NBA 4920/6921 Lecture 11

Linear Model Stepwise Selection Application

Murat Unal

10/05/2021

```r
rm(list=ls())
options(digits = 3, scipen = 999)
library(tidyverse)
library(ISLR)
library(cowplot)
library(ggcorrplot)
library(stargazer)
library(corrr)
library(lmtest)
library(sandwich)
library(MASS)
library(car)
library(jtools)
library(caret)
library(leaps)
library(future.apply)
hitters <- ISLR::Hitters
hitters <- na.omit(hitters)
set.seed(2)
```

```r
dim(hitters)
```

```
[1] 263  20
```

```r
names(hitters)
```

```
 [1] "AtBat"     "Hits"      "HmRun"     "Runs"      "RBI"
 [7] "Years"     "CAtBat"    "CHits"     "CHmRun"    "CRuns
[13] "CWalks"    "League"    "Division"  "PutOuts"   "Assis
[19] "Salary"    "NewLeague"
```

Best subset selection

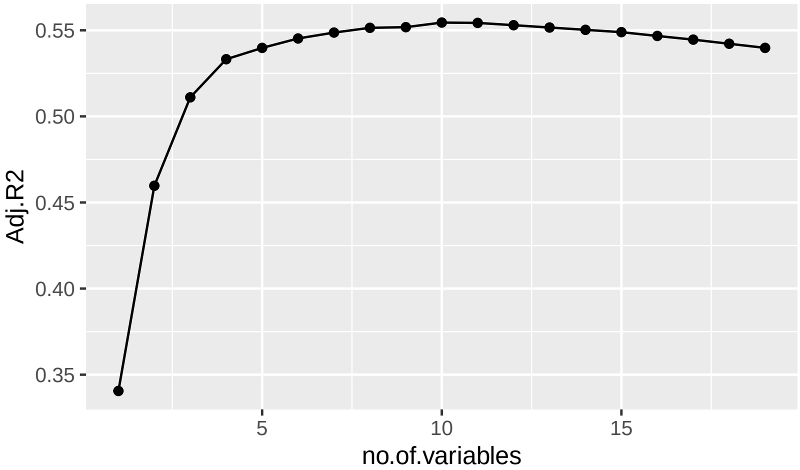
```r
# Draw validation set
hit_validation_data = hitters %>% sample_frac(size = 0.3)
# Create the remaining training set
hit_training_data = setdiff(hitters, hit_validation_data)
```

```r
nvars = 19
regfit.best=regsubsets(Salary~.,data=hit_training_data,
                                            nvmax=nvars)
best.sum <- summary(regfit.best)
best.model <- which.max(best.sum$adjr2)
best.model
```

```
[1] 10
```

```r
coef(regfit.best,id=best.model)
```

```
(Intercept)       AtBat        Hits       Walks      CAtBat
     88.308      -1.687       6.052       5.587      -0.130
      CRuns        CRBI      CWalks   DivisionW     PutOuts
      1.448       1.204      -0.912     -87.035       0.233
```



0.55
0.50
0.45
0.40
0.35
Adj.R2
5
10
15
no.of.variables

Validation set approach

```
validation.mat=model.matrix(Salary~.,
                            data=hit_validation_data)

val.errors = numeric(nvars)
for(each in 1:nvars){
    coefi = coef(regfit.best,id=each)
    pred = validation.mat[,names(coefi)]%*%coefi
    val.errors[each]=
      mean((hit_validation_data$Salary-pred)^2)
}

which.min(val.errors)
```

```
[1] 8
```

