中國農業大學
CHINA AGRICULTURAL UNIVERSITY

R在近红外光谱分析中的应用

在测定玉米营养物质时，需要通过化学实验来测定其各种营养成分的含量，而化学实验的成本比较高，而且会造成环境污染。近红外谱区（**800~2500nm**）主要由分子振动的倍频与和频产生，谱带比较宽，吸收强度较弱，从该区间可得到分子中含氢基团的震动信息，其信息量可成功用于定量分析（**1**）。

中國農業大學
CHINA AGRICULTURAL UNIVERSITY

数据说明：X: **The wavelength range is 1100- 2498nm at 2 nm intervals (700 channels)**

X: 数据收集:近红外光谱仪**(NIR spectrometers)**

Y: **The moisture, oil, protein and starch values for each of the samples**

中國農業大學
CHINA AGRICULTURAL UNIVERSITY

如何利用X的信息来预测Y,即 moisture, oil, protein and starch 的 含量?

由于样本资料矩阵X是80*700,即变量数远远大于
样本数目,所以需要降维

我们的方法:

主成分回归, 岭回归, 偏最小二乘法回归, Lasso

模型比较:模型的解释性,预测准确性(十折交叉验证)

中國農業大學
CHINA AGRICULTURAL UNIVERSITY

主成分回归(PCR):

它的数学处理方法是：将原来的**p**个指标作线性组合，作为新的综合指标。如果将选取的第一个线性组合即为第一个综合指标记作 **F1**，希望其尽可能多的包含原来的信息，这里的信息用 的方差来刻画，即 **Var(F1)**越大,**F1**包含的信息越多。如果第一主成分不足够表达原来**p**个指标的信息，就考虑选取第**2**主成分 ，且 **F1**中有的信息不需要再出现在 中，即**Cov(F1,F2) =0**，依次类推.可以构造出第**3,4,5…**主成分,这些主成分之间互相不相关,它们的方差依次递减

并且可以证明当**F1**最大时候,**F1**关于关于指标的线性组合系数是**X**协差矩阵的最大特征值所对应的特征向量

中國農業大學
CHINA AGRICULTURAL UNIVERSITY

读入数据

```
X<-read.table("X.txt")
dim(X)
[1] 80 700
```

考虑提取主成分个数:

```
a=cov(X)                          ##############    x的协差阵
eigen(a)$value    ##################求x的协差阵的特征值
eigen(a)$value[1]/sum(eigen(a)$value)
[1] 0.9907834     ###########说明提取第一主成分已经足够
```

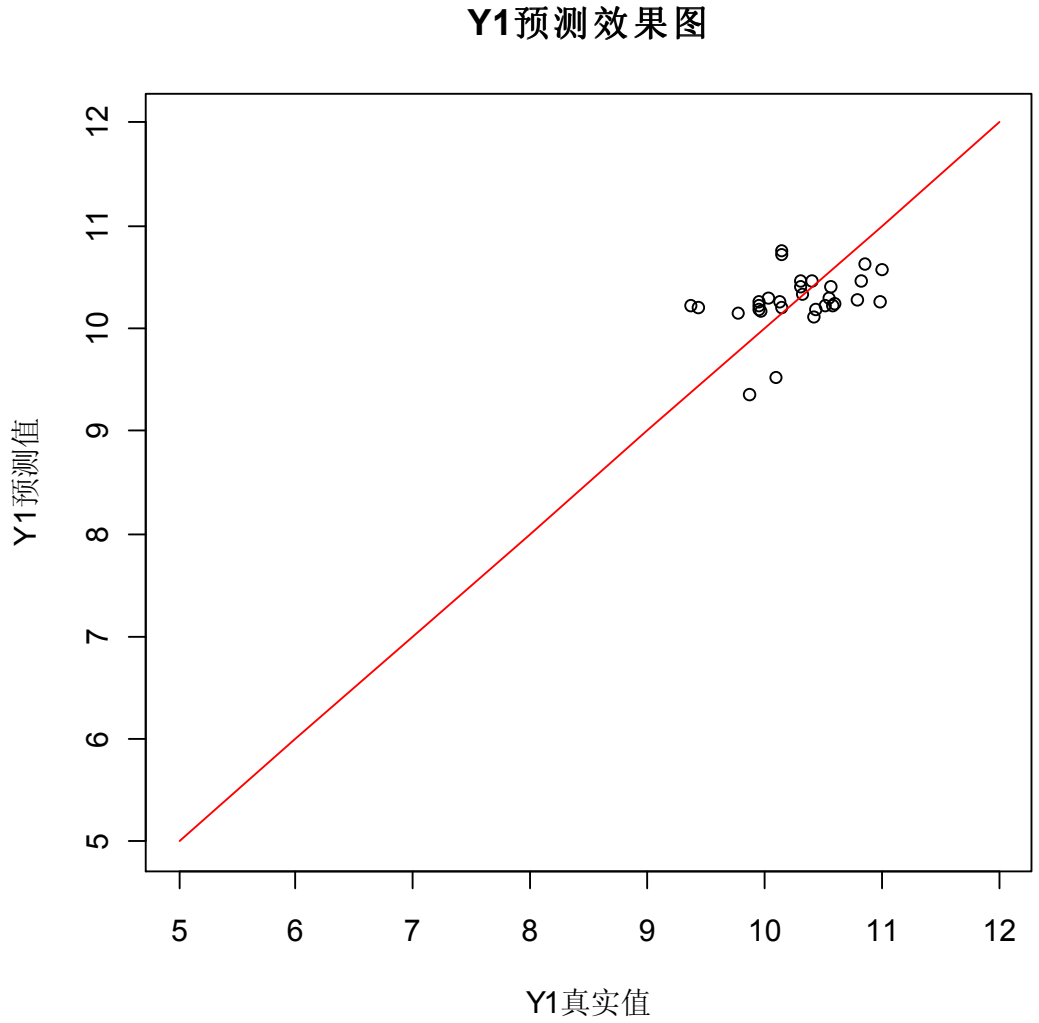
将变量维数从700降到1:

```
b=eigen(a)$vector[,1] ####第一特征值所对应的特征向量
fcomp <- numeric(80)
for(i in 1:80)
{
  fcomp[i] <- sum(X[i,]*b)
}
fcomp
fcomp<-as.matrix(fcomp)  ####新的样本数据阵80*1
objective<-lm(Y[1:50,1]~fcomp[1:50]) ####建模型
Call:
lm(formula = Y[1:50, 1] ~ fcomp[1:50])
Coefficients:
(Intercept)  fcomp[1:50]
    13.6506       0.3032
```

```
b<-as.matrix(objective$coef)      #####求模型中的回归系数
Xtest<-cbind(1,fcomp[51:80])
fit<-Xtest%*%b     ######代入testgroup数据

rss<-sum((fit-Y[51:80,1])^2) ###残差平方和
[1] 4.885731


plot(xlim=c(5,12),ylim=c(5,12),fit~Y[51:80,1],
xlab="Y1真实值",ylab="Y1预测值",main="Y1预测效果图")
lines(5:12,5:12,col="red")
```


