线性回归：最小二乘法的几何与概率解释及过拟合与正则化介绍

原创 NK冬至 首席数据科学家 5月13日

收录于话题
#大数据 16 #机器学习 6 #数据科学 16 #数据分析 30 #数据产品 51

> “线性回归是机器学习中的入门模型，也是比较常用的模型之一。”

关于机器学习，之前分享过《机器学习的分类》。今天和大家分享一下线性回归模型的一些内容。有

01
线性回归介绍

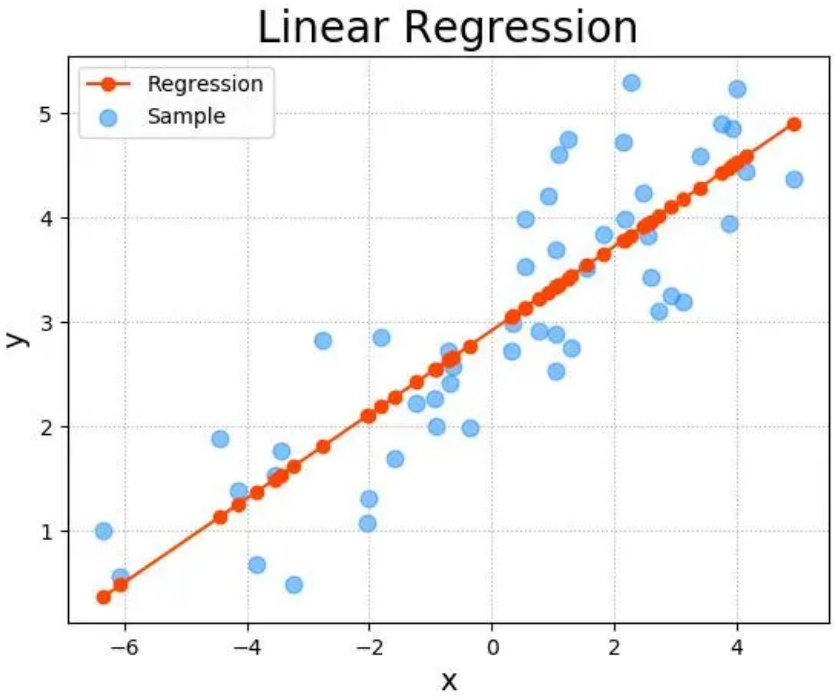
其实线性回归模型，很多朋友应该都在高中阶段接触过。但高中接触的，只是一元线性回归。

所谓一元线性回归，指的是自变量的个数只有1个的情况下（即Y=aX+b），而自变量个数有多个时候，则是多元线性回归。直观的表达式：

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

从应用的角度上讲，明显多元线性回归的实践应用更广泛一些，毕竟一个自变量的情况很少。关于多元分析（即自变量个数是多个），有历史文章可以参考《多元分析概述》。

下面我们看一下回归拟合。由于多元（即多维）的情况下，图形是画不出来的，因此我们这里以二维的图形为例。



图中的横轴即自变量，纵轴即因变量。每个蓝点是样本点，样本点汇合在一起，也就是数据集，我们用下面的集合来表示数据集：

$$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$$

其中，x是p维向量，y是实数。因此，关于自变量、因变量可以记：

$$X = (x_1, x_2, \cdots, x_N)^T, Y = (y_1, y_2, \cdots, y_N)^T$$

红色直线则是拟合后的回归方程，直线上的红点是样本点的映射。

回归的目的，即基于样本点，找到这样的一条直线，使得这条直线能尽可能拟合样本点的趋势。

$$f(w) = w^T x$$

如何找到这条直线呢，也就是求w，使得“尽可能拟合”？其实关于拟合的方式，有很多方法，我们今天讲最常用的最小二乘法，以及在此基础上改良的岭回归。

02

最小二乘法

下面我们首先详细介绍一下最小二乘法的概念以及不同思路的解释。

（1）最小二乘法的定义

我们首先定义一下损失函数（关于损失函数，后面会有一篇专门的介绍，主要的作用是评价模型预测值和真实值之间的差距，也就是模型准确度，并以此进行机器学习）。

$$L(w) = \sum_{i=1}^{N} ||w^T x_i - y_i||_2^2$$

这个比较好理解，其实就是模型的预测值f（w）与真实值y的差的平方和。我们的目标是求参数向量w，使得损失函数L（w）达到最小值，这样就使得模型（也就是回归直线）与真实的样本集拟合最好。用这种方法，就是最小二乘法。

