线性回归(Linear_Regression)

2019年11月1日　　18:30

作者：zx青

线性回归算法

一、回归

回归问题的目标是给定D维输入变量x，并且每一个输入矢量x都有对应的值y，要求对于新来的数据预测它对应的连续的目标值$\hat{y}$。

假定我们现有一大批数据，包含房屋的面积和对应面积的房价信息，如果我们能得到房屋面积与房屋价格间的关系，那么，给定一个房屋时，我们只要知道其面积，就能大致推测出其价格了。通常，这类预测问题可以用回归模型（regression）进行解决，回归模型定义了输入与输出的关系，输入即现有知识，而输出则为预测。

回归分析研究的主要是因变量（目标）和自变量（经验）之间的依存关系。按关系类型，可分为**线性回归分析**和**非线性回归分析**。

二、线性回归

线性回归算法一般用来解决回归问题，结果具有很好的可解释性。

1、简单线性回归

简单线性回归也就是指一元一次回归方程：$y = ax + b$，其中，a是直线的斜率，b是直线的截距。

写成矩阵形式为：$y = X^T a$

其中：$X = \begin{bmatrix} x0 \\ x1 \end{bmatrix}$（令$x0 = 1$），$a = \begin{bmatrix} b \\ a \end{bmatrix}$

从方程可看出，简单线性回归是研究一个因变量与一个自变量间线性关系的方法。

2.多元线性回归

$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$
$\theta = (\theta_0, \theta_1, \theta_2, \ldots, \theta_n)^T$
$x = (x_1, x_2, x_3 + \cdots + x_{n)}$

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_2 \\ \cdots \\ \theta n \end{bmatrix} \quad x = \begin{bmatrix} x_0 \\ x_2 \\ \cdots \\ xn \end{bmatrix} (x_0 = 1)$$

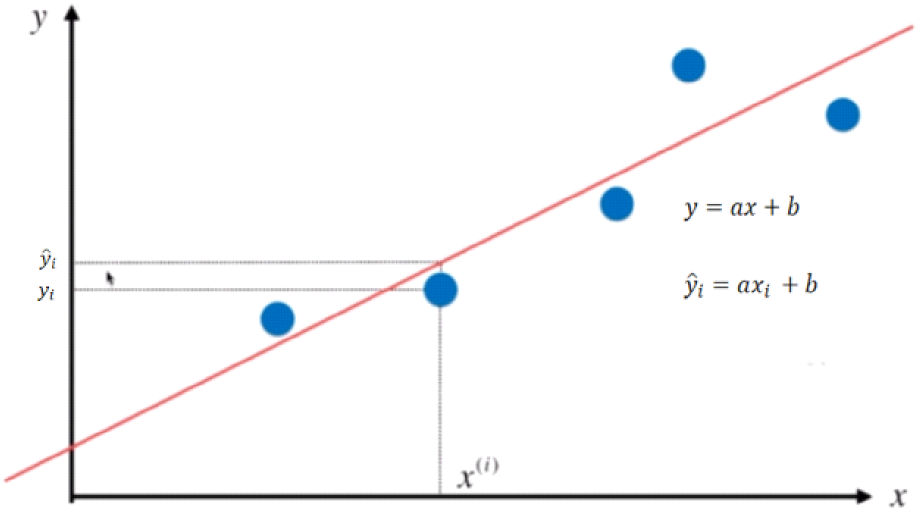
$y = \theta^T x = x\theta$

假如有m个训练样本，则

三、线性回归的损失函数（cost function)

以简单线性回归问题为例

问题：有一堆样本点$(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)$，如果想要探索$x,y$之间的关系，就可以利用一元线性回归找出一条直线，最大程度的拟合样本点之间的关系。



上图中，假设找到了最佳拟合直线方程：$y = ax + b$

则根据方程，每一个样本点x_i的预测值为：$\hat{y}_i = ax_i + b$，真值为 y_i

我们的目标：使y_i和$\hat{y}_i$的差距尽量小 => 使 $\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$尽可能小（这里差值表示为方差而不是绝对值差值主要是为了求导方便）

由$\hat{y}_i = ax_i + b$，目标：找到a和b，使 $\sum_{i=1}^{m}(y_i - ax_i - b)^2$尽可能小（a和b是未知的，x和y是已知的样本点）

该目标是典型的最小二乘法问题（最小化误差的平方）。

令$J(a,b) = \sum_{i=1}^{m}(y_i - ax_i - b)^2$,分别求$\frac{\partial J(a,b)}{\partial b} = 0$，$\frac{\partial J(a,b)}{\partial a}$=0

$$\frac{\partial J(a,b)}{\partial b} = \sum_{i=1}^{m} 2(y_i - ax_i - b)(-1) = 0$$

$$\Rightarrow \sum_{i=1}^{m}(y_i - ax_i - b) = 0$$

$$\Rightarrow \sum_{i=1}^{m}(y_i) - a\sum_{i=1}^{m} x_i - \sum_{i=1}^{m} b = 0$$

$$\Rightarrow \sum_{i=1}^{m}(y_i) - a\sum_{i=1}^{m} x_i - mb = 0$$

$$\Rightarrow mb = \sum_{i=1}^{m}(y_i) - a\sum_{i=1}^{m} x_i \Rightarrow b = \bar{y} - a\bar{x}$$

