7=0.3。

6.3 判定边界

参考视频: 6 - 3 - Decision Boundary (15 min).mkv

现在讲下决策边界(decision boundary)的概念。这个概念能更好地帮助我们理解逻辑回归的假设函数在计算什么。

在逻辑回归中，我们预测：

当 h_θ 大于等于 0.5 时，预测 y=1

当 h_θ 小于 0.5 时，预测 y=0

根据上面绘制出的 S 形函数图像，我们知道当

z=0 时 g(z)=0.5

z>0 时 g(z)>0.5

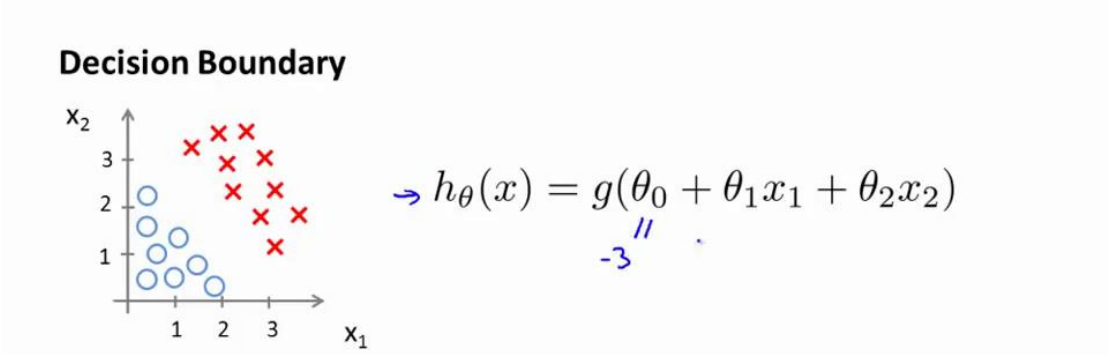
z<0 时 g(z)<0.5

又 $z=\theta^T X$，即：

$\theta^T X$ 大于等于 0 时，预测 y=1

$\theta^T X$ 小于 0 时，预测 y=0

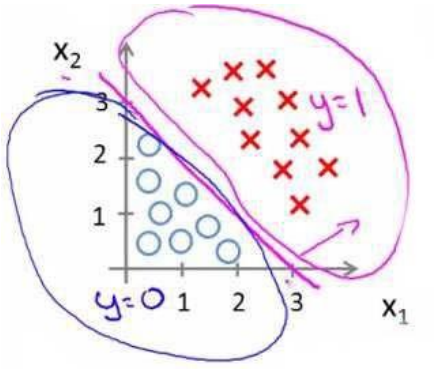
现在假设我们有一个模型：



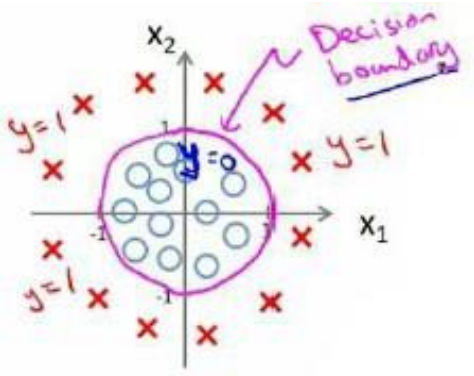
并且参数 θ 是向量[-3 1 1]。 则当$-3+x_1+x_2$ 大于等于 0，即 x_1+x_2 大于等于 3 时，模型将预测 y=1。

我们可以绘制直线 $x_1+x_2=3$，这条线便是我们模型的分界线，将预测为 1 的区域和预测

为 0 的区域分隔开。



假使我们的数据呈现这样的分布情况，怎样的模型才能适合呢？



因为需要用曲线才能分隔 y=0 的区域和 y=1 的区域，我们需要二次方特征： 假设参数：$h_\theta(x)=g\left(\theta_0+\theta_1x_1+\theta_2x_2+\theta_3x_1^2+\theta_4x_2^2\right)$是[-1 0 0 1 1]，则我们得到的判定边界恰好是圆点在原点且半径为 1 的圆形。