从名称中也可以看出来，就是使得“二乘法”最小。

当然了，损失函数如果是其他形式，就是其他进行线性回归寻找直线的方法了，就不是最小二乘法。

（2）最小二乘的代数解法

代数解法具体就不展开了，简单说一下思路，了解即可。

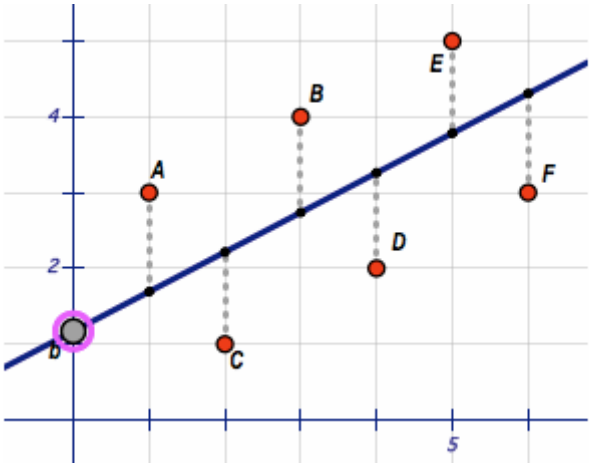
首先，对损失函数经过一些列的变形计算，然后令导数为0，则可以取最值，我们可以得出最终w的取值：

$$\hat{w} = (X^T X)^{-1} X^T Y$$

这就是通过代数方法，求解最小二乘法的矩阵表达式。这个表达式比较重要，相当于最终的结论，下文也会提及。

（3）最小二乘的几何视角

从几何上，最小二乘法该如何表示呢？参考下图：



其实最小二乘就是使得上图中虚线部分的长度求平方后的加和值最小的直线。

近期看了一个视频，讲最小二乘法的另外一种几何表达。令

$$f(w) = X\beta$$

来求β使得Y与X的组合距离最近。由于投影最近，因此有：

$$X^T \cdot (Y - X\beta) = 0 \longrightarrow \beta = (X^T X)^{-1} X^T Y$$

更容易可以得到表达式，和代数的解法是一致的。

（4）最小二乘的概率视角

最后再看一下概率和最小二乘的关系。首先，我们记：

$$y = w^T x + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2)$$

根据统计学的知识，很容易知道：

$$y \sim \mathcal{N}(w^T x, \sigma^2).$$

根据极大似然的参数估计方法（可参考《极大似然估计方法详解》），（省略一万步），最终有：

$$
\begin{aligned}
L(w) = \log p(Y|X,w) &= \log \prod_{i=1}^{N} p(y_i|x_i,w) \\
&= \sum_{i=1}^{N} \log(\frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(y_i - w^T x_i)^2}{2\sigma^2}}) \\
\underset{w}{argmax}\, L(w) &= \underset{w}{argmin} \sum_{i=1}^{N} (y_i - w^T x_i)^2
\end{aligned}
$$

