

## 偏最小二乘法回归模型

### 1、偏最小二乘回归模型原理

偏最小二乘回归法是一种能进行多元统计分析的方法，它具有主成分分析法、典型相关分析法、线性回归分析的特点，能有效地找出变量间的统计关系<sup>[1]</sup>。其原理为：

考虑存在  $q$  个因变量  $\{y_1, y_2, \dots, y_q\}$  以及  $p$  个自变量  $\{x_1, x_2, \dots, x_p\}$  时的偏最小二乘回归问题。在  $n$  个样本点中，构成两个  $n$  次观测矩阵  $X = (x_1, x_2, \dots, x_p)_{n \times p}$  和  $Y = (y_1, y_2, \dots, y_q)_{n \times q}$ ，用以探究因变量与自变量之间的统计关系。接下来，分别在观测矩阵  $X$  和  $Y$  中，提取出成分  $t_1$  以及  $u_1$ ，其中， $t_1$  是  $x_1, x_2, \dots, x_p$  的线性组合， $u_1$  是  $y_1, y_2, \dots, y_q$  的线性组合。进行回归分析，需要达到下述两个要求：

- (1)  $t_1, u_1$  中应尽量多地提取原自变量集中的变异信息，使所提取的成分方差最大；
- (2)  $t_1, u_1$  的相关程度最大。

达到这两个要求时，不仅能使  $t_1, u_1$  最大程度地携带  $X, Y$  构成的观测矩阵的信息，并且能够保证  $u_1$  对  $t_1$  的解释能力最强。

当第一个成分被提取后，实施  $X$  对  $t_1$  的回归及  $Y$  对  $u_1$  的回归。若回归方程达到所需要的满意精度，则终止算法；否则，提取第2对成分，用  $X$  被  $t_1$  解释后的剩余信息代替  $X$ ，及  $Y$  被  $u_1$  解释后的剩余信息代替  $Y$ ，重复进行回归的步骤，直到达到所需要的满意精度为止。若最终对  $X$  提取的成分数为  $m$  个，即提取的所有成分为  $t_1, t_2, \dots, t_m$ ，则偏最小二乘回归通过实行  $y_r (r=1, 2, \dots, q)$  对  $t_1, t_2, \dots, t_m$  的回归，表达为  $y_r$  关于自变量  $x_1, x_2, \dots, x_p$  的回归方程式，这些方程式就是最小二乘回归方程式。

### 2、偏最小二乘回归操作步骤

本题应做的是多因变量偏最小二乘法回归分析。

将某两个指标中的六个理化因子分别作为自变量集  $X = \{x_1, x_2, \dots, x_p\}^T$ （在这里  $p=6$ ）和因变量集  $Y = \{y_1, y_2, \dots, y_q\}^T$ （同取  $q=6$ ）， $t_1, u_1$  分别是自变量集  $X$ 、因变量集  $Y$  的线性组合：

$$t_1 = w_{11}x_1 + w_{12}x_2 + \dots + w_{1p}x_p = w_1^T X$$

$$u_1 = v_{11}y_1 + v_{12}y_2 + \dots + v_{1q}y_q = v_1^T Y$$

- (1) 第1个成分  $t_1$  的提取

由这两组变量集  $X, Y$  分别构成观测矩阵  $X_\alpha, Y_\alpha$ ，可以分别计算出第1对成分的得分向量  $\hat{t}_1, \hat{u}_1$ ：

$$\hat{t}_1 = X_\alpha w_1^T = \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{bmatrix} \begin{bmatrix} w_{11} \\ \vdots \\ w_{1p} \end{bmatrix} = \begin{bmatrix} t_{11} \\ \vdots \\ t_{n1} \end{bmatrix}$$

$$\hat{u}_1 = Y_\alpha v_1^T = \begin{bmatrix} y_{11} & \cdots & y_{1p} \\ \vdots & \ddots & \vdots \\ y_{n1} & \cdots & y_{np} \end{bmatrix} \begin{bmatrix} v_{11} \\ \vdots \\ v_{1p} \end{bmatrix} = \begin{bmatrix} u_{11} \\ \vdots \\ u_{n1} \end{bmatrix}$$