Y1预测效果图
Y1预测值
Y1真实值
5
6
7
8
9
10
11
12

回归系数的 的岭估计为 $\beta(k)=(X'X+kI)^{-1}X'y$
这里 k>0 是可选参数，称为岭参数或者偏参数。如果 取k与实验数据 Y无关的常数，则 $\beta(k)$为线性估计。

我们知道，“ X'X的特征根很小”，等价于设计阵 X之间存在共线性关系，并且 X'X有几个特征值很小，设计阵 X 就存在几个复共线性关系。（线性模型引论，王松桂等）

在最小二乘法（LS）估计中，β 的均方误差（MSE）为 $\sigma^2\sum_1^p 1/\lambda_i$
所以复共线性是LS变坏的原因。

与LS估计 相比，岭回归是把 XX' 换成了XX'+kI 得到的，直观上想，当 X呈病态时，XX' 的特征值至少有一个非常接近0，而 XX'+kI的特征根 λ_i+k，接近于0的程度就会改善。

读入数据，X为变量，Y为应变量

```
X = as.matrix(read.table("X.txt"))
Y = as.matrix(read.table("Y.txt"))
```

选取10个k的初始值，

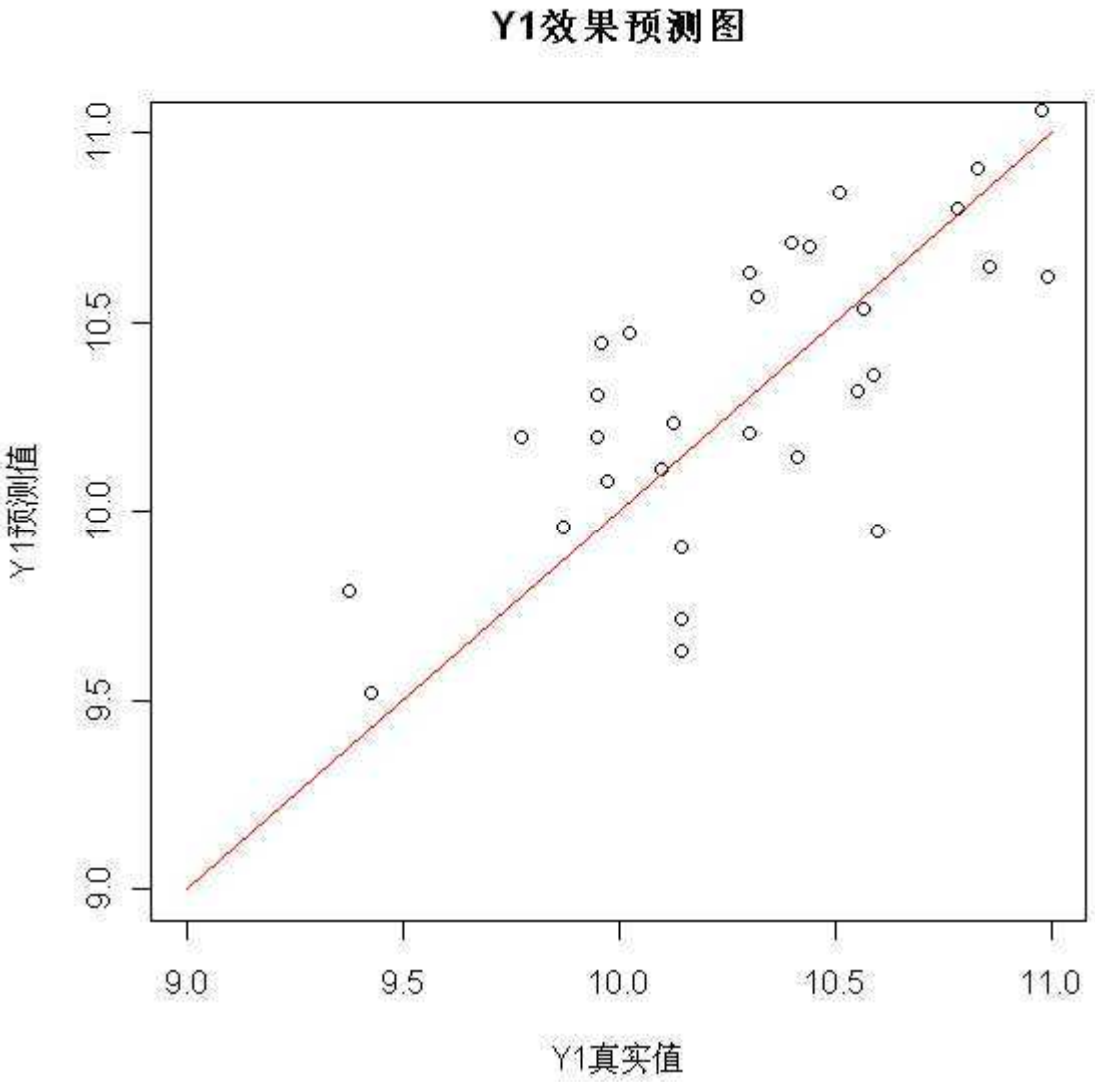
```
 k= seq(0.0001,0.001,0.0001)
```

下面在残差平方和意义下，利用十折交叉效应选取最好的k

```
nreps = 10
RSS=numeric(nreps)
for(i in 1:10)
   {
    subsample = sample(1:80)
    for(j in 1:nreps)
       {
        index = subsample[(((j-1)*8+1):(j*8))]
        X1=X[-index,]
        X2=X[index,]
        coef=solve(t(X1)%*%X1+k[i]*diag(700))%*%t(X1)%*%Y[-
index,1]   # k==k[i]时回归系数
        RSS[i] = RSS[i]+sum((Y[index,1] - X2%*%coef)^2)
       }
    RSS[i] = RSS[i]/10 #####RSS[i] 表示在k==k[i]时候的残差平方和
   }
```

index = which(RSS==min(RSS)) ###则最优的k所在位置为
由此，可以得到岭回归模型
pred=X%*%solve(t(X)%*%X+lambda[index]*diag(700))%*%t(X)%*%Y[,1]

预测效果：
plot(cbind(Y[51:80,1],pred[51:80,]),xlim=c(9,11),ylim=c(9,11),xlab="Y1真实值",ylab="Y1预测值",main="Y1效果预测图")
lines(9:11,9:11,col="red")


Y1效果预测图
Y1预测值
Y1真实值
9.0
9.5
10.0
10.5
11.0

偏最小二乘回归分析在建模过程中集中了主成分分析，典型相关分析和线性回归分析方法的特点，它提供了一种多对多对的线性回归建模的方法，特别当两组的变量特别多，且都存在多重相关性，而观测数据的样本量又都较少时，偏最小二乘法具有传统经典回归分析等没有的优点。

偏最小二乘法的基本做法是：

考虑 P个因变量 $Y_1, Y_2, \ldots Y_p$ 与m 个自变量 $X_1, X_2, \ldots X_m$ 的建模问题。

首先，在自变量集中提取第一主成分 T1（ T1是 $X_1, X_2, \ldots X_m$ 的线性组合，尽可能多地提取原自变量集中的信息），同时在因变量中提取第一主成分 U1，并要求 T1与 U1相关程度达最大。然后建立 $Y_1, Y_2, \ldots Y_p$ 与 T1的回归方程。如果回归方程已经达到满意的精度，则算法终止，否则继续第二主成分的提取，直到达到满意精度为止。若最终对自变量集提取 r个成分T1 ,T2,T3…Tr偏最小二乘法将建立 $Y_1, Y_2, \ldots Y_p$ 与 T1 ,T2,T3…Tr 的回归方程，然后再表示为 $Y_1, Y_2, \ldots Y_p$ 与原自变量的回归方程。（应用多元统计分析，高慧璇）

