《系统工程导论》黑箱建模作业 2

多元线性回归（F 检验）

【1】试说明：病态线性回归问题中，显著性检验是否需要？如果需要，是在自变量降维去线性之前，还是之后，还是前后都检验？给出理由证明你的结论。

答：病态回归也需要显著性检验。检验在去线性化之前。

首先，显著性检验用于确定变量之间的线性相关程度，因此，即使是病态问题，在进行线性回归时，也需要进行显著性检验。

其次，显著性检验是通过最终得到的拟合预测值和原始数据相互比较进行检验的，因此，需要将降维之后的数据映射到原始数据（即为不降维数据），再进行检验。

【2】编程实现多元线性回归，自适应多元、病态。

【算法原理】

在最小二乘意义下，求线性多元线性拟合。

【算法步骤】

【1】数据规范化处理：将 Y、X 进行规范化处理，化为均值为 0，方差为 1 的数据。

目的：消除数据单位对拟合带来的影响，使得回归参数更接近临界值

【2】求矩阵 XX^T 的特征值和特征向量。

【3】将特征值从大到小进行排序，对应的特征向量调整位置。

【4】根据病态问题的阈值要求，从小到大删除特征值，选取最小的 m，使得前 m 个特征值满足阈值条件。

【5】判断：若 m 等于所有特征值的个数，则该问题不是病态问题；否则，该问题为病态问题。

【6】选取前 m 个特征向量，进行下面的计算：

$$
\begin{aligned}
Z &= Q_N X \\
\dot{d} &= \left(ZZ^T\right)^{-1} ZY^T \\
\hat{c} &= Q\hat{d}
\end{aligned}
$$

既得规范化数据的线性回归结果。

【7】对数据进行去规范化，得到最终的回归直线方程，打印输出回归方程。

【8】进行显著性检验（此处选取常用的 F 检验）：

$$
F = \frac{\frac{ESS}{\sigma_\mu^2}/k}{\frac{RSS}{\sigma_\mu^2}/(n-k-1)} = \frac{ESS/k}{RSS/(n-k-1)} = \frac{\sum\left(\hat{Y}_i - \overline{Y}\right)^2/k}{\sum\left(Y_i - \hat{Y}_i\right)^2/(n-k-1)}
$$

通过查表得到临界值，若 F>临界值，线性相关；否则，线性无关。

【9】求得置信区间:

$$(-Z_{\alpha/2}S_a, +Z_{\alpha/2}S_a)$$

其中:

$$s_a = \sqrt{\frac{\sum_{i=1}^{n}(y_i-\hat{y})^2}{N-2}} = \sqrt{\frac{(1-r^2)L_m}{N-2}}$$

$Z_{\alpha/2}$ 为标准正态分布上α/2 分为点。

【算法测试 】

【1】病态问题:

题目做给数据即使一个病态问题:

观测号	x1	x2	x3	x4	y
1	149.3	4.2	80.3	108.1	15.9
2	161.2	4.1	72.9	114.8	16.4
3	171.5	3.1	45.6	123.2	19.0
4	175.5	3.1	50.2	126.9	19.1
5	180.8	1.1	68.8	132.0	18.88
6	190.7	2.2	88.5	137.7	20.4
7	202.1	2.1	87.0	146.0	22.7
8	212.4	5.6	96.9	154.1	26.5
9	226.1	5.0	84.9	162.3	28.1
10	231.9	5.1	60.7	164.3	27.6
11	239.0	0.7	70.4	167.6	26.3

回归输出：

```
[1] 经检验，该线性回归问题为病态线性回归问题
[2] 线性回归方程为： y = 0.073*X1 + 0.599*X2 + 0.002*X3 + 0.105*X4 + (-9.151)
[3] F检验的值： 125.43203179472978　　F检验的临界值： 4.533676950275243
[4] 在显著水平取： 0.05 时，经检验，线性相关
[5] 置信区间（误差范围） ：[ -1.2483153313864315 , 1.2483153313864315 ]
```

【2】正常问题：

数据：

```
X1 = [[100,4],
      [50,3],
      [100,4],
      [100,2],
      [50,2],
      [80,2],
      [75,3],
      [65,4],
      [90,3],
      [90,2]]
Y1 =[9.3, 4.8, 8.9, 6.5, 4.2, 6.2, 7.4, 6, 7.6, 6.1]
```

```
[1] 经检验，该线性回归问题没有病态
[2] 线性回归方程为： y = 0.061*X1 + 0.923*X2 + (-0.869)
[3] F检验的值： 32.87836742581299　　F检验的临界值： 4.73741412777588
[4] 在显著水平取： 0.05 时，经检验，线性相关
[5] 置信区间（误差范围） ：[ -1.1233379761785336 , 1.1233379761785336 ]
```

