线性模型和多项式模型

线性回归

假设 特征 和 结果 都满足线性。即不大于一次方。这个是针对 收集的数据而言。
收集的数据中，每一个分量，就可以看做一个特征数据。每个特征至少对应一个未知的参数。这样就形成了一个线性模型函数，向量表示形式：

$$h_\theta(x) = \theta^T X$$

对于这个模型的求解，主要就是求解里面中的未知参数。一个线性矩阵方程，利用训练集直接求解。利用训练集，其模型与数据的误差最小的形式，模型与数据差的平方和最小：

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

$$\min_\theta J_\theta$$

这个就是损失函数的来源，利用梯度下降法，本质上是通过偏导数，步长/学习率，更新，收敛通过近似值来当做真实解。

$x_0 = 1$

Hypothesis: $h_\theta(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

Parameters: $\cancel{\theta_0, \theta_1, \ldots, \theta_n}$ θ n+1 - dimensional vector

Cost function:

$$\cancel{J(\theta_0, \theta_1, \ldots, \theta_n)} = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

$J(\theta)$

Gradient descent:

Repeat {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} \cancel{J(\theta_0, \ldots, \theta_n)} \; J(\theta)$$

} (simultaneously update for every $j = 0, \ldots, n$)

其中的Hypothesis是线性模型，x_i表示不同的向量特征，Cost是损失函数

1，梯度下降法（Gradient Descent）

根据平方误差，定义该线性回归模型的损耗函数（Cost Function）为：

$$J(\theta) = J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

，（系数是为了方便求导展示）线性回归的损耗函数的值与回归系数θ的关系是碗状的，只有一个最小点。

多项式回归

假设函数：

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

通过函数转换

其中
$$x_1 = (size)$$
$$x_2 = (size)^2$$

从而得到 多项式回归模型，再次利用梯度下降法进行求解

补充内容：

批量梯度下降法：进一步得到特征的参数θj的迭代式。因为这个迭代式需要把m个样本全部带入计算，所以我们称之为批量梯度下降

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

随机梯度下降法：

批量梯度下降法每次迭代都要用到训练集的所有数据，计算量很大，针对这种不足，引入了随机梯度下降法。随机梯度下降法每次迭代只使用单个样本，迭代公式如下：

$$\theta_j' = \theta_j - \alpha(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$