在R中可以直接调用pls()

```
library("pls")
```

读入数据：

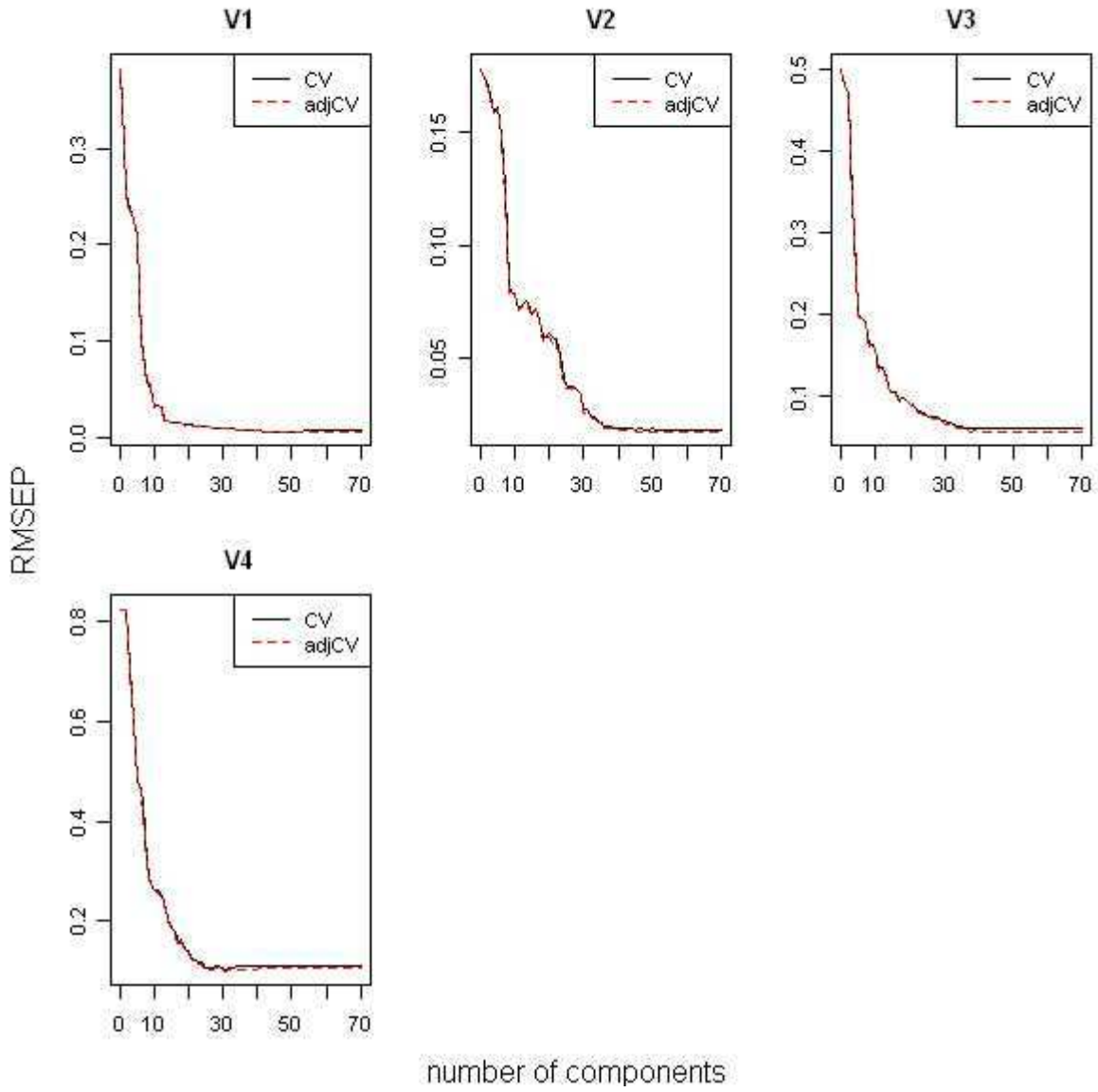
```
X<-as.matrix(read.table("X.txt",as.is=T))
Y<-as.matrix(read.table("Y.txt",as.is=T))
corn<-data.frame(X,Y)      #####pls()调用时候需要data.frame
corntrain<-corn[1:50,]      #######以前60个为实验组
```

对实验组数据进行偏最小二乘回归，刚开始不知道应该提取多少和主成分对，不妨取大点，在这里，我们取了70

```
plsr<-plsr(Y~X,ncomp=70,data=corntrain,validation="CV")
```

