我对线性回归的理解

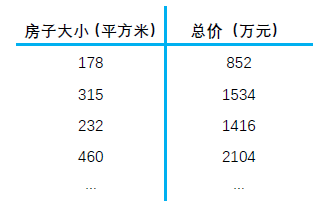
**概述**

本文旨在简明扼要的阐述我对线性回归算法的理解，线性回归是属于监督学习，根据其特征取值可分为一元线性回归和多元线性回归，本文主要以一元线性回归为主。

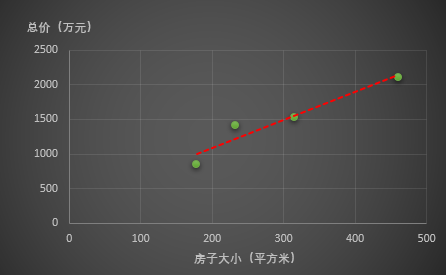
另外，插播一则小广告（本人笔名由以前的**Allen**改为**Anuo.**）谢谢！

**例子**

假设您要出售一套房子，您想预测下到底能买多少钱？您从中介那里得到一些房价信息，我们拿到的数据如下：



在图表上画出的效果如下：



我们的目标就是要找出如上图的红色线条函数，让它尽可能的拟合实际的数据，这样我们就可以预测出房子大小x时可买y刀啦。

上面从中介那里得到的数据就是我们的**训练集**，我们希望通过它来预测，当房子大小为x时能买多少钱y。所以，我们的**假设函数**如下：

根据上面的训练数据，可以推导出:

x=852时，y=178

x=1416时，y=232

以此类推，我们的最终目标就是通过这些训练数据，找到和的取值，来确定我们的线性函数，这个过程就叫**线性拟合**。那么如何进行线性拟合呢？

**模型**

线性回归中我们要解决的是最小化的问题，上面说了，我们的目标是要找到和的取值，来使我们的预测值最接近真实值y，所以需要得到的是的值最小，公式如下：

进一步演变公式：

其中，m为训练集样本数，是我们要尝试尽量减少平均误差，只是为了数学更直白一点。因此对这个求和值的二分之一求最小值，应该得出和的值来。

简单地说，我们要做的就是找到能使训练集中预测值和真实值的差的平方和的1/2m最小的和的值，这就是线性回归的整体目标函数。最终得到我们的**代价函数**如下：

**代价函数**

假设函数：

参数：

代价函数：

目标函数：取得的最小值

假设=0，那么我们的公式就得以简化，如下图所示：

|  |  |  |
| --- | --- | --- |
|  | **公式** | **简化后（=0）** |
| 假设函数 |  |  |
| 参数 | 、 |  |
| 代价函数 |  |  |
| 目标函数 |  |  |

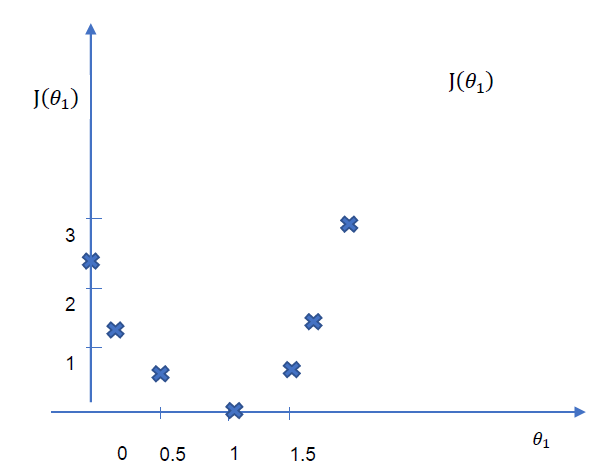
假设我们的训练集为(1,1)，(2,2)，(3,3)…

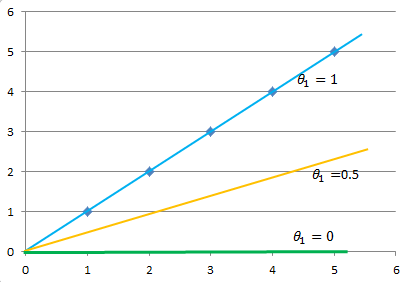
我们假设=1，套用公式:===0

假设=0.5，套用公式:

==

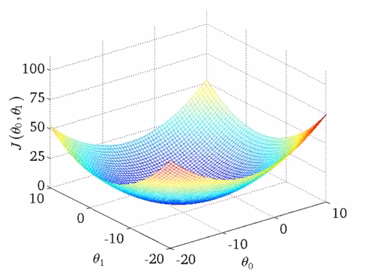
假设=0，继续套用公式，可以得到，以此类推，根据假设值我们可以得到若干个点，(0,2.3)，(0.5,0.58)，(1,0)... 可以画出如下所示的图形：





如上图所示，每一个的取值都对应着一条函数直线，如上图所示，当=1时的值最小，且刚好与我们的训练集重合，完美的拟合了训练数据。

当我们只有一个参数时，我们画出的图形是一个二维的平面碗状图形，当我们有两个参数和时，我们的代价函数的图形就变成如下图形：



这是一个3D曲面图，坐标为和，曲面的高度就是的值，当取不同的和值时，可以得到对应的的值。

如何得到和的值呢？我们可以采用梯度下降法。

**梯度下降**

梯度下降的算法为重复以下算法直到收敛：

(for j=0 and j=1)

其中 为赋值，为**学习率**，求的偏导。

*学习率相当于步进，决定收敛算法的快慢，学习率较小，则收敛速度较慢，学习率较大，则收敛速度较快，但如果学习率过大，则可能导致发散而无法收敛。*

**线性回归与梯度下降**

前面我们已经得到线性回归模型和梯度下降算法，这里将介绍如何结合两者，产生机器学习的第一个算法**线性回归算法。**

根据已得到的模型，进行如下推导：

=

采用多元微分法可得到j=0（）和1（）时的偏导函数如下：

j=0:=

j=1:=

最终，我们得到**线性回归梯度下降算法**如下：

*重复直到收敛* {

} 需要同步更新

后面部分分别是对应和的偏导数

**结束**

好啦，以上是最基本、最简单的线性回归算法，本文注重理论，可能稍显枯燥，后面我会继续分析算法的优化，多元线性回归，并结合实例进行阐述。

我写文章的目的，一是帮助自己加深对机器学习算法的理解；二是很希望能帮助到需要的同学，大家共同进步。

本人知识水平有限，如有错误之处，还请不吝斧正。

Anuo.

成都

Aug 28,2018