

多特征量(多元线性回归)

多特征量（多元线性回归）： Θ 元素有 $n+1$ 个， x 元素有 n 个，为了方便用矩阵表示方程式，增加一个 $x_0=1$ ，这样两个矩阵维数可以相同。将其中一个矩阵转置，就可以做矩阵乘法

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

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

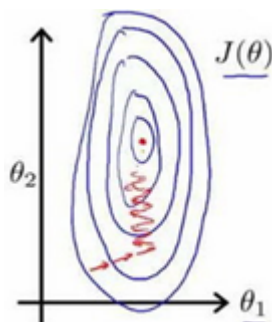
$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{m+1} \quad \Theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \in \mathbb{R}^{m+1}$$
$$h_{\theta}(x) = \theta_0 x_0 + \theta_1 x_1 + \dots + \theta_n x_n$$
$$= \Theta^T x$$

Θ^T is a $(n+1) \times 1$ matrix.

多元梯度下降法

特征缩放

当多个特征之间的取值范围相差较大时，他们的等值线形成的椭圆就会很扁平，梯度下降速度会减慢



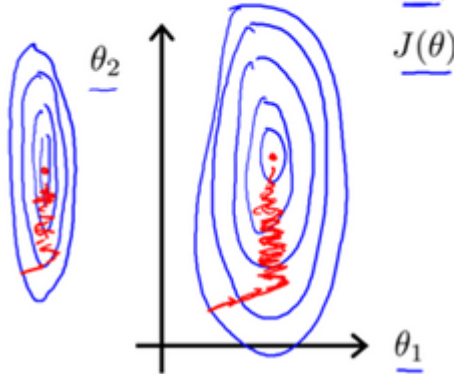
当特征之间的取值范围较小时，收敛的更快，所以为了更快的得到梯度下降的收敛结果，将特征的值进行相应的处理，以尽量保证特征值的取值范围在 $[-1,1]$ 。

Feature Scaling

Idea: Make sure features are on a similar scale.

E.g. $x_1 = \text{size (0-2000 feet}^2\text{)}$ ←

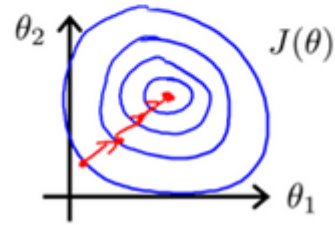
$x_2 = \text{number of bedrooms (1-5)}$ ←



$$\rightarrow x_1 = \frac{\text{size (feet}^2\text{)}}{2000}$$

$$\rightarrow x_2 = \frac{\text{number of bedrooms}}{5}$$

$$0 \leq x_1 \leq 1 \quad 0 \leq x_2 \leq 1$$



均值归一化操作:

用 $x_i - \mu_i$ 来替代 x_i 。下图的1000是平均值，2000是范围，即最大值减去最小值

Replace x_i with $x_i - \mu_i$ to make features have approximately zero mean (Do not apply to $x_0 = 1$).

E.g. $x_1 = \frac{\text{size} - 1000}{2000}$

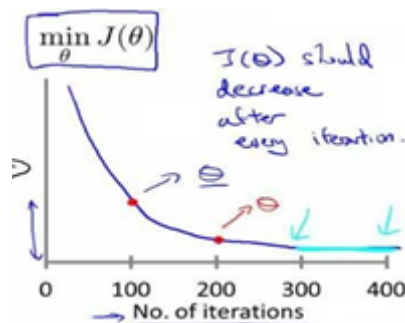
$$x_2 = \frac{\# \text{bedrooms} - 2}{5}$$

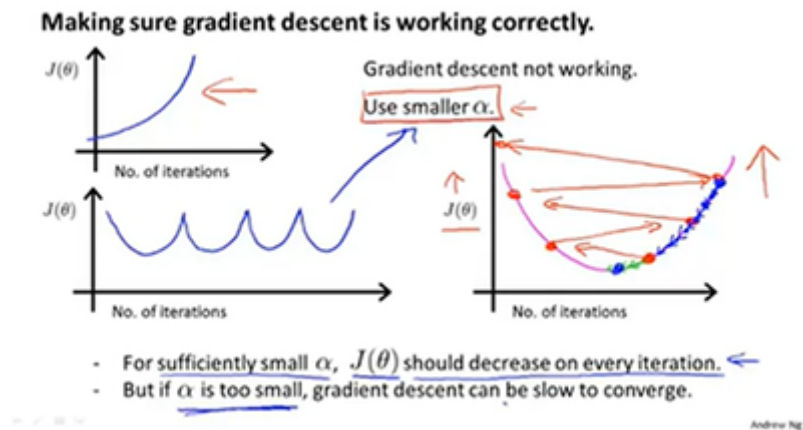
$$-0.5 \leq x_1 \leq 0.5, -0.5 \leq x_2 \leq 0.5$$

特征缩放不要求特别精确，只要尽量使各个特征的取值范围相近就可以了，只是为了使梯度下降的更快

学习率

下图：迭代次数和 $J(\theta)$ 最小值的关系。一般来说，随着迭代次数的不断增加 $J(\theta)$ 是在不断变小的，然后就收敛，趋于稳定。这个图可以看出这个算法是否正常（是否随着迭代次数增加， $J(\theta)$ 在减小），以及大概迭代多少次时， $J(\theta)$ 会收敛





如果是上图这两种情况，不断上升或者先下降在上升不断重复，一般是学习率 α 的值较大，往小更改一下。但是不要让 α 太小，如果太小的话，梯度下降的就会很慢

通常建议考虑的学习率：0.01,0.03,0.1,0.3,1,3,10.... 有一个十倍的关系和一个三倍的关系

特征和多项式回归

如房价预测问题，



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

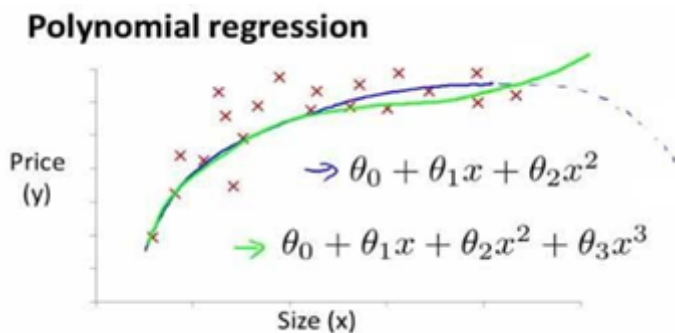
$x_1 = \text{frontage}$ (临街宽度)， $x_2 = \text{depth}$ (纵向深度)， $x = \text{frontage} * \text{depth} = \text{area}$ (面积)，则： $h_{\theta}(x) = \theta_0 + \theta_1 x$ 。

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：

$$h_{\theta}(x) = \theta_0 + \theta_1(size) + \theta_2 \llbracket (size) \rrbracket^2$$

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。