通过作图，根据root mean squared error of prediction(RMSEP)判断多少和主成分对数合理，

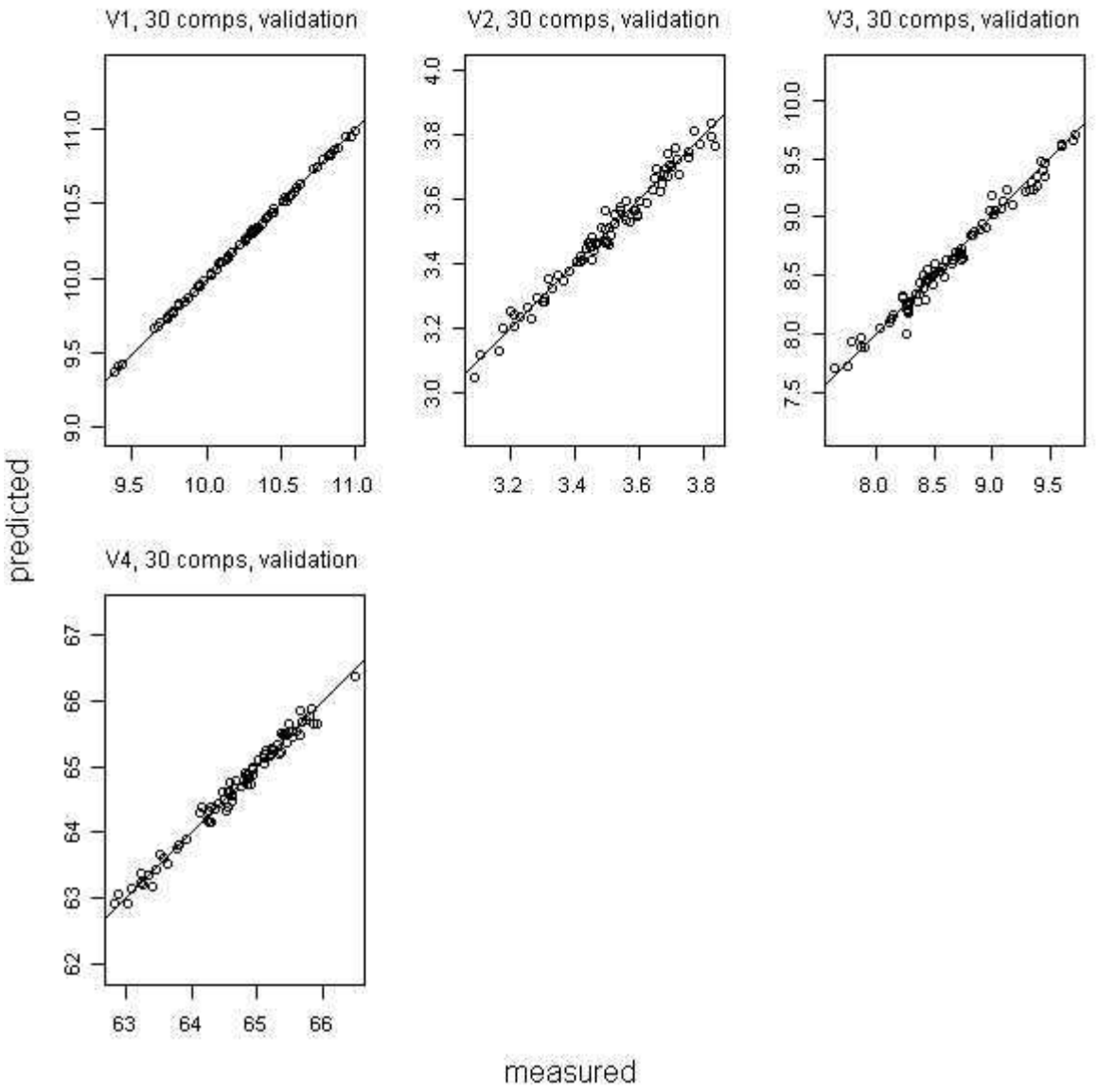
```
plot(RMSEP(plsr),legendpos="topright")
```


V1
V2
V3
V4
CV
adjCV
RMSEP
number of components
0 10 30 50 70
0.0 0.1 0.2 0.3
0.05 0.10 0.15
0.1 0.2 0.3 0.4 0.5
0.2 0.4 0.6 0.8

从图中看，30个主成分数足够了,这比主成分回归中提取的成分数要多。

确定主成分数后，可以画拟合的效果图

```
plot(plsr,ncomp=30,asp=1,line=T)
