PLSR for corn V2 PDF Markdown

Hongwei Peng

25/06/2019

```
library(R.matlab)
```

```
## R.matlab v3.6.2 (2018-09-26) successfully loaded. See ?R.matlab for help.

##
## Attaching package: 'R.matlab'

## The following objects are masked from 'package:base':
##
##     getOption, isOpen
```

```
library(pls)
```

```
## Warning: package 'pls' was built under R version 3.5.2

##
## Attaching package: 'pls'

## The following object is masked from 'package:stats':
##
##     loadings
```

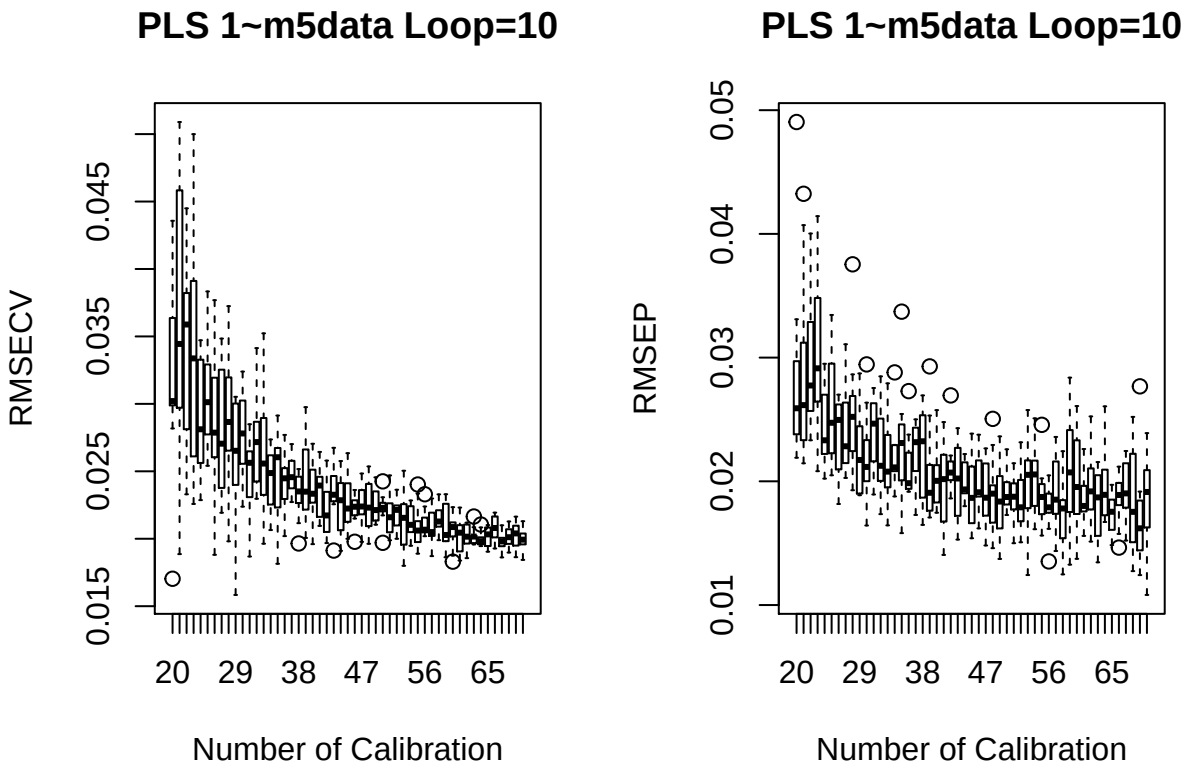
```
library(lars)
```

```
## Loaded lars 1.2
```

```
library(ggplot2)
rawdata <- readMat("/Users/hongwei/Downloads/corn.mat")
m5data <- rawdata$m5spec$data
mp5data <- rawdata$mp5spec$data
mp6data <- rawdata$mp6spec$data
propvals <- rawdata$propvals$data

corn_PLS=function(n){                                   #n is the number of calibration
  NV <- 10                                              #number of variables
  sample <- sample(1:80)                                #set random order; the begin of reset order
  DF <- data.frame(NIR = I(m5data),                     #input data
                   y=propvals[,1])
  class(DF$NIR) <- "matrix"                             # just to be certain, it was "AsIs"
  #str(DF)                                              #check point
  DF$train <- rep(FALSE, 80)
  DF$train[sample<=n] <- TRUE                           #chose calibration
  corn.pls <- plsr(y ~ NIR, data = DF, ncomp = NV, validation="LOO", jackknife = TRUE, subset = train)
  #summary(corn.pls,what="all")                         #check point
  RMSECV <- RMSEP(corn.pls)$val[1,1,NV]
  #print(RMSECV)                                        #check point
  predict <- predict(corn.pls, ncomp = NV, newdata = DF[!DF$train,])
  RMSEP <- sqrt(sum((predict-DF[!DF$train,]$y)^2)/(80-n))
  #print(RMSEP)                                         #check point
  #plot(R2(corn.pls))                                   #check point
  return(cbind(RMSECV,RMSEP))                           #return 1x2matrix
```

