线性回归

线性回归方法，是指假定自变量与因变量之间呈线性关系，因此可以用因变量乘以权重或者参数去模拟自变量的值，这种方法就是线性回归方法。用于模型因变量的模型就叫线性回归模型。或者也可以叫做线性回归方程。

0 引入

Living area (feet^2)	#bedrooms	Price (1000$s)
2104	3	400
1600	3	330
2400	3	369
1416	2	232
3000	4	540
⋮	⋮	⋮

如上图所示，我们可以假定房价为自变量h，居住区域的大小和卧室数量作为因变量x1,x2。

于是我们可以得到如下方程：

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

在这里，θ就是参数，也可以把它叫做权重。为了简化方程，我们假设x0等于1，所以有：

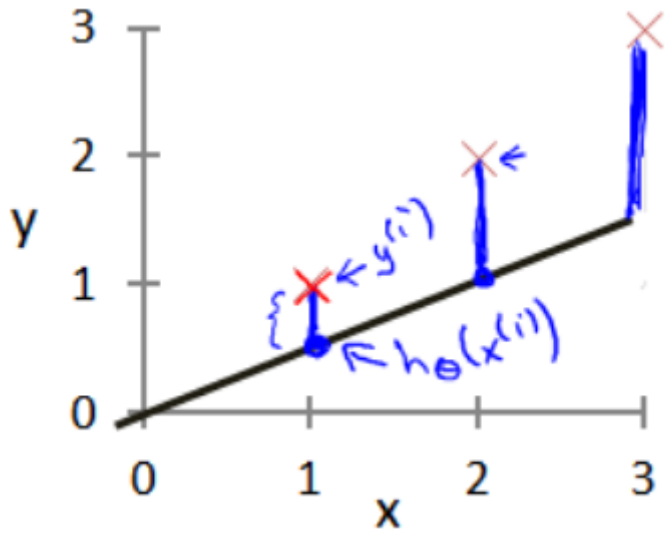
$$h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x$$

方程的右边就可以看做是两个向量，前一组是参数(权重)，后一组是变量(或者叫特征)。

1 代价函数

那么问题来了，在已经知道特征值和y值的情况下，我们如何去选择参数呢?

显然，选择的参数决定了我们得到的直线相对于我们的训练集的准确程度。参数确定后，模型所预测的值，与训练集中实际值之间的差距（下图中蓝线所指）就是建模误差（modeling error）。我们的任务是减少这种误差，这样我们的模型才能算好模型。

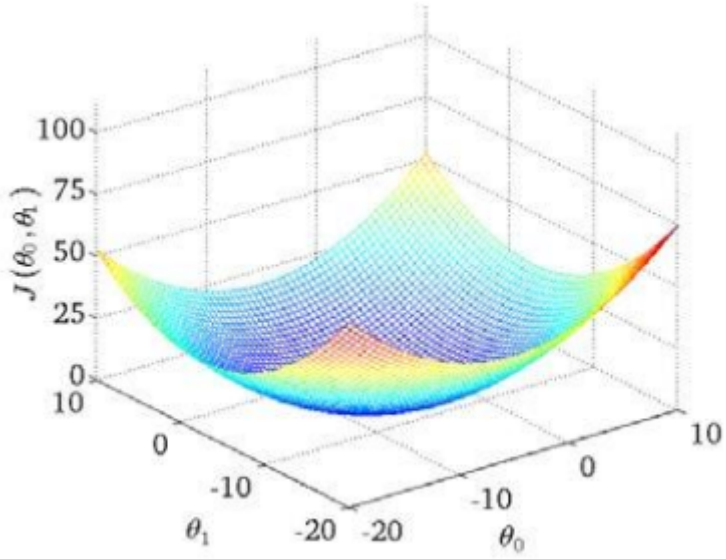


衡量此误差的函数就叫做代价函数 。我们的目标便是选择出，可以使得误差的平方和能够最小的模型参数。为什么要使用平方呢？是因为最小化的过程中涉及求导，平方函数更方便求导。

代价函数如下：

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

我们绘制一个等高线图，三个坐标分别为θ0和θ1和 J(θ0,θ1)：



如果我们随机得到了一组参数，这时我们就能知道代价函数的值。即上图曲面上的某一点，我们现在的问题是如何去改变参数，才能让代价最小呢？

2 梯度下降

答案是让参数沿着梯度的方向去下降就能让代价函数变小。因为从数学来说，梯度其实就是所谓的方向导数，也就是某个参数值对于整体函数的偏导数，沿着偏导数的方向相反的方向去变化，就是所谓的梯度下降。所有的参数在梯度下降的过程中会使得代价函数不断变小，直到收敛。