```


V1, 30 comps, validation
V2, 30 comps, validation
V3, 30 comps, validation
V4, 30 comps, validation
predicted
measured
9.0
9.5
10.0
10.5
11.0
3.0
3.2
3.4
3.6
3.8
4.0
7.5
8.0
8.5
9.0
9.5
10.0
62
63
64
65
66
67

以后30个作为验证组（test group）

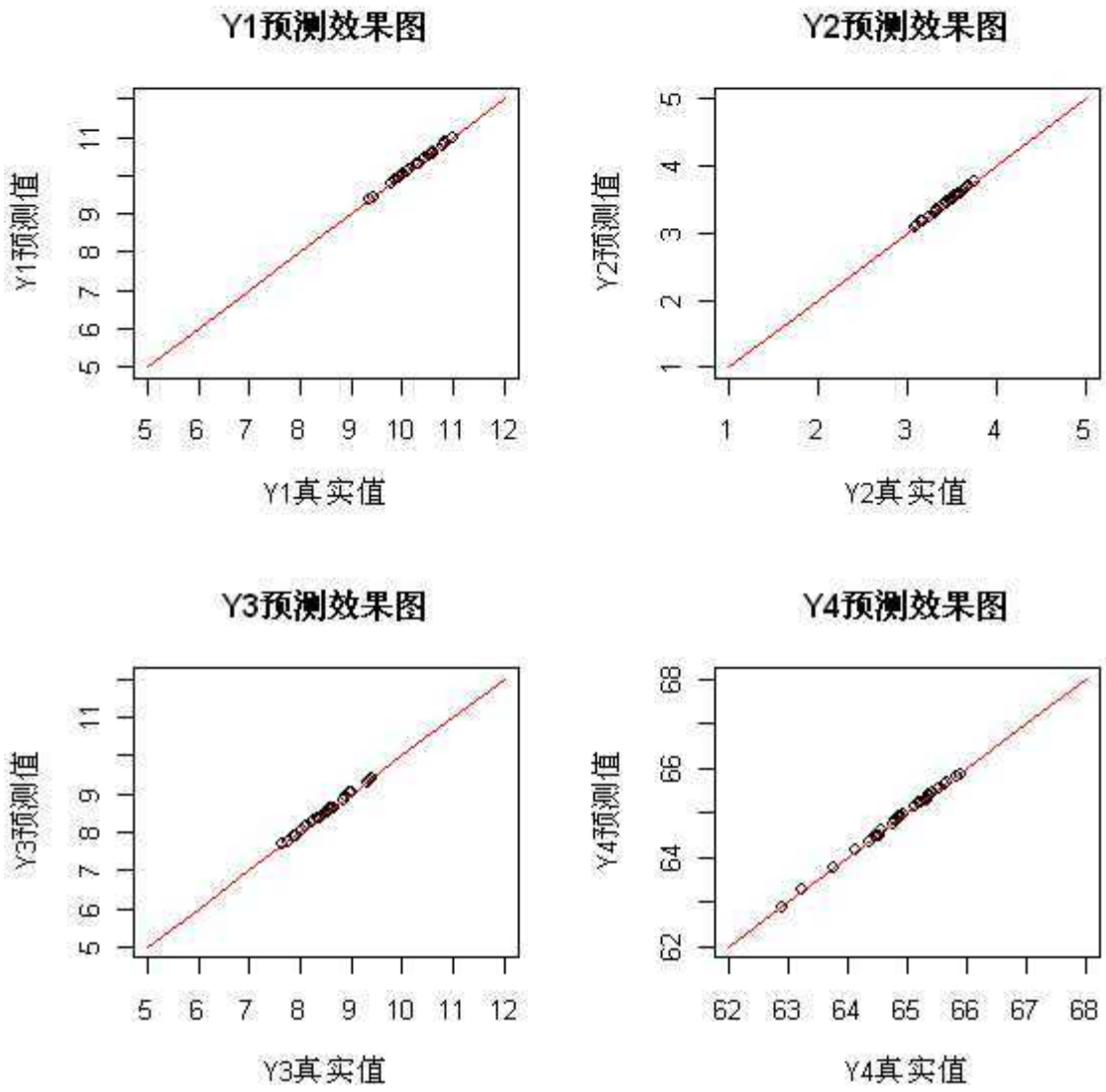
```
corntest<- as.matrix (X[51:80,])
pre<- predict(plsr,corntest,ncomp=30)
```

