**概述**

之前我们讲了最基本、最简单的一元线性回归算法，本章主要讲解多元线性回归,以及梯度下降、特征缩放和正规方程解等方法的应用。

**模型**

正如之前我们讲的那样，简单线性回归在只有一个特征参数时的假设函数如下所示：

那么，如果我们有多个特征时，情况是怎么样的呢？我们还是从卖房子的例子说起，影响我们房子的售价除了大小之外，还有房子的楼层，房间数，房子的年龄，实时上房子的卖价与其大小、楼层、年龄这些都有关系，所以我们得到如下表的房价关系：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **大小（平方）** | **房间数量** | **楼层** | **年龄** | **售价（万元）** |
| 852 | 2 | 1 | 36 | 178 |
| 1534 | 3 | 2 | 30 | 315 |
| 1416 | 3 | 2 | 40 | 232 |
| 2104 | 5 | 1 | 45 | 460 |
| … | … | … | … | … |

根据上面的数据（**训练集**），我们用x\_1表示大小，x\_2表示房间数，x\_3表示楼层，x\_4表示年龄，y表示售价，n表示特征数，那么我们的假设函数表示如下：

我们如何来演变这个公式？假设x\_0=1，那么上面的公式可以写成：

麻烦您再回想一下矩阵的和向量的乘法（如果您不太清楚，可以翻看下我之前写的矩阵和向量的文章），看你能不能找到一些端倪呢？我们可以将参数θ和特征x看做两个向量，如下：

其中x\_0=1，,我们的假设函数是不是可以这样写：

我们来验证一下吧：

是不是就还原回去啦，这就是我们的多元线性回归的**假设函数**。

**梯度下降**

一元线性和多元线性的区别就是特征数从1变成了n，n≥1，也就是说多元包含一元，一元只是特例，从上面得知其假设函数为：

参数为：θ\_0，θ\_1，…，θ\_n，可以看做n+1维向量

那么我们的代价函数为：

梯度下降算法为：

重复直到收敛 {

(for j=0,1,…,n)

}

结合我们在讲一元线性回归的梯度下降，采用多元微分法，我们不难得出：

与一元相同(因为x\_0=1)

与一元相同

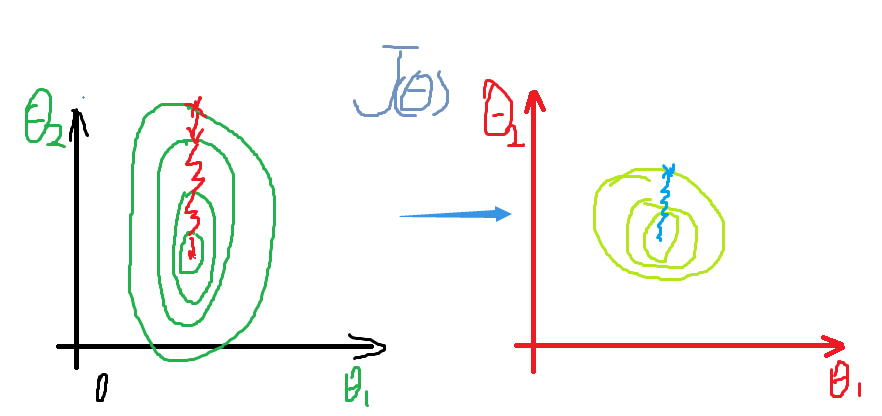
…

这样我们就求出的θ值，从而可以得到我们的算法模型。

**特征缩放**

特征缩放的方法：**确保多个特诊取值在相近的范围内**，这样梯度下降法就能更快地收敛。怎样理解它的意思呢？

假设我们的房子大小x\_1取值（0,2000），卧室数量x\_2取值（1,5），如果其中有的特征值特别大，那么我们画出来的等值线就可能是瘦高的椭圆，如果我们对特征值进行缩放，x\_1 := x\_1/2000，x\_2:=x\_2/5，两个特征值都约束在（0,1）之间，我们画出的代价函数J(θ)等值线是这样的：



从上图可以看出，经过特征缩放之后，我们的梯度下降算法将更快的达到收敛。

除了特征除以最大值来缩放之外，我们还可以采用**均值归一化**的方法来进行缩放，具体如何操作呢？其公式：特征值 = （值 – 平均值）/（最大值-最小值），如房子大小x\_1取值（0,2000），卧室数量x\_2取值（1,5），那么我对x\_1和x\_2重新赋值如下：

这样x\_1的取值范围变成[-0.5,0.5]，x\_2的取值范围变成[-0.25,0.75]，就达到了我们特征缩放的目的，x\_1和x\_2的取值都在（-1,1）区间。（在特征缩放时不需要太过精确，只要能达到我们的目的【**确保多个特诊取值在相近的范围内**】即可）

一般来说，我们进行特征缩放时，通常情况下是将特征的取值约束到-1到+1的范围内，当然也不是严格执行的，只要约束在(-3,+3)的范围还是可以接受的。

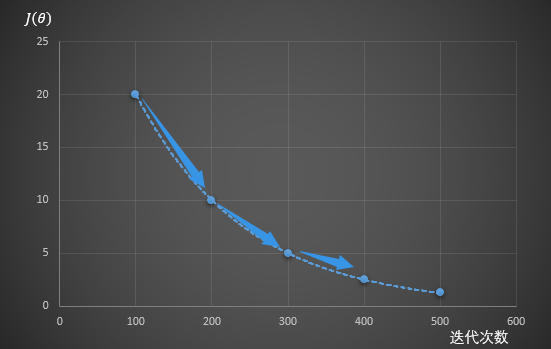
强调一点，特征缩放是针对x\_1,x\_2,…,x\_n，而不是x\_0，因为x\_0是我们为了运算方便人为赋值1的。

