链滴

多元线性回归解决机器学习问题的一般方法

作者：vcjmhg

原文链接：https://ld246.com/article/1591100963946

来源网站：链滴

<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/bing/20181207.jpg?imageView2/1/w/960/h/540/interlace/1/q/100">
<h2 id="前言">前言</h2>
线性回归作为机器学习中经典方法之一，以其形式简单、易于建模而被广泛应用。虽然说线性回
形式简单但却蕴含着机器学习中一些非常重要的思想。许多更能更为强大的非线性模型可在线性模型
基础上通过引入层级结构或者高纬映射而得到。因此线性模型在机器学习领域是极为重要的模型之一

而线性回归中多元线性回归较之一元线性回归应用范围更广，应用过程也更加复杂。因此有必要
多元线性回归在机器学习相关领域一些问题的一般过程进行归纳和总结。当然这一系列步骤成立的前
在于：通过一定分析，可以预见该问题可以通过多元线性回归来解决。
<h2 id="多元线性回归解决问题的一般方法">多元线性回归解决问题的一般方法</h2>
<h2 id="问题抽象">问题抽象</h2>
当遇到一个具体问题时，首先要对具体的问题抽象成数学语言，并以恰当的数学符号来表示，从
建立起数学模型，使得问题更加直观便于分析。 而一般来说线性回归遇到的问题一般是这样的：给定
dd个属性描述的示例\
oldsymbol{x}=\left(x_{1} ; x_{2} ; \ldots ; x_{d}\right)\boldsymbol{x}=\left(x_{1} ; x_{2} ; \ldots ; x
{d}\right)，其中 x_{i} x_{i}是<span class="lan
uage-math">xx在第ii个属性上的取值，线性
型（linear model）试图通过属性的线性组合来进行预测的函数,即
f(\boldsymbol{x})=w_{1} x_{1}+w_{2} x_{2}+\ldots+w_{d} x_{d}+

一般向量形式写成
f(x)=w^{T} x+b
其中w=(w_1;w_2;w_3...;w_d)w=(w_1;w_2;w_3...;w_d)</spa
>。ww和bb</span
学得之后，模型就得以确定。因此通过该步的数学抽象就可以明确，整个线性回归的整个过程主要目
就是确定合适的ww 和
b使得训练的模型尽可能地与实际情况相吻合。
<h2 id="数据预处理">数据预处理</h2>
其实这一步对所有算法都是适用的。毕竟机器学习算法都是建立在数据的基础之上的，因此对于
据的预处理对所有算法来说都是极其重要的。该过程大体上分为以下几步：

数据收集：简单来说就是通过各种手段收集自己所需要的数据，例如爬虫、开源数据集等
数据清洗：包括数据格式的转化，数据的清洗（处理噪声数据缺失值数据），以及数据的采样</l
>
数据等价转换：包括统一数据的度量（这在距离计算时十分重要）、零均值化、属性分解以及合

属性的分解是，一个属性能够分解成多个属性，只有某些子属性对于输出有显著的影响。那我们
可以只存储这些子属性对于输出具有显著影响，那我们就可以只存储这些子属性，而不用去存储原来
属性。其中数据的合并和属性的分解是对立的。将一些子属性合并成一个新属性后，这个属性对于输
的影响会更加显著。那就将这个属性进行合并。
通过数据预处理阶段后，数据会变得更加规整，便于训练模型时直接使用。
<h2 id="确定假设函数">确定假设函数</h2>
经过数据预处理之后，一般来说就可以确定出一些与输出相关性强的属性，因而可以开始尝试构
出解决问题的假设模型。比如说如果确定出与输出相关的属性是 n 维的，那么就可以建立起如下的假
函数：
f(\boldsymbol{x})=w_{1} x_{1}+w_{2} x_{2}+\ldots+w_{n} x_{n}+

其中：

x=\left(x_{0}, x_{1}, x_{2}, \ldots, x_{n}\right)^{T}x=\left(x_{0
, x_{1}, x_{2}, \ldots, x_{n}\right)^{T}为输入，n+1n+1<
span>维的列向量，其中x_0=1x_0=1

w=\left(b, w_{1}, w_{2}, \ldots, w_{n}\right)^{T}w=\left(b, _{1}, w_{2}, \ldots, w_{n}\right)^{T}为参数，n+1n+1</pan>维的列向量，bb是<span class="language-math>biasbias