我们发现当噪音服从均值为0的正态分布时，通过极大似然估计得到的回归方程和最小二乘法的表达式是一样的。

03

过拟合及正则化相关

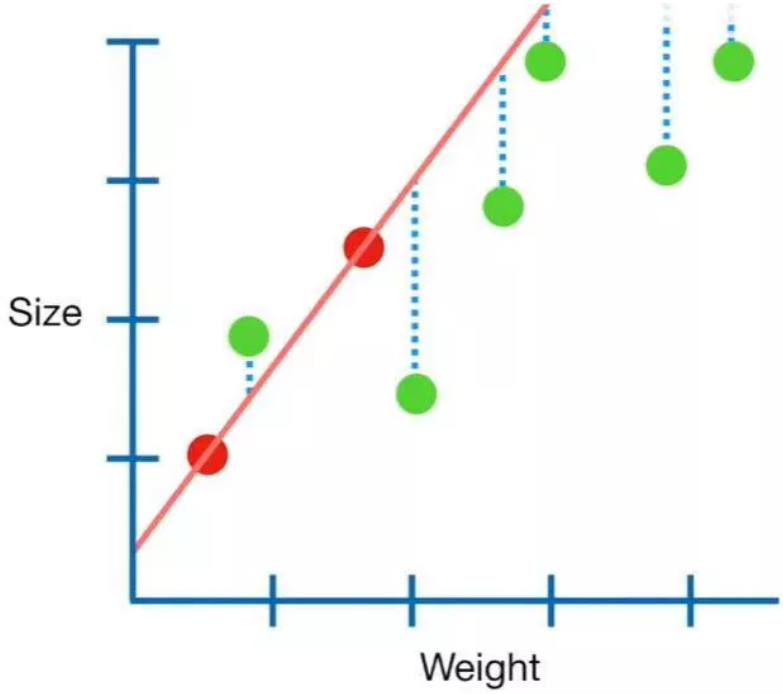
下面我们聊一下关于过拟合以及解决过拟合的方法。

（1）什么是过拟合

上面我们讲的最小二乘法的过程，其实是比较理想化的，即认为样本量远远大于特征维度。在$(X^TX)^{-1}X^TY$表现为(X^TX)是可逆的。

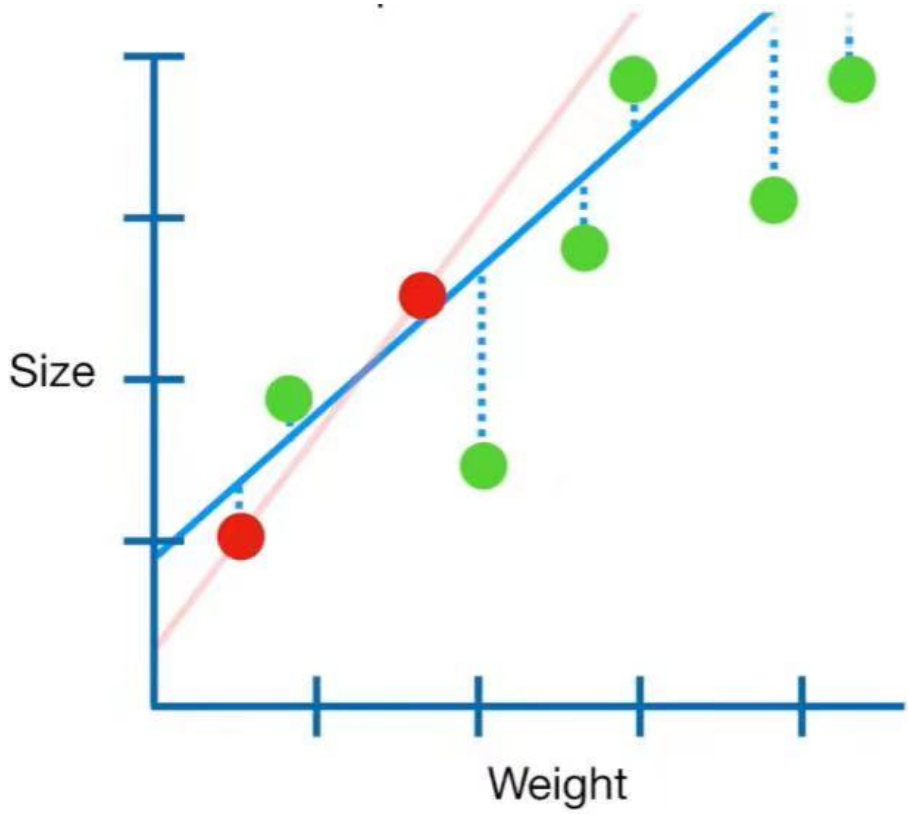
但在实践操作中，经常出现样本容量过少或者特征维度过多的情况。这时(X^TX)其实不可逆。