**学习率α**

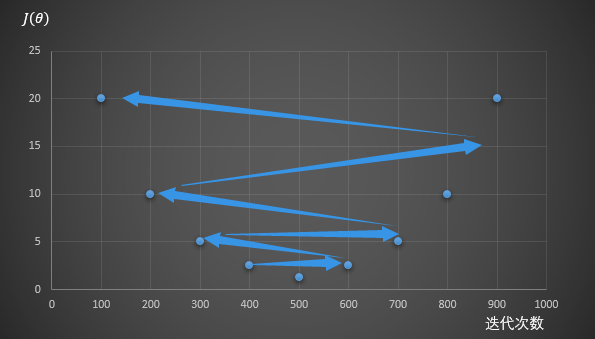
在梯度下降中，我们的迭代公式如下：

上一篇文章中我们提到过学习率，学习率会影响梯度下降算法执行的快慢，学习率太小，则收敛速度慢，过大又可能造成发散而无法收敛，怎样确保我们的梯度下降算法是在正确执行呢？

梯度下降算法的目的是找到一个θ值，能够得到代价函数J(θ)的最小值。如果梯度下降是正确运行的话，画出来的函数图形应该大致如下：



随着迭代次数的增加J(θ)的值越来越小，且到最后变化很小，趋近于0。但是如果我们画出来的图形如下：



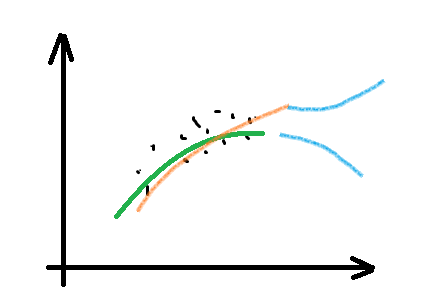
这种情况应该就是学习率过大，导致代价函数的值发散，会造成无法收敛。所以，我们可以通过绘制迭代次数与代价函数的值，可以直观的看出梯度下降是否正确工作。

我们又怎样来选择学习率呢？我们通常会尝试一系列的α值，比如0.001,，0.01，0.1，1，然后取不同的值绘制代价函数J(θ)随迭代次数变化的曲线，然后选择快速下降的一个α值，这个过程一般是经验使然。

**多项式回归**

当我们进行线性回归时，是可以有多种模型来选择的，具体选择哪种模型，需要结合实际情况，如我们拟合房子价格的问题，我们的模型是一个二次函数，正如您所知道的，二次函数在达到一个顶点之后是要下降的，但是实际情况是房价不可能下降，或者比之前的还低，所以我们可以考虑使用多项式模型来进行拟合，比如：

这样的话就更贴合实际情况，绘制出来的图形如下所示：



如上图，绿色线条表示二次函数，淡黄色线条表示多次函数，蓝色表示两者的延长线。（画得很挫，但是只要您明白意思就行了，不要太过纠结。）

假设函数模型并不是固定或者唯一的，比如我们之前的一元线性，因为它不够理想，所以，我们可以选择如下两个模型：

最终选择哪个模型，需要比较看哪个模型对训练集的数据拟合得最好。

**正规方程**

我们除了采用梯度下降来求θ的值之外，还可以采用正规方程法来计算θ的值，它可以直接一次性求解θ的最优值。那么正规方程究竟是如何做的呢？

假设我们的代价函数如下：

其中，θ为实数，这是一个二次函数，如何求解它的最优值呢？在梯度下降中我们得知：

如上面的公式，当J(θ)函数的偏导等于0时，梯度下降的迭代停止，也就是我们需要求解的θ。（这里可能需要一些微积分的知识，不过不必担心，就算没有储备也没关系，我们继续）

以文章开头的训练数据为例：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **x\_0** | **大小（平方）**  **x\_1** | **房间数量**  **x\_2** | **楼层**  **x\_3** | **年龄**  **x\_4** | **售价（万元）**  **y** |
| **1** | 852 | 2 | 1 | 36 | 178 |
| **1** | 1534 | 3 | 2 | 30 | 315 |
| **1** | 1416 | 3 | 2 | 40 | 232 |
| **1** | 2104 | 5 | 1 | 45 | 460 |

数据样本数m=4，我们新加一个x\_0，构建出一个矩阵X和向量y，如下：

X是一个mx(n+1)矩阵，y是一个m维向量，我们可以用如下公式计算θ的值：

解释下这个公式：X转置乘以X的逆乘以X转置乘以y。我们暂时不推导这个公式的由来，后期再研究，现在您先知道是这么回事就行。

这就是正规方程法，之前我们提到了特征缩放，您在采用方程法求解时，是不需要进行特征缩放的。

我们来对比梯度下降和方程法的区别有哪些呢？

|  |  |  |
| --- | --- | --- |
|  | 梯度下降 | 正规方程法 |
| 需要选择学习率 | 是 | 否 |
| 需要迭代多次 | 是 | 否 |
| 使用于超大数据 | 是 | 否(时间复杂度=n\*n\*n) |
| 适用于多种算法（如逻辑回归算法） | 是 | 否 |

当特征数n小于10000时可采用方程法，不然您就需要考虑梯度下降法了。

**结束**

以上就是我对多元回归的一些简单理解和总结，希望可以和大家交流。

Anuo.

成都

Aug 31,2018