**对于线性回归的一些心得**

**一元线性回归**

一元线性回归就是寻找某一个自变量与某一因变量的线性关系，其方程式通常为Y = wX + b 的 类型

而求出这一个方程的方法叫做最小二乘法，即寻找一条直线，并使这条直线到所有数据点在y方向上的和最小

可视化如下（beta1即是w）：

Diagram

Description automatically generated with medium confidence

为了求出这个方程，我们需要最小化（y的预测值与y的实际值的差），根据南瓜书的求解过程，可求出w

Text, letter

Description automatically generated

求出b之后，可根据b的偏导方程，令该方程等于0，从而求出b

Text

Description automatically generated with medium confidence

打开spyder，先读取数据：

import pandas as pd

#读取数据 data = pd.read\_excel(r'C:\Users\14272\Desktop\test\sklearn\线性回归\sklearn实现一元线性回归\data1.xlsx')

分离自变量与因变量

x = data[['广告投入']] y = data[['销售额']]

用statsmodel库调出线性回归

import statsmodels.api as sm

#添加常数项 X = sm.add\_constant(x)

#最小二乘法 model = sm.OLS(y,X) result = model.fit()

#系数 result.params

#汇总结果 result.summary()

最后得如下结果：

A screenshot of a computer

Description automatically generated with medium confidence

**多元回归方程**

多元回归的核心思想也是最小二乘法，希望能用n-1个维度的方程来解释n维度的数据

可视化如下：

Chart, funnel chart

Description automatically generated

此时，我们规定w^ 是beta0, beta1,.....组成的一个向量，根据南瓜书的推导过程，

该向量可以被如下方程求出

Schematic

Description automatically generated with low confidence

由于 样本是多变的，因此无法保证 XTX是 满秩矩阵，当 XTX 非正定矩阵时，则不可逆，如果不可逆，则该方程求不出来。除了引入正则化外，也可用 XTX 的伪逆矩阵

Text

Description automatically generated

求解出 ˆw∗，只是此时并不保证求解得到的 ˆw∗ 一定是全局最优解。除此之外，也可用梯度下降法，同样也不保证求得全局最优解。

打开spyder，先读取数据：

import pandas as pd

#读取数据 data = pd.read\_csv(r'C:\Users\14272\Desktop\test\sklearn\线性回归\statsmodels实现多元线性回归\多元线性回归.csv',encoding='gbk',engine='python')

分离自变量与因变量

x = data[['体重','年龄']] y = data[['血压收缩']]

用statsmodel库调出线性回归

import statsmodels.api as sm

#添加常数项 X = sm.add\_constant(x)

#最小二乘法 model = sm.OLS(y,X) result = model.fit()

#系数 result.params

#汇总结果 result.summary()

最后得如下结果：

Text

Description automatically generated

statsmodel的一些api参数：

class statsmodel.OSL(fit\_intercept=True, normalize=False, copy\_X=True, n\_jobs=None)

1.参数： fit\_intercept: 是否有截据，如果没有则直线过原点。 normalize:是否将数据归一化。 copy\_X ：是否对X复制，如果选择false，则直接对原数据进行覆盖。（即经过中心化，标准化后，是否把新数据覆盖到原数据上） n\_jobs：计算时设置的任务个数(number of jobs)。如果选择-1则代表使用所有的CPU。这一参数的对于目标个数>1（n\_targets>1）且足够大规模的问题有加速作用。 2.Attributes: 【返回值】 Coef*:对于线性回归问题计算得到的feature的系数。如果输入的是多目标问题，则返回一个二维数组(n\_targets, n\_features)；如果是单目标问题，返回一个一维数组 (n\_features,)。 intercept* ： 线性模型中的独立项。

3.方法： fit(X, y[, n\_jobs]) ： 对训练集X, y进行训练。 predict(X)： 使用训练得到的估计器对输入为X的集合进行预测（X可以是测试集，也可以是需要预测的数据）。 score(X, y[,]sample\_weight)： 预测效果评分。