对于  $\hat{t}_1$  的计算公式，组合系数  $w_1^T$  是  $X_\alpha$  的第1个轴，且  $\|w_1^T\|=1$ ；对于  $\hat{u}_1$  的计算公式，组合系数  $v_1^T$  是  $Y_\alpha$  的第1个轴，且  $\|v_1^T\|=1$ 。

根据对主成分分析和典型相关分析的思路，取得：

$$w_1 = \frac{X_\alpha^T Y_\alpha}{\|X_\alpha^T Y_\alpha\|}$$

即可满足进行回归分析的两个要求。

得到  $w_1$ ，也就可以得到  $t_1$ ，于是就能分别得到  $X_\alpha$ 、 $Y_\alpha$  对  $t_1$  的回归方程：

$$\begin{aligned} X_\alpha &= t_1 p_1^T + X_\beta \\ Y_\alpha &= t_1 s_1 + Y_\beta \end{aligned}$$

在上式中， $p_1 = \frac{X_\alpha^T}{\|t_1\|^2}$ ， $s_1 = \frac{Y_\alpha^T}{\|t_1\|^2}$ ； $X_\beta$ 、 $Y_\beta$  为回归方程的残差矩阵。

(2) 第2个成分  $t_2$  的提取

以  $X_\beta$  取代  $X_\alpha$ ，以  $Y_\beta$  取代  $Y_\alpha$ ，用求  $t_1$  的方法，求到第2个轴  $w_2$  以及第2个成分  $t_2$ 。

$$w_2 = \frac{X_\beta^T Y_\beta}{\|X_\beta^T Y_\beta\|}$$

同样， $X_\beta$ 、 $Y_\beta$  分别对  $t_2$  进行回归，得到  $X_\beta$ 、 $Y_\beta$  对  $t_2$  的回归方程：

$$\begin{aligned} X_\beta &= t_2 p_2^T + X_\delta \\ Y_\beta &= t_2 s_2 + Y_\delta \end{aligned}$$

(3) 第  $h$  个成分的提取

第  $h$  个成分的提取与第1、第2个成分的提取原理相同。并且在提取的每一个成分的过程中，都需要保证每两个变量组之间的相关性达到最大。

$h$  的个数需要小于观测矩阵的秩，并且  $h$  的个数可以用交叉有效性原则进行识别。

### 3、交叉有效性检验

简单来说，交叉有效性的操作过程，是在模型可用的情况下，确定了成分个数为  $h$  时，再增添一个成分，使总成分个数变为  $h+1$  个。当成分个数变为  $h+1$  个时，如果导致模型不可用或结果没有明显的改进，就可以证明成分个数应为  $h$ 。

接下来，本文将会叙述交叉有效性的详细进行过程<sup>[2]</sup>：

记原始数据为  $y_i$ ，提取的主成分有  $m$  个 ( $t_1$ 、 $t_2$ 、 $\dots$ 、 $t_m$ )， $\hat{y}_{gi}$  是利用所有的样本点以及  $m$  个成分进行回归后，得到的第  $i$  个样本点的拟合值，将其记为：

$$S_h = \sum_{i=1}^n (y_i - \hat{y}_{gi})^2$$

$\hat{y}_{g(i-1)}$  是删去了样本点  $i$ ，但保留  $m$  个成分进行回归后，得到的第  $i-1$  个样本点的拟合值，记为：

$$P_h = \sum_{i=1}^n (y_i - \hat{y}_{h(i-1)})^2$$

基于上述理论，定义  $Q_h^2$  用以得到成分个数  $h$  是否成立。 $Q_h^2$  的计算方式为：

$$Q_h^2 = 1 - \frac{S_h}{P_h}$$

根据  $Q_h^2$  的计算方式，容易看出，每得到一个成分，均可以利用交叉性检验，判断出成分个

数，避免了多进行无意义的操作的情况发生。

一般认为：在第 $t$ 个成分出有 $Q_h^2 < 0.0975$ 时，则模型精度要求较高，可停止提取成分； $Q_h^2 \geq 0.0975$ 时，增加的成分 $t_i$ 的贡献效果显著，应继续提取成分。

