HW5 (Due Fr. Dec. 5 by 5pm in D2L)

Statistical Computing (STT 802, EPI 853b)

11/26/2024

In this Homework, we will develop models to predict a phenotype (wheat grain yield) using DNA markers. We will use a data set (included in the BGLR R-package) that has data for 599 inbred lines of wheat. For each inbred line the data set has four phenotypes (we will use just one of them) and 1279 DNA markers.

Packages and data sets

To complete this HW you will need to install the `BGData` and `BGLR` R-packages.

```
install.packages(pkg=c('BGLR','BGData'),repos='https://cran.r-project.org/',type='binary')
```

Use the following code to load the data set into the environment. The `X` matrix has the DNA markers and the `y` vector will have the phenotypes.

```
suppressMessages(library(BGLR))
data(wheat)
X=wheat.X
y=wheat.Y[,2]
dim(X)
```

```
## [1]  599 1279
```

```
str(y)
```

```
##  Named num [1:599] -1.7275 0.4095 -0.6486 0.0939 -0.2825 ...
##  - attr(*, "names")= chr [1:599] "775" "2166" "2167" "2465" ...
```

In the HW, we will build prediction models using forward (FWD) regression and Lasso. We will compare these two methods based on their prediction accuracy in testing data.

For Questions 1 and 2 use this training-testing partition.

```
N<-nrow(X) ; p<-ncol(X)
set.seed(12345)
tst<-sample(1:N,size=150,replace=FALSE)
XTRN<-scale(X[-tst,],center=TRUE,scale=FALSE)
yTRN<-scale(y[-tst],center=TRUE,scale=FALSE)
XTST<-scale(X[tst,],center=TRUE,scale=FALSE)
yTST<-scale(y[tst],center=TRUE,scale=FALSE)
```

Q1) Evaluating prediction performance of a Forward regression along the forward path

The following script shows how to fit a FWD regression over 30 steps (i.e., including up to 30 predictors).

```r
suppressMessages(library(BGData))

FM=FWD(y=yTRN,X=XTRN,df=30,verbose=FALSE)
```

The object `FM$B` has the effects, rows are used for predictiors and each column gives the effects on the i^{th} step. Note that the first row corresponds to the intercept, rows 1279-1280 contain the effects of the DNA markers.

```r
dim(FM$B)
```

```
## [1] 1280   31
```

```r
head(rownames(FM$B))
```

```
## [1] "Int"      "wPt.0538" "wPt.8463" "wPt.6348" "wPt.9992" "wPt.2838"
```

At the j^{th} step there are only j predictors in the model. Thus, in the first column, only the intercept is included

```r
sum(FM$B[,1]!=0)
```

```
## [1] 1
```

```r
which(FM$B[,1]!=0)
```

```
## Int
##   1
```

At the j^{th} step there are j active predictors (counting the intercept).

```r
j=4
sum(FM$B[,j]!=0)
```

```
## [1] 4
```

```r
which(FM$B[,j]!=0)
```

```
##      Int wPt.2151 wPt.7160 c.305387
##        1      314      518      756
```

To derive predictions in the training data set using the model of the j^{th} step you can use

```r
j=4
yHatTRN=cbind(1,XTRN)%*%FM$B[,j]
```

Tasks:

- Run the Forward regression for up to 100 steps (note, this may take a few minutes),
- For each step, evaluate the squared-correlation between predictions and observations in the training and testing data set.
- Report a plot with step number in the x-axis and the pprediction squared correlation in the trainig data in the y-axis.
- Report a similar plot for the prediction squared-correlation in testing data.
- What was the maximum sqaured correlation in testing data achieved by the Forward regression?
- How many steps do you recommend to use?

Q2) Evaluating prediction performance of a Lasso regression along the regularization path.

The following code shows how to fit a Lasso regression, by default, `glmnet()` fits the model for a grid of 100 values of the regularization parameter λ.

```
library(glmnet)
```

```
## Loading required package: Matrix
## Loaded glmnet 4.1-6
```

```
fmL=glmnet(y=yTRN,x=XTRN,alpha=1)
```

The fitted object has the values of the regularization parameter used `fmL$lambda` and the estiamted effects.

The object `fmL$beta` has the estiamted effects for each of the value if λ and `fmL$a0` has the estimated intercept for each value of lambda. `fmL$df` tells you how many predictors were active for each value of λ

```
str(fmL$lambda)
```

```
##  num [1:100] 0.248 0.237 0.226 0.215 0.206 ...
```

```
str(fmL$a0)
```

```
##  Named num [1:100] 5.98e-17 6.10e-17 6.23e-17 6.30e-17 6.09e-17 ...
##  - attr(*, "names")= chr [1:100] "s0" "s1" "s2" "s3" ...
```

```
dim(fmL$beta)
```

```
## [1] 1279  100
```

The function `predict()` can be called on the fitted object.

Tasks:

- Fit the Lasso regression using the training data
- Use the model to derive predictions for the training and testing data set.
- Report a plot with step number in the x-axis and the pprediction squared correlation in the trainig data in the y-axis.
- Report a similar plot for the prediction squared-correlation in testing data.
- What was the maximum sqaured correlation in testing data achieved by Lasso?
- What value of λ do you recommend to use?
- How many active predictors did Lasso have for that value of λ?

Q3: Conclussions and recommendations

Summarize, in no more than 200 words your findings and recommendations.