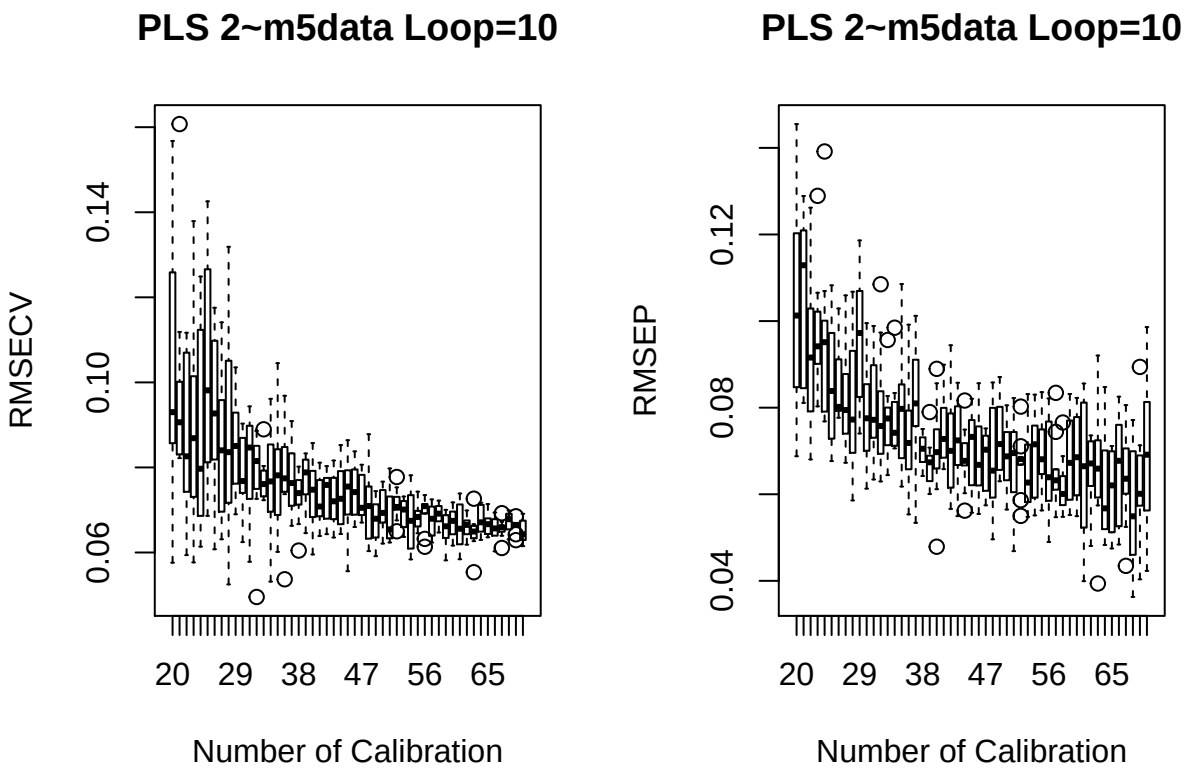
```
}
#corn_PLS(n)
n <- as.matrix(rep(20:70,10))                    #the number of calibration, rep(a:b,c): from a to
PlsResult <- apply(n,1,corn_PLS)                 #loop
PlotData <- as.data.frame(cbind(n,t(PlsResult)))  #combind the results
par(mfrow=c(1,2))
boxplot(V2~V1,data=PlotData,xlab="Number of Calibration", ylab="RMSECV",main="PLS 1~m5data Loop=10")
boxplot(V3~V1,data=PlotData,xlab="Number of Calibration", ylab="RMSEP",main="PLS 1~m5data Loop=10")
```




```
library(R.matlab)
library(pls)
library(lars)
library(ggplot2)
rawdata <- readMat("/Users/hongwei/Downloads/corn.mat")
m5data <- rawdata$m5spec$data
mp5data <- rawdata$mp5spec$data
mp6data <- rawdata$mp6spec$data
propvals <- rawdata$propvals$data

corn_PLS=function(n){                            #n is the number of calibration
  NV <- 10                                       #number of variables
  sample <- sample(1:80)                         #set random order; the begin of reset order
  DF <- data.frame(NIR = I(m5data),              #input data
                   y=propvals[,2])
  class(DF$NIR) <- "matrix"                      # just to be certain, it was "AsIs"
  #str(DF)                                       #check point
  DF$train <- rep(FALSE, 80)
  DF$train[sample<=n] <- TRUE                    #chose calibration
  corn.pls <- plsr(y ~ NIR, data = DF, ncomp = NV, validation="LOO", jackknife = TRUE, subset = train)
```