<h2 id="构建代价函数LMS">构建代价函数 LMS</h2>
在确定假设函数之后，需要确定模型中的参数，具体如何确定那？显然关键在于衡量f(x)f(x)与yy之间的差别因此需要建立起一个代价函数来进行衡量。均方误差是回归任务重最常见的性能度量，因此我们以均误差为例，建立起如下的代价函数：
E(w,b) =\underset{(w, b)}{\arg \min } \sum_{i=1}^{n}\left(f\left(x{i}\right)-y_{i}\right)^{2}
<h2 id="模型训练">模型训练</h2>
对于不同的参数，成本函数有不同的值，而我们所需要的是让代价函数最小化即：
\begin{aligned}
\left(w^{*}, b^{*}\right) &=\underset{(w, b)}{\arg \min } \sum_{i=1}^{n}\left(f\left(x_{i}\right)-y_{i}\right)^{2} \\
&=\underset{(w, b)}{\arg \min } \sum_{i=1}^{n}\left(y_{i}-w x_{i}-b\right)^{2}
\end{aligned}
求解ww和bb使得E_(w,b)E_(w,b)最小化的过程，称之为线性回归型的最小二乘参数估计（parameter estimation）。通过最小二乘法我们可以推导出通用的线性回归型如下：
f\left(\hat{\boldsymbol{x}}_{i}\right)=\hat{\boldsymbol{x}}_{i}^{mathrm{T}}\left(\mathbf{X}^{\mathrm{T}} \mathbf{X}\right)^{-1} \mathbf{X}^{\mathrm{T}} \bodsymbol{y}
其中：

XX:数据集DD</san>所表示的
m \times(d+1) m \times(d+1) 大小的矩阵
\left(\mathbf{X}^{\mathrm{T}} \mathbf{X}\right)^{-1}\left(mathbf{X}^{\mathrm{T}} \mathbf{X}\right)^{-1}为\left\mathbf{X}^{\mathrm{T}} \mathbf{X}\right)\left(\mathbf{X}^{\mathrm{T}} \mathbf{X}\right)</pan>的逆矩阵

如果训练集的矩阵维数不是太大，并且\mathbf{X}^{\mathrmT}} \mathbf{X}\mathbf{X}^{\mathrm{T}} \mathbf{X}刚好为满秩矩阵。我们可以直接套上边公式得出结果。
当然现实任务中\mathbf{X}^{\mathrm{T}} \mathbf{X}\matbf{X}^{\mathrm{T}} \mathbf{X}往往不是满秩矩阵。例如在许多任务中我们会遇到大量的量，其数目甚至超过样例数导致XX的列数多于行数，此\mathbf{X}^{\mathrm{T}} \mathbf{X}\mathbf{X}^{\mathrm{T}} \mathbf{X}显然不是满秩的。此时需要解出多个\hat{w}\at{w}，它们都能使均方差最小化，具体选择哪一种需要通过学习算法的归纳偏好来决定，见的做法是引入正则化项。当然如果如果训练集的矩阵维数很大这种方法，显然不太合适，此时我们以考虑通过梯度下降算法来求出参数。关于梯度下降算法的具体使用，可以参考周志华老师的《机器习》，此处不再详述。
<h2 id="预测与结果优化">预测与结果优化</h2>
在参数求出来之后，我们便建立起了一个针对该问题的线性回归模型，然后便可以使用该模型进预测。当然可能此时得出的模型与现实问题特别吻合，可能造成输入结果与实际结果有出入。此时，要对结果进行优化。
<h2 id="总结">总结</h2>
本文尝试从实际问题入手，归纳出了多元线性回归解决实际问题的一般方法的步骤。一般来说，

元线性回归在解决实际问题时，需要经过问题抽象、数据预处理、假设函数的确定、构建代价函数、型训练和预测与结果优化六个部分。一个实际问题经过这几个步骤之后，一般来说便可以建立起一个较合适的多元线性模型。</p>
<h2 id="参考文献">参考文献</h2>

https://www.cnblogs.com/ordil/p/10262569.html
https://zhuanlan.zhihu.com/p/49222906
https://blog.csdn.net/tqinag/article/details/54730360
https://baie.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%A2%84%E5%A4%84%E7%90%86/27112883](https://baike.baidu.com/item/数据预处理/2711288#3)
周志华 《机器学习》[M].北京：清华大学出版社,2016,74-8