作验证组的预测效果图

```
par(mfrow=c(2,2))      ######在同一窗口中画四幅图形
plot(xlim=c(5,12),ylim=c(5,12), pre[,1,] ~Y[51:80,1],
xlab="Y1真实值",ylab="Y1预测值",main="Y1预测效果图")

lines(5:12,5:12, col= “red”)# 直线y=x, 点越接近它说明预 测越好

plot(xlim=c(1,5),ylim=c(1,5),pre[,2,]~Y[51:80,2],xlab="Y
2真实值",ylab="Y2预测值",main="Y2预测效果图")
lines(1:5,1:5,col= "red")
plot(xlim=c(5,12),ylim=c(5,12),pre[,3,]~Y[51:80,3],xlab=
"Y3真实值",ylab="Y3预测值",main="Y3预测效果图")
lines(5:12,5:12,col= "red")
plot(xlim=c(62,68),ylim=c(62,68),pre[,4,]~Y[51:80,4],xla
b="Y4真实值",ylab="Y4预测值",main="Y4预测效果图")
lines(62:68,62:68,col="red")
```


Y1预测效果图
Y1预测值
Y1真实值
5 6 7 8 9 10 11 12
5 6 7 8 9 10 11
Y2预测效果图
Y2预测值
Y2真实值
1 2 3 4 5
1 2 3 4 5
Y3预测效果图
Y3预测值
Y3真实值
5 6 7 8 9 10 11 12
5 6 7 8 9 10 11
Y4预测效果图
Y4预测值
Y4真实值
62 63 64 65 66 67 68
62 64 66 68

5.1 Lasso

Lasso 方法用模型系数的绝对值函数作为惩罚来压缩模型系数，使绝对值小的系数自动压缩为0，从而同时实现显著性变量的选择和对应参数的估计。

在简单的线性模型中，

$$Y = X\beta + e$$

其中

$$y = (y_1, y_2, \ldots y_n)^T, x_j = (x_{1j}, x_{2j}, \ldots x_{nj})^T$$

$$j = 1,2,\ldots p$$

β

$$x = (x_1, x_2, \ldots x_p)$$

β 是 p维列向量，为待估参数，误差向量 e满足 $E(e) = 0, Var(e) = \sigma^2$

并且假定：

$$E(y \mid x) = \beta_1 x_1 + \ldots + \beta_p x_p$$

注意该模型是稀疏集（即 β 中有很多系数为0。变量选择的目的就是要根据获取的数据来识别哪些数据是0，并且估计其他非0参数，即寻找稀疏模型。

对于线性模型选择实际上可以考虑成如下问题：

$$\beta = \underset{\beta}{\operatorname{argmin}} ||y - X\beta||^2, \sum_{1}^{p} |\beta_l| <= t$$