4、求回归系数（程序中有详细说明）

略。。。

案例

自变量			因变量
X1	X2	X3	
1. 7547	1. 2575	1. 5678	1. 7482
2. 276	2. 8407	2. 0759	2. 4505
3. 6797	3. 2543	3. 054	3. 0838
4. 6551	4. 8143	4. 5308	4. 229
5. 1626	5. 2435	5. 7792	5. 9133
6. 119	6. 9293	6. 934	6. 1524
7. 4984	7. 35	7. 1299	7. 8258
8. 9597	8. 1966	8. 5688	8. 5383
9. 3404	9. 2511	9. 4694	9. 9961
10. 5853	10. 616	10. 0119	10. 0782
11. 2238	11. 4733	11. 3371	11. 4427
12. 7513	12. 3517	12. 1622	12. 1067
13. 2551	13. 8308	13. 7943	13. 9619
14. 506	14. 5853	14. 3112	14. 0046
15. 6991	15. 5497	15. 5285	15. 7749
16. 8909	16. 9172	16. 1656	16. 8173
17. 9593	17. 2858	17. 602	17. 8687
18. 5472	18. 7572	18. 263	18. 0844
19. 1386	19. 7537	19. 6541	19. 3998
20. 1493	20. 3804	20. 6892	20. 2599

求多元回归方程系数，数据如上表所示。

```

clear
clc
%load pz.txt %原始数据存放在纯文本文件 pz.txt 中
%两种数据录入方式，选择一种就好了
pz=[1.7547 1.2575 1.5678 1.7482
2.276 2.8407 2.0759 2.4505
3.6797 3.2543 3.054 3.0838
4.6551 4.8143 4.5308 4.229
5.1626 5.2435 5.7792 5.9133
6.119 6.9293 6.934 6.1524
7.4984 7.35 7.1299 7.8258
8.9597 8.1966 8.5688 8.5383
9.3404 9.2511 9.4694 9.9961
10.5853 10.616 10.0119 10.0782
11.2238 11.4733 11.3371 11.4427
12.7513 12.3517 12.1622 12.1067
13.2551 13.8308 13.7943 13.9619
14.506 14.5853 14.3112 14.0046
15.6991 15.5497 15.5285 15.7749
16.8909 16.9172 16.1656 16.8173
17.9593 17.2858 17.602 17.8687
18.5472 18.7572 18.263 18.0844
19.1386 19.7537 19.6541 19.3998
20.1493 20.3804 20.6892 20.2599];%前 3 个为自变量，后 1 个为因变量
mu=mean(pz);sig=std(pz); %求均值和标准差
rr=corrcoef(pz); %求相关系数矩阵
data=zscore(pz); %数据标准化
n=3;m=1; %n 是自变量的个数,m 是因变量的个数
x0=pz(:,1:n);y0=pz(:,n+1:end);%读取 pz 矩阵中的自变量和因变量
e0=data(:,1:n);f0=data(:,n+1:end);%读取数据标准化后的自变量和因变量数值
num=size(e0,1);%求样本点的个数
chg=eye(n); %w 到 w*变换矩阵的初始化
r=[];%设置空矩阵
for i=1:n
%以下计算 w, w*和 t 的得分向量,
matrix=e0'*f0*f0'*e0;
[vec,val]=eig(matrix); %求特征值和特征向量
val=diag(val); %提出对角线元素
[val,ind]=sort(val,'descend');
w(:,i)=vec(:,ind(1)); %提出最大特征值对应的特征向量
w_star(:,i)=chg*w(:,i); %计算 w*的取值
t(:,i)=e0*w(:,i); %计算成分 ti 的得分
alpha=e0'*t(:,i)/(t(:,i)'*t(:,i)); %计算 alpha_i
chg=chg*(eye(n)-w(:,i)*alpha'); %计算 w 到 w*的变换矩阵
e=e0-t(:,i)*alpha'; %计算残差矩阵
e0=e;
%以下计算 ss(i)的值
beta=[t(:,1:i),ones(num,1)]\f0; %求回归方程的系数
beta(end,:)=[]; %删除回归分析的常数项
cancha=f0-t(:,1:i)*beta; %求残差矩阵
ss(i)=sum(sum(cancha.^2)); %求误差平方和
%以下计算 press(i)
for j=1:num