K-fold cross validation

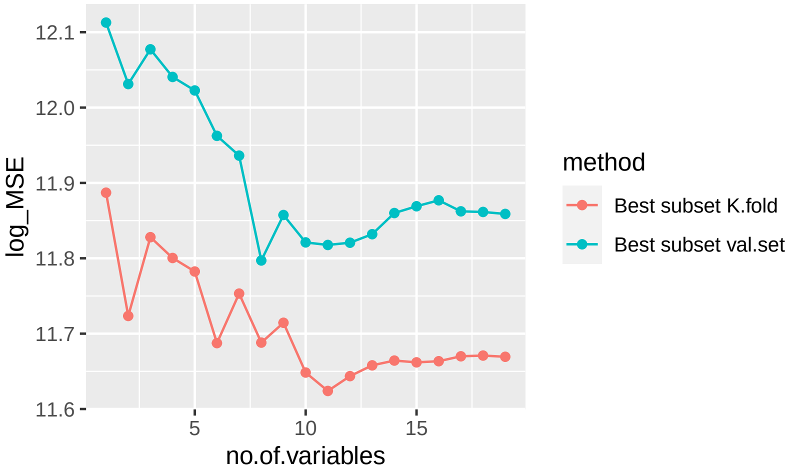
```r
nvars = 19
nfold = 10
# Create folds
fold.list <- createFolds(rownames(hitters),nfold)
# Empty vector to store the resulting MSEs
cv.errors =matrix(0,nfold,nvars,
                dimnames =list(NULL,paste (1:nvars)))

for(each in 1:nfold){
 train <- hitters[-fold.list[[each]],]
 validate <- hitters[fold.list[[each]],]

 best.fit=regsubsets(Salary~.,data=train,nvmax =19)
 validation.mat=model.matrix(Salary~.,data=validate)

}
```

..continued from before

```
for(i in 1:nvars){
   coefi = coef(regfit.best,id=i)
   pred = validation.mat[,names(coefi)]%*%coefi
   cv.errors[each,i] = mean( (validate$Salary-pred)^2)
   }
}
```

```
mean.cv.errors=apply(cv.errors ,2, mean)
best.subset.model <- which.min(mean.cv.errors)
best.subset.model
```

```
11
11
```


12.1
12.0
11.9
11.8
11.7
11.6
log_MSE
5
10
15
no.of.variables
method
Best subset K.fold
Best subset val.set

To obtain the final model we perform best subset selection on the full data set and obtain the 11'-variable model.

```
best.fit=regsubsets(Salary~.,data=hitters,nvmax =19)
coef(best.fit,best.subset.model)
```

```
(Intercept)        AtBat         Hits        Walks        CAtBat
    135.751       -2.128        6.924        5.620        -0.139
       CRBI       CWalks      LeagueN    DivisionW       PutOuts
      0.785       -0.823       43.112     -111.146         0.289
```

This is your final model that you'd deploy to predict the salary of baseball players.

Forward Stepwise Selection

We can also use the `regsubsets()` function to perform forward stepwise or backward stepwise selection, using the argument `method="forward"` or `method="backward"`

```
regfit.fwd=regsubsets(Salary~.,data=hitters,
                      nvmax=19,method="forward")
fwd.sum <- summary(regfit.fwd)
fwd.model <- which.max(fwd.sum$adjr2)
fwd.model
```

```
[1] 11
```

```r
coef(regfit.fwd, id=fwd.model)[1:4]
```

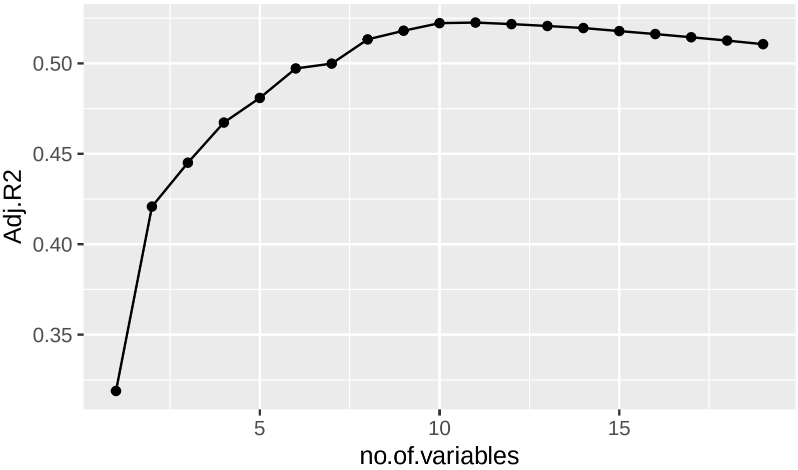
```
(Intercept)        AtBat         Hits        Walks
     135.75        -2.13         6.92         5.62
```

```r
coef(regfit.fwd, id=fwd.model)[5:9]
```

```
 CAtBat    CRuns     CRBI   CWalks LeagueN
 -0.139    1.455    0.785   -0.823  43.112
```

```r
coef(regfit.fwd, id=fwd.model)[10:12]
```

```
DivisionW    PutOuts    Assists
 -111.146      0.289      0.269
```