在R中调用lar()

```
library("lars")
```

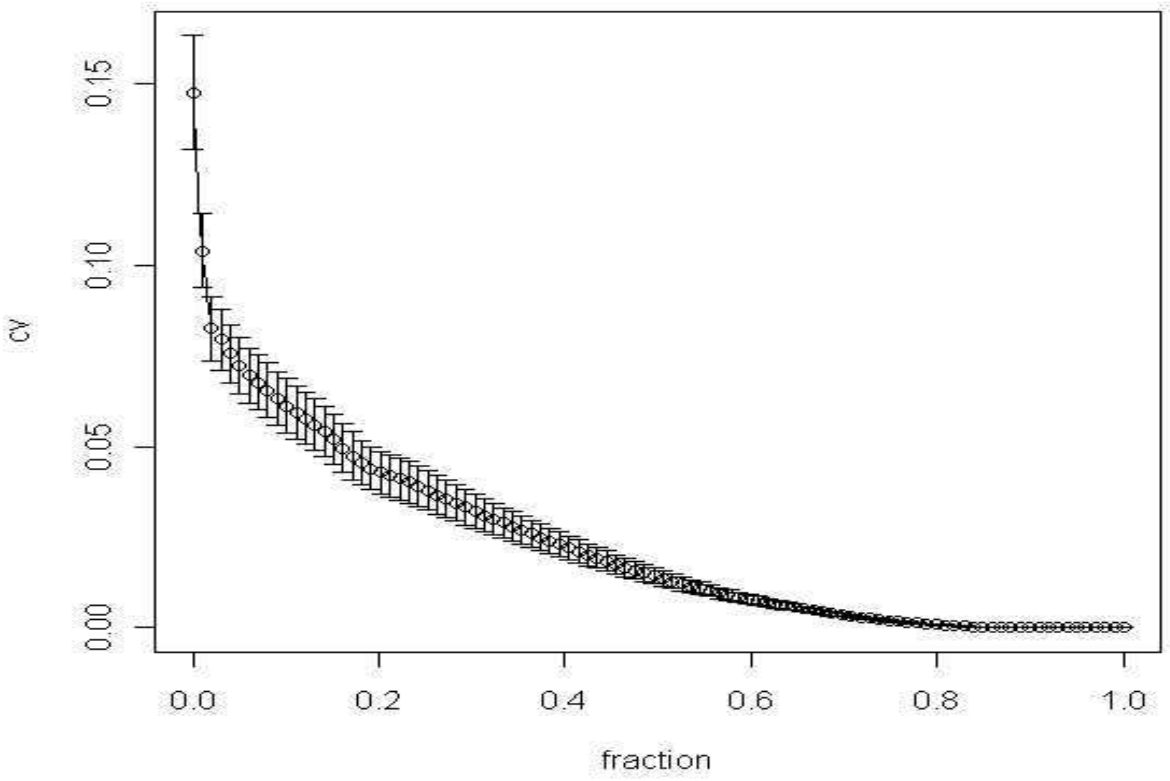
读入数据，X是变量，Y是因变量，转换成沮阵

```
X = as.matrix(read.table("X.txt",header=FALSE))
Y = as.matrix(read.table("Y.txt",header=FALSE))
```

前50个样本作为实验组（train group）,后30个样本作为验证组（test group）

```
train=X[1:50,]
test=X[51:80,]
```

```
lars1=lars(train,Y[1:60,1],type="lasso",max.step=80,t
race=F,use.Gram=F)
cv.lars(X,Y[,1],type="lasso",use.Gram=FALSE)
```



预测验证组数据：

```
pre= predict.lars(lars1, test)
```

```
summary(pre)
       Length Class  Mode
s         81   -none- numeric
fraction  81   -none- numeric
mode       1   -none- character
fit     2430   -none- numeric
```

而由前面的图形可以知道，在fraction=0.8可能取道比较好的预测值

```
attach(pre)                    ###R可以直接读s,fraction,mode,fit这些变量
coef = coef.lars(lars1)        ###这里有81组回归系数
best=s[fraction==0.8]          ####选出fraction==0.8时候对应的最好的s
coef.best=coef[best,]          ###对应fraction==0.8的回归系数
```

哪些回归系数非零呢?

```
which(coef.best!=0)
```

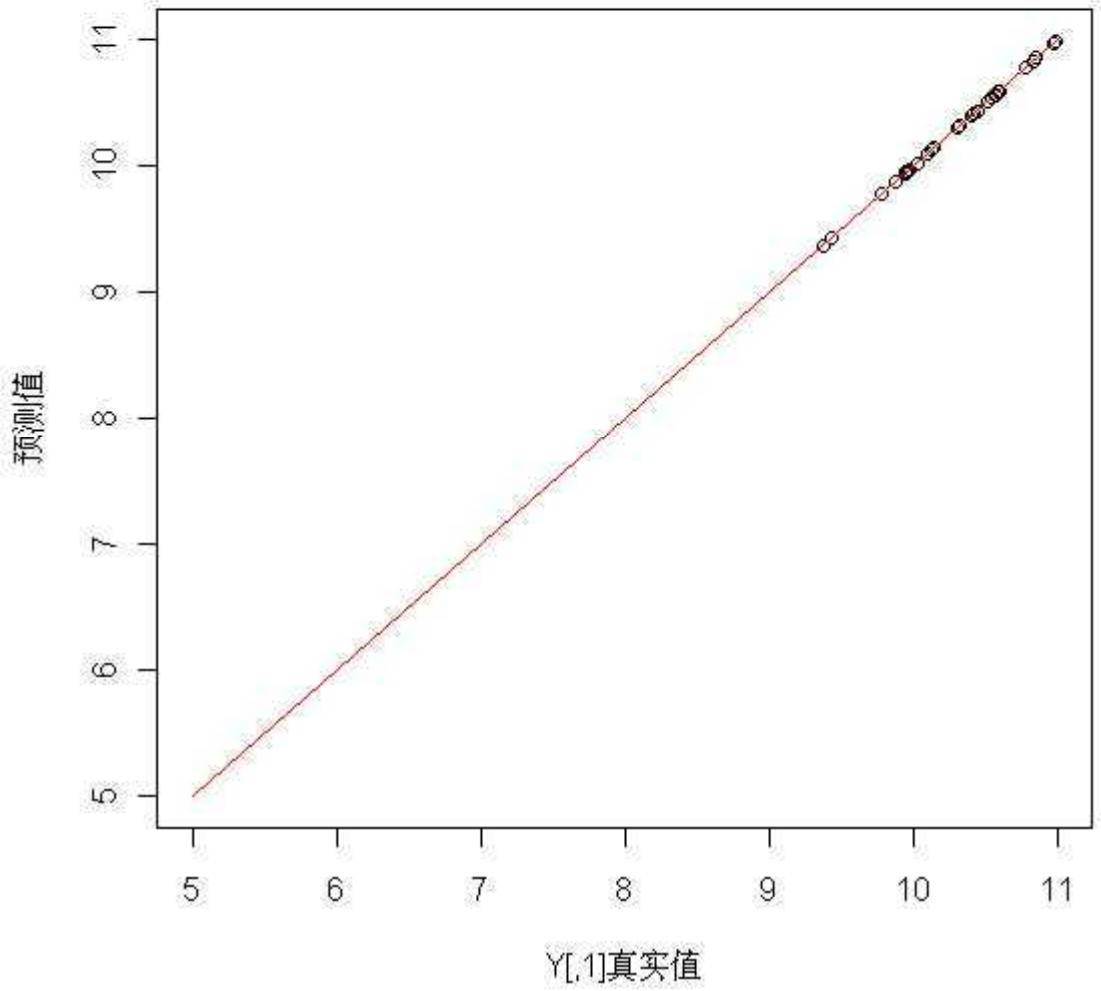
```
V9  V32 V102 V155 V201 V404 V405 V407 V413 V417 V503 V505 V513 V578 V652  9   32
102  155  201  404  405  407  413  417  503  505  513  578  652
```