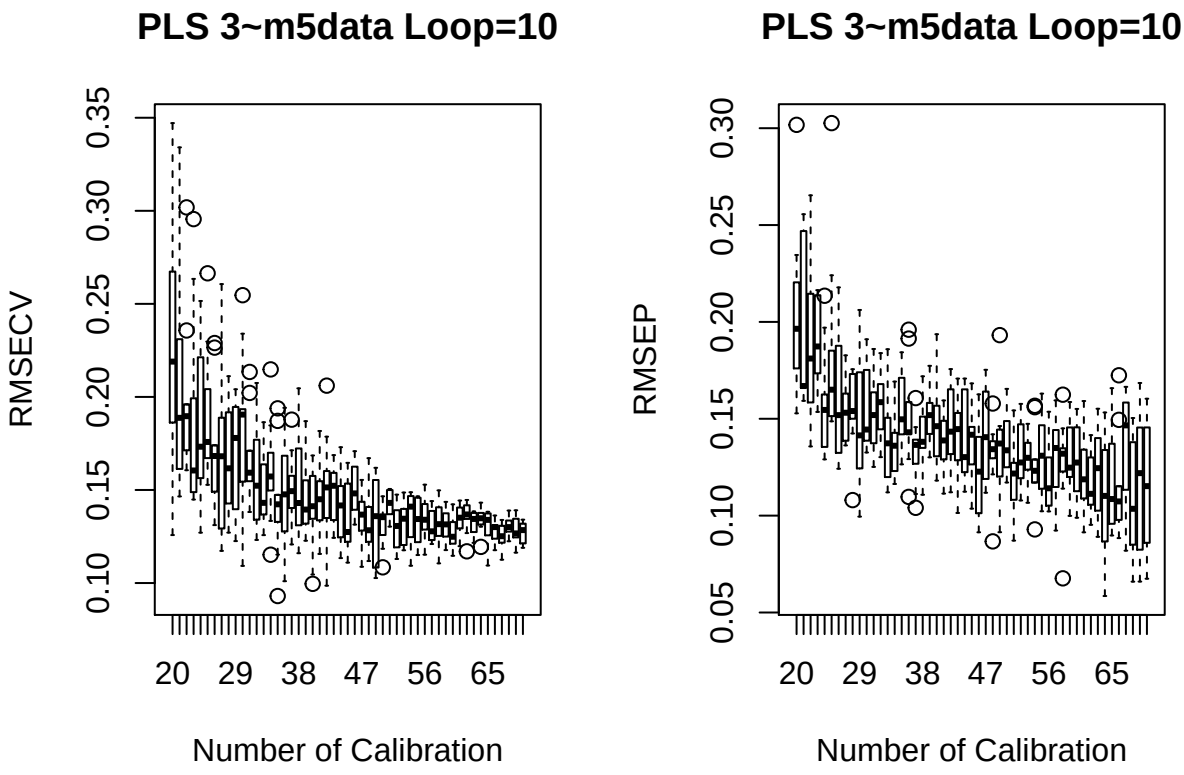
```
  #summary(corn.pls,what="all")                          #check point
  RMSECV <- RMSEP(corn.pls)$val[1,1,NV]
  #print(RMSECV)                                         #check point
  predict <- predict(corn.pls, ncomp = NV, newdata = DF[!DF$train,])
  RMSEP <- sqrt(sum((predict-DF[!DF$train,]$y)^2)/(80-n))
  #print(RMSEP)                                          #check point
  #plot(R2(corn.pls))                                    #check point
  return(cbind(RMSECV,RMSEP))                            #return 1x2matrix
}
#corn_PLS(n)
n <- as.matrix(rep(20:70,10))                            #the number of calibration, rep(a:b,c): from a to
PlsResult <- apply(n,1,corn_PLS)                         #loop
PlotData <- as.data.frame(cbind(n,t(PlsResult)))         #combind the results
par(mfrow=c(1,2))
boxplot(V2~V1,data=PlotData,xlab="Number of Calibration", ylab="RMSECV",main="PLS 2~m5data Loop=10")
boxplot(V3~V1,data=PlotData,xlab="Number of Calibration", ylab="RMSEP",main="PLS 2~m5data Loop=10")
```




```
library(R.matlab)
library(pls)
library(lars)
library(ggplot2)
rawdata <- readMat("/Users/hongwei/Downloads/corn.mat")
m5data <- rawdata$m5spec$data
mp5data <- rawdata$mp5spec$data
mp6data <- rawdata$mp6spec$data
propvals <- rawdata$propvals$data

corn_PLS=function(n){                                    #n is the number of calibration
  NV <- 10                                               #number of variables
```