用数学符号来表示就是：

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

这里的α叫做学习率，右边部分就是针对参数j的偏导数，即参数j的梯度。参数j不停地减去梯度，就是所谓的梯度下降算法。α在这里的作用是控制梯度下降的速度，如果α过大会使得梯度下降跳过最优点，过小会使得优化过程减慢，需要实践过程中自行体会。

假设我们只有一个样本(x,y)，那么针对θ的梯度就是：

$$\begin{aligned}\frac{\partial}{\partial\theta_j}J(\theta) &= \frac{\partial}{\partial\theta_j}\frac{1}{2}(h_\theta(x)-y)^2 \\ &= 2\cdot\frac{1}{2}(h_\theta(x)-y)\cdot\frac{\partial}{\partial\theta_j}(h_\theta(x)-y) \\ &= (h_\theta(x)-y)\cdot\frac{\partial}{\partial\theta_j}\left(\sum_{i=0}^{n}\theta_i x_i - y\right) \\ &= (h_\theta(x)-y)\,x_j\end{aligned}$$

那么更新这个θ的方式就是：

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

注意这里变成了加号，是因为减去梯度后负负得正。

当样本不只一个而是很多是时，梯度下降的方式也随之改变。针对大量的样本，有两种梯度下降的方法。

批量梯度下降

$$\begin{aligned}&\text{Repeat until convergence } \{ \\ &\quad \theta_j := \theta_j + \alpha\textstyle\sum_{i=1}^{m}\left(y^{(i)} - h_\theta\left(x^{(i)}\right)\right)x_j^{(i)} \quad (\text{for every } j) \\ &\}\end{aligned}$$

由上面的更新公式我们可以看出，**我们每一次的参数更新都用到了所有的训练数据**（比如有m个，就用到了m个），如果训练数据非常多的话，**则非常耗时。**

随机梯度下降

$$\begin{aligned}&\text{Loop } \{ \\ &\quad \text{for i} = 1 \text{ to m}, \{ \\ &\quad\quad \theta_j := \theta_j + \alpha\left(y^{(i)} - h_\theta\left(x^{(i)}\right)\right)x_j^{(i)} \quad (\text{for every } j) \\ &\quad \ldots\ldots \} \\ &\}\end{aligned}$$

随机梯度下降是通过每个样本来迭代更新一次参数值，这样则会使得梯度下降的速度变快。但问题是针对一个样本来更新参数有可能走“歪路”。尽管随机梯度下降迭代的次数较多，在空间的搜索过程看起来很盲目。**但是大体上是往着最优值方向移动。**

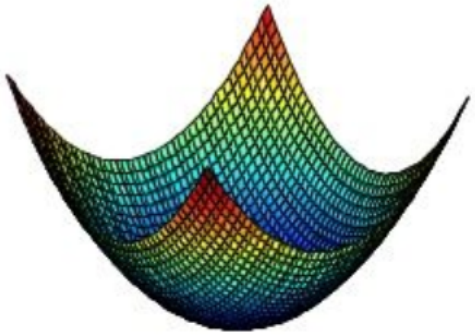
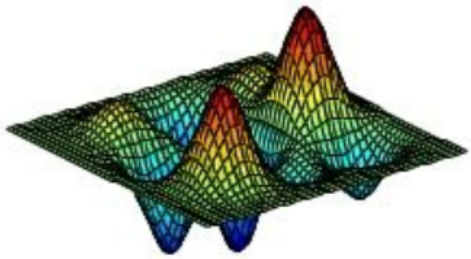
小批量梯度下降

从前两种梯度下降法可以看出，两种方法各自均有优缺点。那么在两种方法的之间取得一个折衷点，这就是小批量梯度下降，它的特点是：**算法的训练过程比较快，而且也能保证最终参数训练的准确率。**小批量梯度下降法中，每次更新参数的时候用到的样本数为n个，n<m。

注意点

当代价函数为凸函数时(下图左边)，梯度下降总是能取得全局最优解。

当代价函数为非凸函数时(下图右边)，梯度下降总是能取得局部最优解。

3 正则方程

所谓正则方程，就是用矩阵求导的方法去求解最佳参数，而不必依靠梯度下降那样一次一次去迭代。这里就涉及到了矩阵方程和矩阵求导。

首先我们给定样本的输入值：

$$X = \begin{bmatrix} -\left(x^{(1)}\right)^T - \\ -\left(x^{(2)}\right)^T - \\ \vdots \\ -\left(x^{(m)}\right)^T - \end{bmatrix}$$

然后再给定样本得输出值：

$$\vec{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

根据θ可以得到方程：

$$\begin{aligned} X\theta - \vec{y} &= \begin{bmatrix} \left(x^{(1)}\right)^T\theta \\ \vdots \\ \left(x^{(m)}\right)^T\theta \end{bmatrix} - \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix} \\ &= \begin{bmatrix} h_\theta\left(x^{(1)}\right) - y^{(1)} \\ \vdots \\ h_\theta\left(x^{(m)}\right) - y^{(m)} \end{bmatrix} \end{aligned}$$