$$\frac{\partial J(a,b)}{\partial a} = \sum_{i=1}^{m} 2(y_i - ax_i - b)(-x_i) = 0$$

$$\Rightarrow \sum_{i=1}^{m}(y_i - ax_i - b)x_i = 0$$

$$\Rightarrow \sum_{i=1}^{m}(y_i - ax_i - \bar{y} + a\bar{x})x_i = 0$$

$$\Rightarrow \sum_{i=1}^{m}(x_i y_i - ax_i x_i - x_i\bar{y} + ax_i\bar{x}) = 0$$

$$\Rightarrow \sum_{i=1}^{m}(x_i\,y_i - x_i\bar{y} - a{x_i}^2 + ax_i\bar{x}) = 0$$
$$\Rightarrow \sum_{i=1}^{m}(x_i\,y_i - x_i\bar{y}) - \sum_{i=1}^{m}(\,a{x_i}^2 - ax_i\bar{x}) = 0$$
$$\Rightarrow \sum_{i=1}^{m}(x_i\,y_i - x_i\bar{y}) - a\sum_{i=1}^{m}(\,{x_i}^2 - x_i\bar{x}) = 0$$
$$\Rightarrow a = \frac{\sum_{i=1}^{m}(x_i\,y_i - x_i\bar{y})}{\sum_{i=1}^{m}(\,{x_i}^2 - x_i\bar{x})}$$

$$因为\sum_{i=1}^{m}(\,x_i\bar{y}) = \bar{y}\sum_{i=1}^{m}x_i = m\,\bar{y}\bar{x} = \bar{x}\sum_{i=1}^{m}y_i = \sum_{i=1}^{m}\bar{x}y_i \;\;又m\bar{y}\bar{x} = \sum_{i=1}^{m}\bar{x}\bar{y}$$

$$a = \frac{\sum_{i=1}^{m}(x_i\,y_i - x_i\bar{y} - \bar{x}y_i + \bar{x}\bar{y})}{\sum_{i=1}^{m}(\,{x_i}^2 - x_i\bar{x} - x_i\bar{x} + \bar{x}^2)} = \frac{\sum_{i=1}^{m}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{m}(x^i - \bar{x})^2}$$

四、最小化损失函数

1、正规方程

$$\theta = \begin{bmatrix}\theta_0\\ \theta_2\\ \cdots\\ \theta n\end{bmatrix} \quad x = \begin{bmatrix}x_0\\ x_2\\ \cdots\\ xn\end{bmatrix}$$

$$\hat{y}_i = \theta^T x = x\theta$$

目标：使$\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$尽可能小

=> 使 $(y_i - x\theta)^T(y_i - x\theta)$尽可能小

令$E_\theta = (y_i - x\theta)^T(y_i - x\theta)$，对$\theta$求导得：

在求解上述优化问题之前先简单的介绍下矩阵求导法则

$$\begin{aligned}\nabla_x x^T b &= \nabla_x\,[x_1,\quad x_2,\quad \cdots,\quad x_n]\begin{bmatrix}b_1\\ b_2\\ \vdots\\ b_n\end{bmatrix}\\ &= \nabla_{x_i}\sum_{i=1}^{n}x_i b_i\\ &= b\end{aligned}$$

$$\begin{aligned}\nabla_x Ax &= \nabla_x\begin{bmatrix}a_{11} & a_{12} & \cdots & a_{1n}\\ a_{21} & a_{22} & \cdots & a_{2n}\\ \vdots & \vdots & \cdots & \vdots\\ a_{n1} & a_{n2} & \cdots & a_{nn}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ \vdots\\ x_n\end{bmatrix}\\ &= \begin{bmatrix}\nabla_{x_i}\sum_{i=1}^{n}a_{1i}x_i\\ \nabla_{x_i}\sum_{i=1}^{n}a_{2i}x_i\\ \vdots\\ \nabla_{x_i}\sum_{i=1}^{n}a_{ni}x_i\end{bmatrix}\\ &= A^T\end{aligned}$$

$$\frac{\partial E_\theta}{\partial\theta} = 2x^T\left(x\theta - y_i\right),\;令\frac{\partial E_\theta}{\partial\theta} = 0得$$
$$\theta = (x^T x)^{-1}x^T y_i$$

$y_i = (x^T x)^{-1} x^T y_i x$

2、梯度下降算法求解线性回归

$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$
$\hat{y}_i = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \ldots + \theta_i x_i$

五、简单线性回归的python实现
https://github.com/zx-qing/Machine_Learning/blob/master/Linear_Regression/simpleLinerRegression.ipynb
https://github.com/zx-qing/Machine_Learning/blob/master/Linear_Regression/LinearRegression.ipynb

六、总结
预测函数：$\hat{y}_i = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$
cost函数：$\sum_{i=1}^{m} (y_i - \hat{y}_i)^2$
最小化cost函数：梯度下降 正规方程解