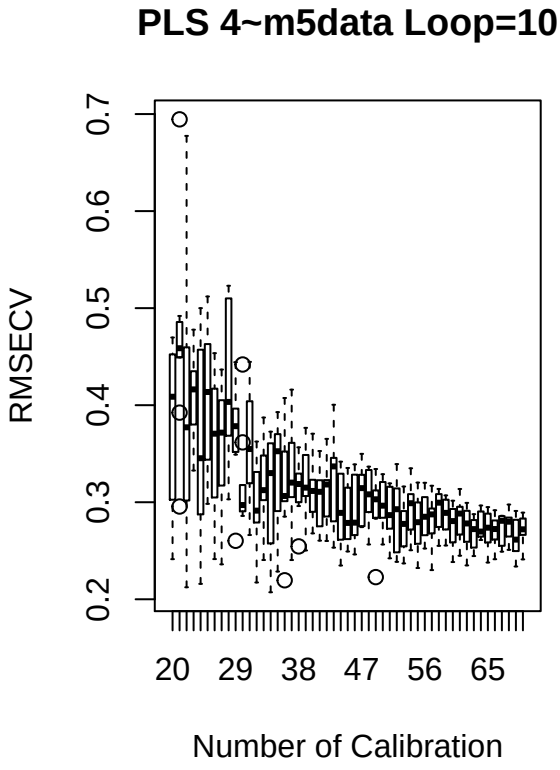
```
  sample <- sample(1:80)                                  #set random order; the begin of reset order
  DF <- data.frame(NIR = I(m5data),                       #input data
                   y=propvals[,3])
  class(DF$NIR) <- "matrix"                               # just to be certain, it was "AsIs"
  #str(DF)                                                #check point
  DF$train <- rep(FALSE, 80)
  DF$train[sample<=n] <- TRUE                             #chose calibration
  corn.pls <- plsr(y ~ NIR, data = DF, ncomp = NV, validation="LOO", jackknife = TRUE, subset = train)
  #summary(corn.pls,what="all")                           #check point
  RMSECV <- RMSEP(corn.pls)$val[1,1,NV]
  #print(RMSECV)                                          #check point
  predict <- predict(corn.pls, ncomp = NV, newdata = DF[!DF$train,])
  RMSEP <- sqrt(sum((predict-DF[!DF$train,]$y)^2)/(80-n))
  #print(RMSEP)                                           #check point
  #plot(R2(corn.pls))                                     #check point
  return(cbind(RMSECV,RMSEP))                             #return 1x2matrix
}
#corn_PLS(n)
n <- as.matrix(rep(20:70,10))                             #the number of calibration, rep(a:b,c): from a to
PlsResult <- apply(n,1,corn_PLS)                          #loop
PlotData <- as.data.frame(cbind(n,t(PlsResult)))          #combind the results
par(mfrow=c(1,2))
boxplot(V2~V1,data=PlotData,xlab="Number of Calibration", ylab="RMSECV",main="PLS 3~m5data Loop=10")
boxplot(V3~V1,data=PlotData,xlab="Number of Calibration", ylab="RMSEP",main="PLS 3~m5data Loop=10")
```