我们可以用非常复杂的模型来适应非常复杂形状的判定边界。

6.4 代价函数

参考视频: 6 - 4 - Cost Function (11 min).mkv

在这段视频中，我们要介绍如何拟合逻辑回归模型的参数 θ 。具体来说，我要定义用来拟合参数的优化目标或者叫代价函数，这便是监督学习问题中的逻辑回归模型的拟合问题。

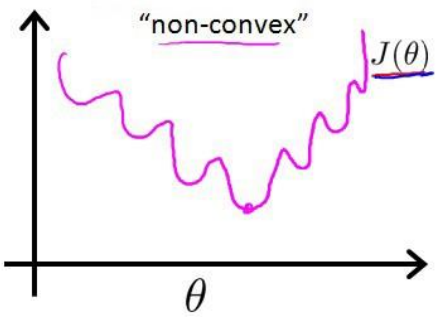
Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(m)}, y^{(m)})\}$

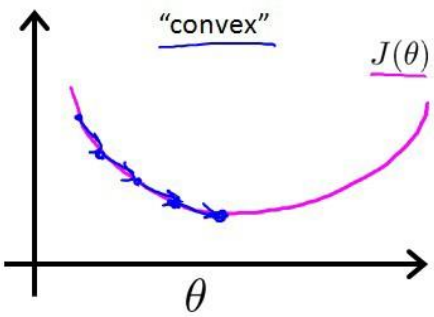
m examples $\quad x \in \begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_n \end{bmatrix} \quad x_0 = 1, y \in \{0, 1\}$

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

How to choose parameters θ ?

对于线性回归模型，我们定义的代价函数是所有模型误差的平方和。理论上来说，我们也可以对逻辑回归模型沿用这个定义，但是问题在于，当我们将 $h_\theta(x) = \frac{1}{1+e^{-\theta^T X}}$ 带入到这样定义了的代价函数中时，我们得到的代价函数将是一个非凸函数（non-convex function）。





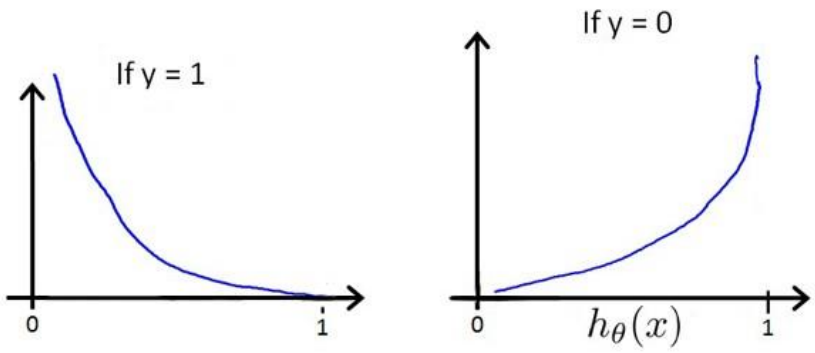
这意味着我们的代价函数有许多局部最小值，这将影响梯度下降算法寻找全局最小值。

线性回归的代价函数为： $J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$ 。

我们重新定义逻辑回归的代价函数为：$J(\theta)=\frac{1}{m}\sum_{i=1}^{m}\mathrm{Cost}\left(h_\theta\left(x^{(i)}\right),y^{(i)}\right)$，其中

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

$h_\theta(x)$与 $\mathrm{Cost}(h_\theta(x),y)$之间的关系如下图所示：



这样构建的 $\mathrm{Cost}(h_\theta(x),y)$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。