我们知道：

$$z^T z = \sum_i z_i^2 :$$

所以有：

$$\begin{aligned} \frac{1}{2}(X\theta - \vec{y})^T(X\theta - \vec{y}) &= \frac{1}{2}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 \\ &= J(\theta) \end{aligned}$$

我们就可以根据这个矩阵方程来求解θ的导数：

$$
\begin{aligned}
\nabla_\theta J(\theta) &= \nabla_\theta \frac{1}{2}(X\theta - \vec{y})^T (X\theta - \vec{y}) \\
&= \frac{1}{2}\nabla_\theta \left(\theta^T X^T X\theta - \theta^T X^T \vec{y} - \vec{y}^T X\theta + \vec{y}^T \vec{y}\right) \\
&= \frac{1}{2}\nabla_\theta \operatorname{tr}\left(\theta^T X^T X\theta - \theta^T X^T \vec{y} - \vec{y}^T X\theta + \vec{y}^T \vec{y}\right) \\
&= \frac{1}{2}\nabla_\theta \left(\operatorname{tr}\theta^T X^T X\theta - 2\operatorname{tr}\vec{y}^T X\theta\right) \\
&= \frac{1}{2}\left(X^T X\theta + X^T X\theta - 2X^T \vec{y}\right) \\
&= X^T X\theta - X^T \vec{y}
\end{aligned}
$$

数学解释：标量$a = tr(a)$，因此可得第二行。第三行（1）对$\vec{y}^T\vec{y}$求θ导可得0；（2）$tr(A+B) = tr(A) + tr(B)$;(3) $-\theta^T X^T \vec{y} - \vec{y}^T X\theta = 2\vec{y}^T X\theta$；。第四行来源于(1)使用之前的迹的性质，其中$A^T = \theta$， $B = B^T = X^T X$， $C = I$;(2) $\bigtriangledown_A trAB = B^T$.

显然，令导数等于零就能得到使得函数最小化的参数θ

$$
\begin{aligned}
X^T X\theta &= X^T \vec{y} \\
\theta &= \left(X^T X\right)^{-1} X^T \vec{y}
\end{aligned}
$$

4 概率解释

为什么代价函数是最小均方误差模型呢？我们可以从概率的角度来推导得出。

首先我们知道输入和输出的关系为：

$$
y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)}
$$

对该模型的解释：

其中$\epsilon^{(i)}$是随机变量，它捕获噪声和非模型的因素。这通常是线性回归的概率模型。
我们还假设噪声是独立同分布$(i.i.d.)$，并来自高斯分布，高斯分布有均值为θ和任意方差σ^2。
因为$\epsilon^{(i)}$是高斯分布的随机变量，并且$\theta^T x^{(i)}$对于这个这个随机变量来说是常数。
向高斯随机变量添加常数，将使该变量的均值移动常数个数量，但它仍然是高斯分布。

可以推导出：

$$
p\left(\epsilon^{(i)}\right) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left(\epsilon^{(i)}\right)^2}{2\sigma^2}\right)
$$

This implies that

$$
p\left(y^{(i)}|x^{(i)};\theta\right) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left(y^{(i)}-\theta^T x^{(i)}\right)^2}{2\sigma^2}\right)
$$

当x已知并具有固定参数θ时，该函数可被视为y的函数。因此，我们可以称之为**似然函数likelihood function**：

$$
L(\theta) = L(\theta; X, \vec{y}) = p(\vec{y}|X;\theta)
$$

由于误差项服从独立同分布，所以有：

$$
\begin{aligned}
L(\theta) &= \prod_{i=1}^{m} p\left(y^{(i)}|x^{(i)};\theta\right) \\
&= \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right)
\end{aligned}
$$

我们需要找到满足以下条件的θ：选定 θ 的情况下，基于给定x，y的概率要最大化。 我们称之为**最大似然法。**为简化运算，我们将其化为**最大对数似然(log likelihood)**：

$$
\begin{aligned}
\ell(\theta) &= \log L(\theta) \\
&= \log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right) \\
&= \sum_{i=1}^{m} \log \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right) \\
&= m \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2} \cdot \frac{1}{2} \sum_{i=1}^{m} \left(y^{(i)} - \theta^T x^{(i)}\right)^2
\end{aligned}
$$

显然：

$$
\begin{gathered}
\text{最大化 } \ell(\theta) \text{ 就相当于最小化以下方程} \\
\frac{1}{2} \sum_{i=1}^{m} \left(y^{(i)} - \theta^T x^{(i)}\right)^2
\end{gathered}
$$

这恰好就是之前最小均方误差的代价函数。所以，这意味着我们用概率的方式证明了我们在最小均方算法中所得的结果。