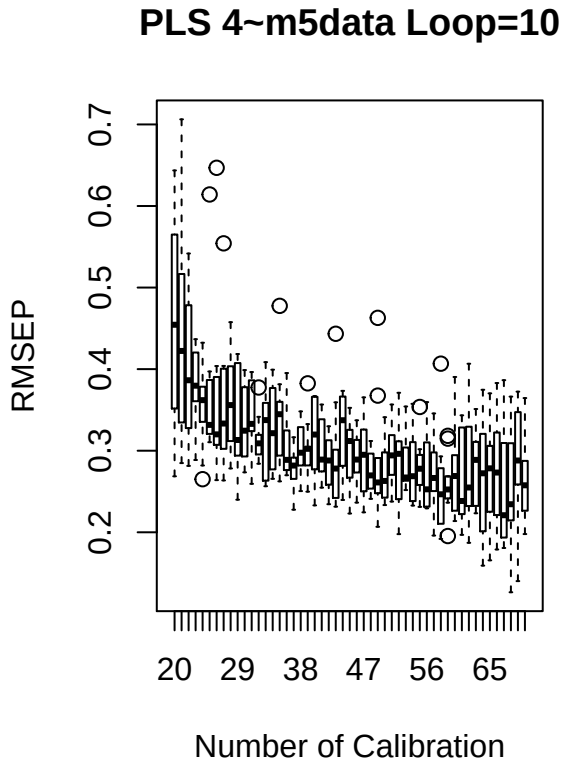



```
library(R.matlab)
library(pls)
library(lars)
library(ggplot2)
```

```
rawdata <- readMat("/Users/hongwei/Downloads/corn.mat")
m5data <- rawdata$m5spec$data
mp5data <- rawdata$mp5spec$data
mp6data <- rawdata$mp6spec$data
propvals <- rawdata$propvals$data

corn_PLS=function(n){                                   #n is the number of calibration
  NV <- 10                                              #number of variables
  sample <- sample(1:80)                                #set random order; the begin of reset order
  DF <- data.frame(NIR = I(m5data),                     #input data
                   y=propvals[,4])
  class(DF$NIR) <- "matrix"                             # just to be certain, it was "AsIs"
  #str(DF)                                              #check point
  DF$train <- rep(FALSE, 80)
  DF$train[sample<=n] <- TRUE                           #chose calibration
  corn.pls <- plsr(y ~ NIR, data = DF, ncomp = NV, validation="LOO", jackknife = TRUE, subset = train)
  #summary(corn.pls,what="all")                         #check point
  RMSECV <- RMSEP(corn.pls)$val[1,1,NV]
  #print(RMSECV)                                        #check point
  predict <- predict(corn.pls, ncomp = NV, newdata = DF[!DF$train,])
  RMSEP <- sqrt(sum((predict-DF[!DF$train,]$y)^2)/(80-n))
  #print(RMSEP)                                         #check point
  #plot(R2(corn.pls))                                   #check point
  return(cbind(RMSECV,RMSEP))                           #return 1x2matrix
}
#corn_PLS(n)
n <- as.matrix(rep(20:70,10))                           #the number of calibration, rep(a:b,c): from a to
PlsResult <- apply(n,1,corn_PLS)                        #loop
PlotData <- as.data.frame(cbind(n,t(PlsResult)))        #combind the results
par(mfrow=c(1,2))
boxplot(V2~V1,data=PlotData,xlab="Number of Calibration", ylab="RMSECV",main="PLS 4~m5data Loop=10")
boxplot(V3~V1,data=PlotData,xlab="Number of Calibration", ylab="RMSEP",main="PLS 4~m5data Loop=10")
```


PLS 4~m5data Loop=10
RMSECV
0.2
0.3
0.4
0.5
0.6
0.7
20
29
38
47
56
65
Number of Calibration


PLS 4~m5data Loop=10
RMSEP
0.2
0.3
0.4
0.5
0.6
0.7
20
29
38
47
56
65
Number of Calibration