【代码】

```
# -*- coding: utf-8 -*-
# 张嘉玮
# 20190409
import numpy as np
from scipy.stats import f
from scipy.stats import norm


def linear_regression(Y,X,alpha):
    """
    可自适应病态问题的多元线性回归问题
    :param Y: 1xN
    :param X: NxX 的维度
    :param alpha: 显著性检验参数
    :return: 无
    """
    x_dim = len(X[0])
    N = len(Y)

    Y = np.array(Y)
    X = np.array(X)

    # X 归一化
    x_means = np.array([np.mean(X[:,i]) for i in range(x_dim)])
    x_vars = np.array([np.var(X[:,i]) for i in range(x_dim)])
    X_new = np.zeros((N,x_dim))
    for i in range(N):
        for j in range(x_dim):
            X_new[i][j] = (X[i][j]-x_means[j])/np.sqrt(x_vars[j])

    # Y 归一化
    y_mean = np.mean(Y)
    y_var = np.var(Y)
    Y_new = np.array([[(i-y_mean)/np.sqrt(y_var) for i in Y]])

    Xt = np.transpose(X_new)
    XXt = np.dot(Xt,X_new)

    # 求得特征值和特征向量
    eigenvalue, featurevector = np.linalg.eig(XXt)
    # 排序
    eigenvalue_index_sort = np.argsort(eigenvalue)
```

```
    m = len(eigenvalue)
    for i in range(len(eigenvalue)):
        idx = eigenvalue_index_sort[0:i+1]
        if np.sum([eigenvalue[j] for j in idx])/np.sum(eigenvalue)>0.1:
            m = m-i
            break

    # 病态与否判断
    if m!=len(eigenvalue):
        print("[1] 经检验，该线性回归问题为病态线性回归问题")
    else:
        print("[1] 经检验，该线性回归问题没有病态")

    Q_m = []
    featurevector.tolist()
    for i in range(m):
        a = featurevector[:,eigenvalue_index_sort[len(eigenvalue)-1-i]]
        Q_m.append(a.tolist())
    Q_m = np.array(Q_m)
    Q_m = np.transpose(Q_m)
    ZZ_T_inverse = np.zeros((m,m))

    for i in range(m):
        ZZ_T_inverse[i][i] =
1/eigenvalue[eigenvalue_index_sort[len(eigenvalue)-1-i]]
    d =
np.dot(np.dot(ZZ_T_inverse,np.transpose(Q_m)),np.dot(np.transpose(X_new),np.
transpose(Y_new)))
    c_0 = np.dot(Q_m,d)

    # 去规范化:
    c = [c_0[i]*np.sqrt(y_var)/np.sqrt(x_vars[i]) for i in range(x_dim)]
    bias = y_mean-np.sum([x_means[i]*c[i] for i in range(x_dim)])

    # 打印输出回归结果
    S = 'y = '
    for i in range(x_dim):
        if c[i][0]>0:
            S = S+str(round(c[i][0],3))+'*X'+str(i+1) + ' + '
        elif c[i][0]<0:
            S = S+' ('+str(round(c[i][0],3))+')*X'+str(i+1) + ' + '
    if bias>0:
        S = S+str(round(bias,3))
    elif bias<0:
```

```
        S = S+'('+str(round(bias,3))+')'
    print("[2] 线性回归方程为：",S)

    # 显著性检验:
    Y_pre = []
    for i in range(N):
        Y_pre.append(bias+np.sum([c[j][0]*X[i][j] for j in range(x_dim)] ))
    ESS = np.sum([np.square(Y_pre[i]-y_mean) for i in range(N)])
    RSS = np.sum([np.square(Y_pre[i]-Y[i]) for i in range(N)])
    F = (ESS/x_dim)/(RSS/(N-x_dim-1))
    F_0 = f.ppf(1-alpha,x_dim,N-x_dim-1)
    print("[3] F 检验的值：",F,"     F 检验的临界值：",F_0)
    if F>F_0:
        print("[4] 在显著水平取：",alpha,"时，经检验，线性相关")
    else:
        print("[4] 在显著水平取：",alpha,"时，经检验，线性无关")
    sig = norm.ppf(1 - alpha / 2)
    S_a = np.sqrt(np.sum([i * i for i in Y - Y_pre]) / (N - x_dim - 1))
    print('[5] 置信区间（误差范围） ：[', -sig * S_a, ',', sig * S_a, ']')

# 病态问题
Y = [15.9,16.4,19.0,19.1,18.88,20.4,22.7,26.5,28.1,27.6,26.3]
X = [[149.3,4.2 ,80.3 ,108.1],
     [161.2,4.1 ,72.9 ,114.8],
     [171.5,3.1 ,45.6 ,123.2],
     [175.5,3.1 ,50.2 ,126.9],
     [180.8,1.1 ,68.8 ,132.0],
     [190.7,2.2 ,88.5 ,137.7],
     [202.1,2.1 ,87.0 ,146.0],
     [212.4,5.6 ,96.9 ,154.1],
     [226.1,5.0 ,84.9 ,162.3],
     [231.9,5.1 ,60.7 ,164.3],
     [239.0,0.7 ,70.4 ,167.6]]
linear_regression(Y,X,0.05)

# 正常问题
X1 = [[100,4],
      [50,3],
      [100,4],
      [100,2],
      [50,2],
      [80,2],
      [75,3],
      [65,4],
```

```
      [90,3],
      [90,2]]
Y1 =[9.3, 4.8, 8.9, 6.5, 4.2, 6.2, 7.4, 6, 7.6, 6.1]
linear_regression(Y1,X1,0.05)
```