将构建的 $\mathrm{Cost}(h_\theta(x),y)$简化如下：

$$Cost(h_\theta(x),y)=-y\times log(h_\theta(x))-(1-y)\times log(1-h_\theta(x))$$

带入代价函数得到：

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

在得到这样一个代价函数以后，我们便可以用梯度下降算法来求得能使代价函数最小的参数了。算法为：

Repeat

$$\theta_j:=\theta_j-\alpha\frac{\partial}{\partial\theta_j}J(\theta)$$

(simultaneously update all)

}

求导后得到：

Repeat

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

(simultaneously update all)

}

在这个视频中，我们定义了单训练样本的代价函数，凸性分析的内容是超出这门课的范围的，但是可以证明我们所选的代价值函数会给我们一个凸优化问题。代价函数 J(θ)会是一个凸函数，并且没有局部最优值。

注：虽然得到的梯度下降算法表面上看上去与线性回归的梯度下降算法一样，**但是这里的 $h_\theta(x)=g(\theta^T X)$与线性回归中不同，所以实际上是不一样的**。另外，在运行梯度下降算法之前，进行特征缩放依旧是非常必要的。

一些梯度下降算法之外的选择： 除了梯度下降算法以外，还有一些常被用来令代价函数最小的算法，这些算法更加复杂和优越，而且通常不需要人工选择学习率，通常比梯度下降算法要更加快速。这些算法有：共轭梯度（Conjugate Gradient），局部优化法(Broyden fletcher goldfarb shann,BFGS)和有限内存局部优化法(LBFGS) fminunc 是 matlab 和 octave 中都带的一个最小值优化函数，使用时我们需要提供代价函数和每个参数的求导，下面是 octave 中使用 fminunc 函数的代码示例：

```
function [jVal, gradient] = costFunction(theta)
        jVal = [...code to compute
              J(theta)...];
        gradient = [...code to compute derivative of J(theta)...];
end
options = optimset('GradObj', 'on', 'MaxIter', '100');
initialTheta = zeros(2,1);
[optTheta, functionVal, exitFlag] = fminunc(@costFunction, initialTheta, options);
```

在下一个视频中，我们会把单训练样本的代价函数的这些理念进一步发展，然后给出整个训练集的代价函数的定义，我们还会找到一种比我们目前用的更简单的写法，基于这些推导出的结果，我们将应用梯度下降法得到我们的逻辑回归算法。

6.5 简化的成本函数和梯度下降

参考视频: 6 - 5 - Simplified Cost Function and Gradient Descent (10 min).mkv

在这段视频中，我们将会找出一种稍微简单一点的方法来写代价函数，来替换我们现在用的方法。同时我们还要弄清楚如何运用梯度下降法，来拟合出逻辑回归的参数。因此，听了这节课，你就应该知道如何实现一个完整的逻辑回归算法。

这就是逻辑回归的代价函数：

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

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

Note: $y = 0$ or 1 always

这个式子可以合并成：

$$Cost(h_\theta(x),y) = -y \times log(h_\theta(x)) - (1-y) \times log(1-h_\theta(x))$$

即，逻辑回归的代价函数：

$$\begin{aligned} J(\theta) &= \frac{1}{m}\sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)}) \\ &= -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))] \end{aligned}$$

根据这个代价函数，为了拟合出参数，该怎么做呢？我们要试图找尽量让 J(θ) 取得最小值的参数 θ。

$$\min_{\theta} J(\theta)$$

所以我们想要尽量减小这一项，这将我们将得到某个参数 θ。

如果我们给出一个新的样本，假如某个特征 x，我们可以用拟合训练样本的参数 θ，来输出对假设的预测。

另外，我们假设的输出，实际上就是这个概率值：p(y=1|x;θ)，就是关于 x 以 θ 为参数，y=1 的概率，你可以认为我们的假设就是估计 y=1 的概率，所以，接下来就是弄清楚如何最大限度地最小化代价函数 J(θ)，作为一个关于 θ 的函数，这样我们才能为训练集拟合出参数 θ。