1、 多变量线性回归(Linear Regression with Multiple Variables)

4.1 多维特征

参考视频: 4 - 1 - Multiple Features (8 min).mkv

目前为止，我们探讨了单变量/特征的回归模型，现在我们对房价模型增加更多的特征，例如房间数楼层等，构成一个含有多个变量的模型，模型中的特征为$(x_1, x_1, \ldots, x_n)$。

Size (feet2)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
...	...	...	...	...

$\boldsymbol{n}$ 代表特征的数量

$\boldsymbol{x^{(i)}}$代表第 i 个训练实例，是特征矩阵中的第i行，是一个**向量**（**vector**）。

比方说，上图的

$$x^{(2)} = \begin{bmatrix} 1416 \\ 3 \\ 2 \\ 40 \end{bmatrix},$$

$\boldsymbol{x_j^{(i)}}$代表特征矩阵中第 i 行的第 j 个特征，也就是第 i 个训练实例的第 j 个特征。

如上图的$x_2^{(2)} = 3, x_3^{(2)} = 2$，

支持多变量的假设 h 表示为：$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n$，

这个公式中有$n+1$个参数和n个变量，为了使得公式能够简化一些，引入$x_0 = 1$，则公式转化为：$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n$

此时模型中的参数是一个$n+1$维的向量，任何一个训练实例也都是$n+1$维的向量，特征矩阵X的维度是 $m * (n+1)$。 因此公式可以简化为：$h_\theta(x) = \theta^T X$，其中上标T代表矩阵转置。

$\rightarrow h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

For convenience of notation, define $x_0 = 1$. $(x_0^{(i)} = 1)$

$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1}$ $\quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \in \mathbb{R}^{n+1}$

$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n$ ($x_0 = 1$)

$= \theta^T x.$

θ^T: $(n+1) \times 1$ matrix

$\theta^T x$

x

4.2 多变量梯度下降

参考视频: 4 - 2 - Gradient Descent for Multiple Variables (5 min).mkv

与单变量线性回归类似，在多变量线性回归中，我们也构建一个代价函数，则这个代价函数是所有建模误差的平方和，即：$J(\theta_0,\theta_1\dots\theta_n)=\frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2$ ，

其中：$h_\theta(x)=\theta^T X=\theta_0+\theta_1x_1+\theta_2x_2+\dots+\theta_nx_n$ ，

我们的目标和单变量线性回归问题中一样，是要找出使得代价函数最小的一系列参数。多变量线性回归的批量梯度下降算法为：

Repeat {

$$\theta_j:=\theta_j-\alpha\frac{\partial}{\partial\theta_j}J(\theta_0,\theta_1,\dots,\theta_n)$$

}

即：

Repeat {

$$\theta_j:=\theta_j-\alpha\frac{\partial}{\partial\theta_j}\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})^2$$

}

求导数后得到：

Repeat {

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

(simultaneously update θ_j for j=0,1,...,n)

}

当$n>=1$时，

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

$$\theta_1:=\theta_1-a\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})\,x_1^{(i)}$$

$$\theta_2:=\theta_2-a\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})\,x_2^{(i)}$$