对应预测价为值

```
fit.best=fit[,best]                                    ###在fraction==0.08时候的预测值
```

作出预测值和真实的效果图

```
plot(xlim=c(5,11),ylim=c(5,11),fit.best~Y[51:80,1],xlab="Y[,
1]真实值",ylab="预测值")
lines(5:11,5:11,col="red")
```


预测值
Y[,1]真实值
5
6
7
8
9
10
11

模型比较

从变量选择角度看,Lasso 自动选择15个变量.

从预测准确性来讲:
方法:分批十折交叉验证:
每次扣留连续的8个观测数据作为检验数据组

读入数据

```
X=as.matrix(read.table("X.txt",header=FALSE))
Y=as.matrix(read.table("Y.txt",header=FALSE))
rss=0                              #####  rss定义为残差平方和
```

PCA:

```
for (j in 1:10)
 {
m<-8*j-7
n<-8*j
train<-fcomp[-c(m:n),]
test<-fcomp[c(m:n),]
objective<-lm(Y[-c(m:n),1]~fcomp[-c(m:n)])
b<-objective$coef
b<-as.matrix(b)
Xtest<-cbind(1,fcomp[c(m:n)])
fit<-Xtest%*%b
rss<-rss+sum((fit-Y[c(m:n),1])^2)
}
rss/10
[1] 0.7973812
```

```
lambda=0.0001      #####由之前的结果知道lambda=0.0001最好
rss=0
coef = matrix(0,700,1)
for (j in 1:10)
{
a<-8*j-7
b<-8*j
train<-X[-c(a:b),]      #########  去除第a:b个做实验组
test<-X[c(a:b),]        #########  a:b作为验证组
coef=solve(t(train)%*%train+lambda*diag(700))%*%t(train)%*%Y[
-c(a:b),1]                  ########第j次建立模型所得到的回归系数
 rss<-rss+sum((test%*%coef-Y[c(a:b),1])^2)    ###求10次的残差平方和
}
rss/10
[1] 0.9261054
```

看偏最小二乘法:

```
rss=0
for (j in 1:10)
    {
    a<-8*j-7
    b<-8*j
    corntrain<-corn[-c(a:b),]
    corntest<- as.matrix (X[c(a:b),])
    plsr<-
plsr(Y~X,ncomp=30,data=corntrain,validation="CV")
    pre<- predict(plsr,corntest,ncomp=30)
    rss=sum((pre[,1,]-Y[c(a:b),1])^2)+rss
    }
rss

>rss
[1] 0.02131073
```

显然偏最小二乘法的效果是最好的。

Lasso:

```
X<-(X-apply(X,2,mean))/(apply(X,2,var))^(0.5)  ### 把X进行标准化
rss=0
for (j in 1:10){
a<-8*j-7
b<-8*j
train<-X[-c(a:b),]
test<-X[c(a:b),]
lars1=lars(train,Y[-
c(a:b),1],type="lasso",max.step=80,trace=F,use.Gram=F)##建模
pre= predict.lars(lars1, test)          ##预测实验组
attach(pre)
best=s[fraction==0.8]
fit.best=fit[,best]
rss=rss+sum((fit.best-Y[c(a:b):1])^2)
}
rss
```

99.83 ######

有交叉验证结果可以看出，从残差平方和标准看，PLSR的效果是最好的,而Lasso最差,这可能与变量间的相关性以及变量数目700>>80有关.通过文献,我们发现,对于变量相关性较高,p>>n的情况, Lasso的改进方法, Elastic Net可能会有更好的表现.

从模型解释性看,Lasso是最好的,它自动选择了15个变量,可能对农业工作者提供宝贵信息..

中國農業大學

CHINA AGRICULTURAL UNIVERSITY

THANK YOU

^ ^

—