比如下图：



如果我们只有两个红色的样本点，如果按照最小二乘的方法，那么将得出红色的回归直线，符合使得损失函数最小。但是呢，实际上的样本是上图中的所有点，明显这条回归直线并不是最优的。

极端情况下，只有一个样本，那通过该样本可以有无数条线，都使得损失函数为0。



这就是过拟合。

（2）正则化

如何解决过拟合呢?

其实上文也提到了，过拟合是由于样本容量远小于特征维度导致的。那么相应的，我们可以通过增加样本容量或者减少特征维度（即降维，例如主成分分析）进行解决过拟合。

除此之外，还有一种方法，这就是正则化。

其实正则化就是在原来的损失函数上加上一个正则化项（也可以称为惩罚项）。主要就是参数λ。最终的优化目标由原来的残差平方和最小优化成使得残差平方和+惩罚项最小。

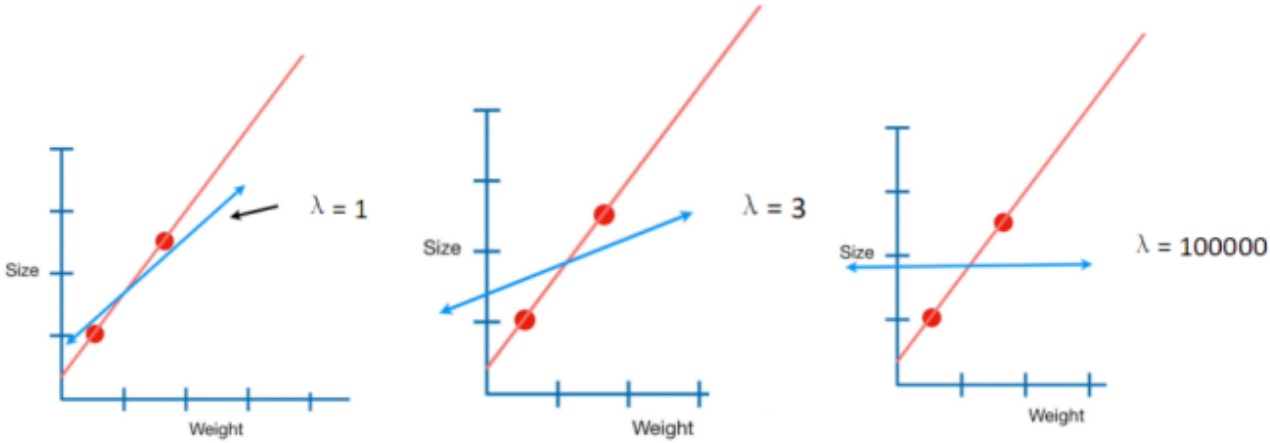
$$L1: \underset{w}{argmin}\, L(w) + \lambda||w||_1, \lambda > 0$$
$$L2: \underset{w}{argmin}\, L(w) + \lambda||w||_2^2, \lambda > 0$$

上面是两种不同的正则化项，主要是范数是1还是2的区别。第一种我们称为L1正则，也叫lasso正则化；第二种我们成为岭回归。

加上正则化项后，原来的最小二乘的矩阵表达式变为（具体计算过程省略）：

$$\hat{w} = (X^T X + \lambda \mathbb{I})^{-1} X^T Y$$

下图是不同λ取值时对回归方程的影响：



关于线性回归相关的内容，今天先介绍这些。很多细节没有过多展开，毕竟内容确实比较多一些。这里就先理了理核心脉络，对该算法有个整体上的认知即可。至于具体的应用，咱们后续结合案例再来分享，谢谢大家！