我们开始随机选择一系列的参数值，计算所有的预测结果后，再给所有的参数一个新的

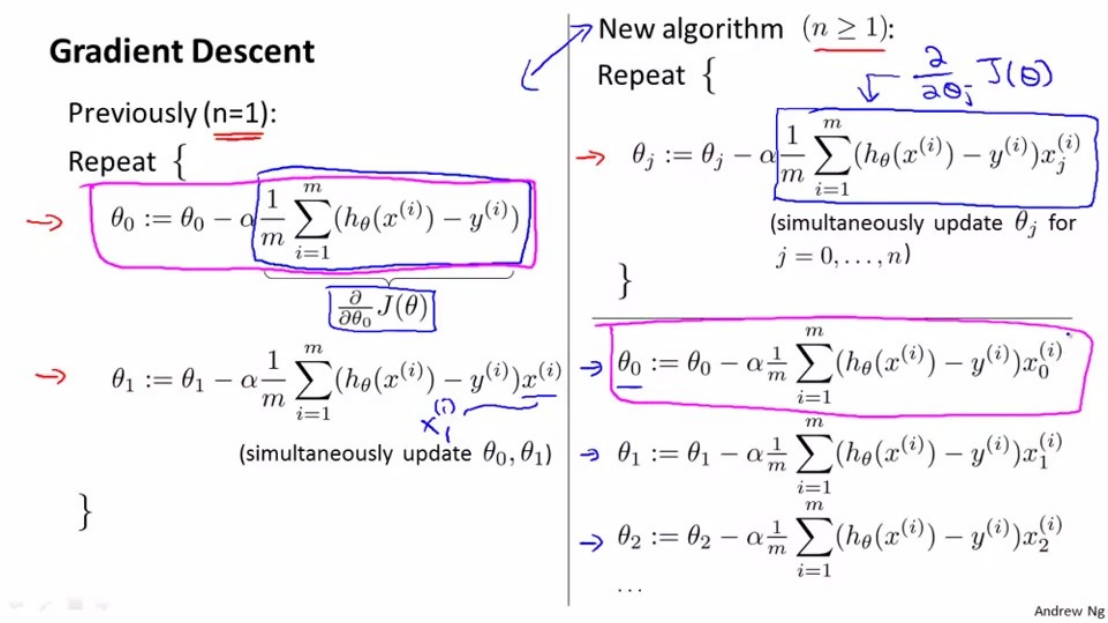
值，如此循环直到收敛。

代码示例：

计算代价函数 $J(\theta)=\frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2$ 其中：$h_\theta(x)=\theta^T X=\theta_0 x_0+\theta_1 x_1+\theta_2 x_2+\ldots+\theta_n x_n$

Python 代码：

```
def computeCost(X, y, theta):
    inner = np.power(((X * theta.T) - y), 2)
    return np.sum(inner) / (2 * len(X))
```

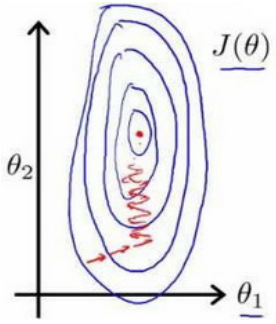


4.3 梯度下降法实践 1-特征缩放

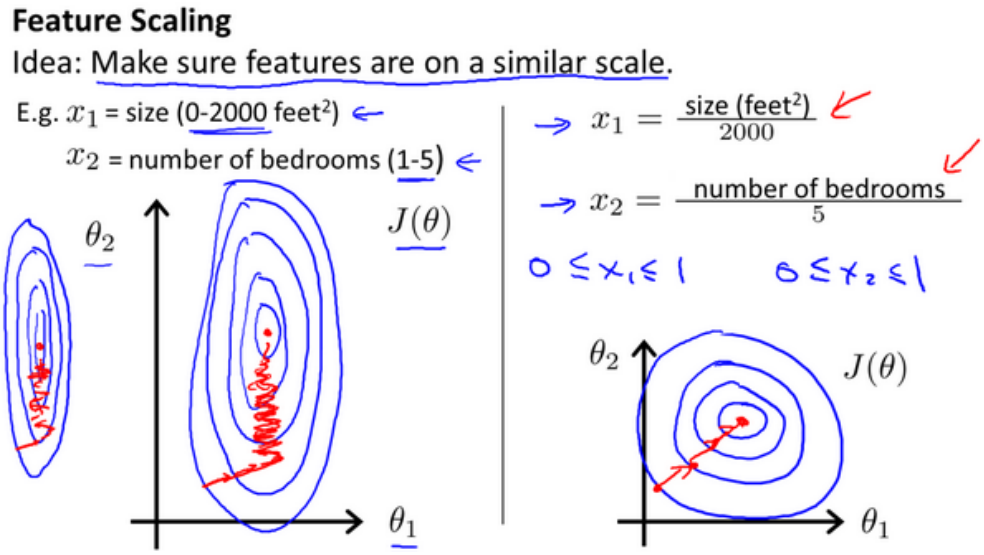
参考视频: 4 - 3 - Gradient Descent in Practice I - Feature Scaling (9 min).mkv