```

```

t1=t(:,1:i);f1=f0;
she_t=t1(j,:);she_f=f1(j,:); %把舍去的第 j 个样本点保存起来
t1(j,:)=[];f1(j,:)=[]; %删除第 j 个观测值
beta1=[t1,ones(num-1,1)]\f1; %求回归分析的系数
beta1(end,:)=[]; %删除回归分析的常数项
cancha=she_f-she_t*beta1; %求残差向量
press_i(j)=sum(canचा.^2);
end
press(i)=sum(press_i);%
if i>1
Q_h2(i)=1-press(i)/ss(i-1);
else
Q_h2(1)=1;
end
if Q_h2(i)<0.0975
fprintf('提出的成分个数 r=%d',i);
r=i;
break
end
press(i)=sum(press_i);
if i>1
Q_h2(i)=1-press(i)/ss(i-1);
else
Q_h2(1)=1;
end
if Q_h2(i)<0.0975
fprintf('提出的成分个数 r=%d',i);
r=i;
break
end
end
beta_z=[t(:,1:r),ones(num,1)]\f0; %求 Y 关于 t 的回归系数
beta_z(end,:)=[]; %删除常数项
xishu=w_star(:,1:r)*beta_z; %求 Y 关于 X 的回归系数，且是针对标准数据的回归系数，每一列是一个回归方程
mu_x=mu(1:n);%读取均值矩阵中的自变量的均值
mu_y=mu(n+1:end);%读取均值矩阵中的因变量的均值
sig_x=sig(1:n);sig_y=sig(n+1:end);%标准化
for i=1:m
ch0(i)=mu_y(i)-mu_x./sig_x*sig_y(i)*xishu(:,i); %计算原始数据的回归方程的常数项
end
for i=1:m
xish(:,i)=xishu(:,i)./sig_x*sig_y(i); %计算原始数据的回归方程的系数，每一列是一个回归方程
end
%最终结果
sol=[ch0;xish] %显示回归方程的系数，每一列是一个方程，每一列的第一个数是常数项，除第行外依次为 x1、x2。。的系数
save mydata x0 y0 num xishu ch0 xish%保存数据

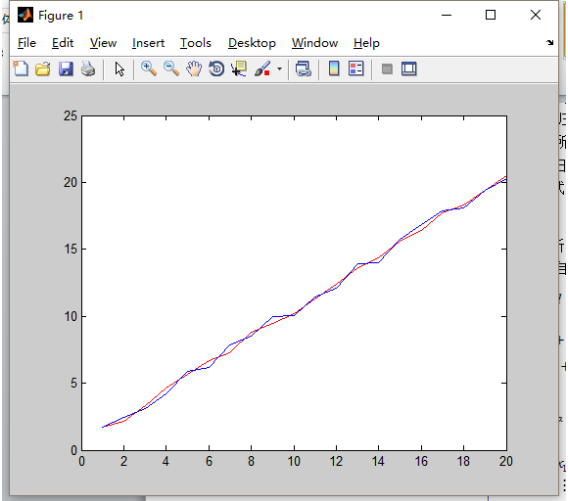
```

提出的成分个数 r=2

```

sol =
    0.0959    %常数
    0.3735    %x1 系数
   -0.0663    %x2 系数

```

0.6871	%x3 系数
计算因变量数据并作检验分析图	
<pre> sol(1,:)=[] A=sol;%删除 sol 矩阵第一行的常数 %读取 pz 矩阵中的前三列 qw=pz; qw(:,4)=[] B=qw; %读取 pz 矩阵中的最后一列 we=pz; we(:,1:3)=[]; F=we; [a,b]=size(pz)%读取 pz 矩阵维数 C=ones(a,1)*ch0;%构造常数项矩阵 D=B*A; E=D+C;%计算的因变量数据 E=E'; z=1:20; plot(z,E,'r-',z,F,'b-')%红线为计算的数据，蓝色为原始数据 </pre>	
<p style="text-align: center;">检验图</p> 	
查看主成分	
<pre> %查看主成分 vec w(:,i)=vec(:,ind(1)) </pre>	
<p>表中数据运行出来，自变量有两个主成分，但均为 x1</p> <pre> vec =      0.0755   -0.9859    0.0874    -0.7422    0.0446    0.6701     0.6659    0.1615    0.7371  &gt;&gt; w(:,i)=vec(:,ind(1))  w =      0.0755    0.0755    -0.7422   -0.7422     0.6659    0.6659 </pre>	