斯坦福大学公开 Machine Learning 学习笔记

第二章 多值线性回归（Linear Regression with Multiple variables）

之前我们已经说过，多变量特征值的现行回归就是特征值不止一个的线性回归，我们先学习单值线性回归的目的主要是其直观性，有助于我们学习理解多值线性回归，两个的原理完全相同，也即单值线性回归就是 n=1 情况下的多值线性回归。

1.多变量特征值的假设函数

其实在第一章的内容中介绍假设函数都给出了多变量特征值的假设函数

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n \quad \text{设 } X_0=1 \qquad \text{(公式 1-1)}$$

令 $X=\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$, $\Theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$，则矩阵化后公式为：

$$h_\theta(x) = \Theta^T X \qquad \text{(公式 1-2)}$$

矩阵化后，对于公式的结果计算我们可以借助一些矩阵化公式，如 Matlab，快速计算矩阵运算，如矩阵的乘或加。

2. 多变量特征值的代价函数和使用梯度下降算法的多值线性回归

代价函数和算法其实和上一章单值回归的没多大什么差别，现直接给出多值线性回归的算法描述，其中也给出了其代价函数的形式。

1. 获取训练集
2. 给出假设函数：

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n \qquad \text{(公式 1-3)}$$

3. 给出假设函数的代价函数：

$$J(\theta_0, \theta_1, \theta_2, \dots, \theta_n) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)^2 \qquad \text{(公式 1-4)}$$

4. 求得 $J(\theta_0, \theta_1, \theta_2, \dots, \theta_n)$ 对 θ_j 的偏导：

$$\frac{\partial}{\partial\theta_j}J(\theta_0, \theta_1, \dots, \theta_n) = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)\cdot x_j$$

$$= \frac{1}{m}\sum_{i=1}^{m}(\theta_0 + \theta_1 x_1^i + \dots + \theta_n x_n^i - y_i)\cdot x_j^i \qquad \text{(公式 1-5)}$$

上式也等于下面的公式，θ_0 和其他 θ_j 不同的原因是对应的 x_0^i 都是常数 1，故忽略掉了：

$$\frac{\partial}{\partial\theta_0}J(\theta_0,\theta_1,\dots,\theta_n)=\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x_i)-y_i)$$

$$\frac{\partial}{\partial\theta_1}J(\theta_0,\theta_1,\dots,\theta_n)=\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x_i)-y_i)\cdot x_1^i$$

…

$$\frac{\partial}{\partial\theta_n}J(\theta_0,\theta_1,\dots,\theta_n)=\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x_i)-y_i)\cdot x_n^i$$

4. 循环执行下面的更新，直到代价函数达到最优解，即算法拟合为止。必须同时更新所有的θ_j的值：

$$\theta_j:=\theta_j-\alpha\frac{\partial}{\partial\theta_j}J(\theta_0,\theta_1,\dots,\theta_n) \quad \text{(公式 1-6)}$$

上式等价于两步：

$$\theta_0:=\theta_0-\alpha\frac{\partial}{\partial\theta_0}J(\theta_0,\theta_1,\dots,\theta_n)=\theta_0-\frac{\alpha}{m}\sum_{i=1}^{m}(h_\theta(x_i)-y_i)$$

$$\theta_1:=\theta_1-\alpha\frac{\partial}{\partial\theta_1}J(\theta_0,\theta_1,\dots,\theta_n)=\theta_1-\frac{\alpha}{m}\sum_{i=1}^{m}(h_\theta(x_i)-y_i)\cdot x_1^i$$

…

$$\theta_n:=\theta_n-\alpha\frac{\partial}{\partial\theta_n}J(\theta_0,\theta_1,\dots,\theta_n)=\theta_n-\frac{\alpha}{m}\sum_{i=1}^{m}(h_\theta(x_i)-y_i)\cdot x_n^i$$

3.特征值缩放（Feature Scaling）

在预测房价的例子中，有些特征值的可以很大，比如房子的面积可以取到 100 以上，而有些值就比较小，如房子的房间数，一般范围在 1 到 10，这样在假设函数中，由于这些值大小范围的不同，我们很难将他们映射到一个数值范围内进行研究，如绘图。这个时候就需要对特征值进行缩放，从而可以将所有的特征值映射到一个范围内，一般范围是$-1\le x_i\le 1$。

均值归一化（Mean Normalization）是常用的特征值缩放方法，公式如下:

$$x_i=\frac{x_i-\mu_i}{S_i} \quad \text{(公式 1-7)}$$

其中，μ_i为所有x_i的均值，S_i为x_i中的最大值-x_i中的最小值，或者x_i中的标准差。

4.学习率α

之前我们已经给出学习率的定义和当学习率过大或过小时产生的后果。此处讨论如何选取学习率α。一般情况下α的从以下中尝试选取：…, 0.001, 0.01, 0.1, 1, …。其中我们可以使用学习曲线的表现来选择最合适的α。以上值只是建议的数值，也可以自行选取，如 0.05,0.5 等。

5.正规方程（Normal Equation）

正规方程可以一次直接计算出J(θ)最优值，并且使用该方程不需要将特征值均值归一化。

设有 n 个特征值的 m 个样本：$\left(\left(x^{(1)},y^{(1)}\right),\left(x^{(2)},y^{(2)}\right),\ldots,\left(x^{(m)},y^{(m)}\right)\right)$，$x^{(i)}=\begin{matrix}x_0^{(i)}\\x_1^{(i)}\\\vdots\\x_n^{(i)}\end{matrix}$，

$$X=\begin{matrix}(x^{(1)})^{\mathrm{T}}\\(x^{(2)})^{\mathrm{T}}\\\vdots\\(x^{(m)})^{T}\end{matrix}$$

$$\Theta=(X^TX)^{-1}X^Ty \qquad \text{(公式 1-8)}$$

下面给出线性回归算法和正规方程的比较。

线性回归	正规方程
需要选择α	没必要选择α
需要迭代很多次	不需要迭代很多次
即使 n（n>1000）很大时，任然运行很好	需要计算$(X^TX)^{-1}$
	n 很到时，运行很慢

6.正规方程不可逆（Noninvertibility）的情况

虽然正规方程在计算 n 比较小的特征值向量时很方便，但是使用正规方程有一个限制，我们从公式 1-8 中可以看出：X^TX必须是可逆的才能计算$(X^TX)^{-1}$，虽然一般的矩阵计算中都提供了伪逆矩阵。那么矩阵不可逆的时候我们应该考虑是否存在冗余的特征值。

个人理解可能存在偏差或错误，欢迎交流和批评指正： wearyoung@outlook.com