在我们面对多维特征问题的时候，我们要保证这些特征都具有相近的尺度，这将帮助梯度下降算法更快地收敛。

以房价问题为例，假设我们使用两个特征，房屋的尺寸和房间的数量，尺寸的值为 0-2000 平方英尺，而房间数量的值则是 0-5，以两个参数分别为横纵坐标，绘制代价函数的等高线图能，看出图像会显得很扁，梯度下降算法需要非常多次的迭代才能收敛。



解决的方法是尝试将所有特征的尺度都尽量缩放到-1 到 1 之间。如图：

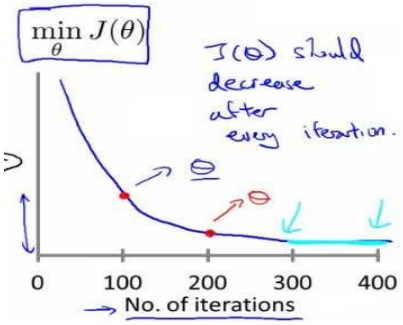


最简单的方法是令：$x_n = \frac{x_n - \mu_n}{s_n}$，其中 μ_n是平均值，s_n是标准差。

4.4 梯度下降法实践 2-学习率

参考视频: 4 - 4 - Gradient Descent in Practice II - Learning Rate (9 min).mkv

梯度下降算法收敛所需要的迭代次数根据模型的不同而不同，我们不能提前预知，我们可以绘制迭代次数和代价函数的图表来观测算法在何时趋于收敛。



也有一些自动测试是否收敛的方法，例如将代价函数的变化值与某个阀值（例如 0.001）进行比较，但通常看上面这样的图表更好。

梯度下降算法的每次迭代受到学习率的影响，如果学习率α过小，则达到收敛所需的迭代次数会非常高；如果学习率α过大，每次迭代可能不会减小代价函数，可能会越过局部最小值导致无法收敛。

通常可以考虑尝试些学习率：

$\alpha = 0.01$，0.03，0.1，0.3，1，3，10

4.5 特征和多项式回归

参考视频: 4 - 5 - Features and Polynomial Regression (8 min).mkv

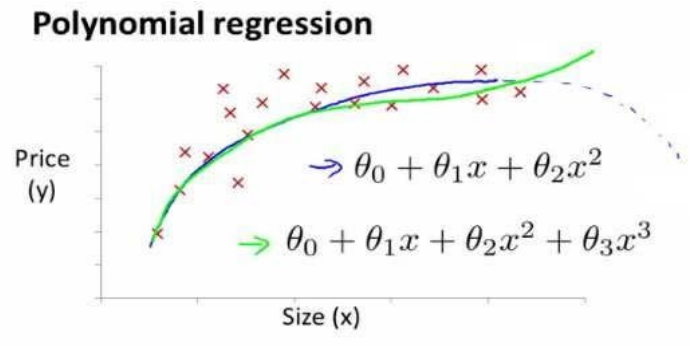
如房价预测问题，

$h_\theta(x) = \theta_0 + \theta_1 \times frontage + \theta_2 \times depth$

$x_1 = frontage$（临街宽度），$x_2 = depth$（纵向深度），$x = frontage * depth = area$（面积），则：$h_\theta(x) = \theta_0 + \theta_1 x$。

线性回归并不适用于所有数据，有时我们需要曲线来适应我们的数据，比如一个二次方模型：$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2^2$

或者三次方模型： $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2^2 + \theta_3 x_3^3$



通常我们需要先观察数据然后再决定准备尝试怎样的模型。 另外，我们可以令：

$x_2 = x_2^2, x_3 = x_3^3$，从而将模型转化为线性回归模型。

根据函数图形特性，我们还可以使：

$$h_\theta(x) = \theta_0 + \theta_1(size) + \theta_2(size)^2$$

或者:

$$h_\theta(x) = \theta_0 + \theta_1(size) + \theta_2\sqrt{size}$$

注：如果我们采用多项式回归模型，在运行梯度下降算法前，特征缩放非常有必要。