0.50
0.45
0.40
0.35
Adj.R2
5
10
15
no.of.variables

Validation set approach

```
nvars=19
regfit.fwd=regsubsets(Salary~.,data=hit_training_data,
                      nvmax=nvars,method="forward")

summary(regfit.fwd)
```

```
Subset selection object
Call: regsubsets.formula(Salary ~ ., data = hit_training_da
    method = "forward")
19 Variables  (and intercept)
           Forced in Forced out
AtBat          FALSE      FALSE
Hits           FALSE      FALSE
HmRun          FALSE      FALSE
Runs           FALSE      FALSE
RBI            FALSE      FALSE
Walks          FALSE      FALSE
Years          FALSE      FALSE
```

K-fold cross validation

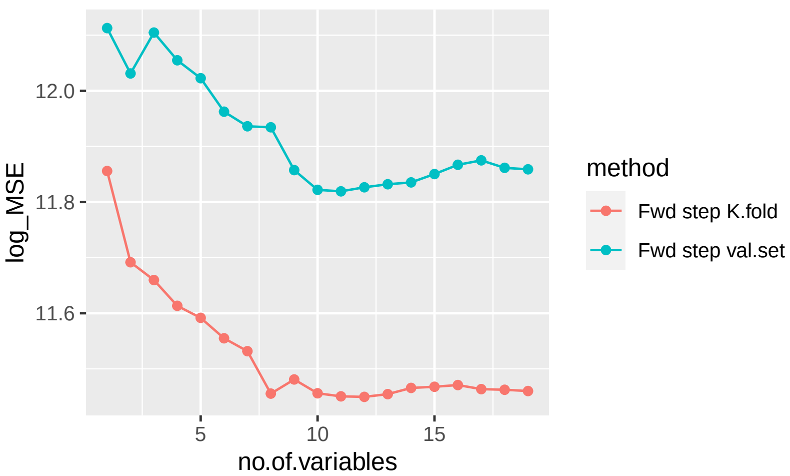
```r
nvars = 19
nfold = 10
# Create folds
fold.list <- createFolds(rownames(hitters),nfold)
# Empty vector to store the resulting MSEs
cv.errors =matrix(0,nfold,nvars,
                  dimnames =list(NULL,paste (1:nvars)))

for(each in 1:nfold){
 train <- hitters[-fold.list[[each]],]
 validate <- hitters[fold.list[[each]],]

 best.fit=regsubsets(Salary~.,data=train,nvmax =19,
                        method = "forward")
 validation.mat=model.matrix(Salary~.,data=validate)

}
```

..continued from before

```
for(i in 1:nvars){
   coefi = coef(best.fit,id=i)
   pred = validation.mat[,names(coefi)]%*%coefi
   cv.errors[each,i] = mean( (validate$Salary-pred)^2)
   }
}
```

```r
mean.fwd.cv.errors=apply(fwd.cv.errors ,2, mean)
best.fwd.cv.model <- which.min(mean.fwd.cv.errors)
best.fwd.cv.model
```

```
12
12
```



log_MSE
12.0
11.8
11.6
5
10
15
no.of.variables
method
Fwd step K.fold
Fwd step val.set

To obtain the final model we perform forward stepwise selection on the full data set and obtain the 12-variable model.

```
best.fwd.fit=regsubsets(Salary~.,data=hitters,nvmax =19,
                        method = "forward")
coef(best.fwd.fit,best.fwd.cv.model)
```

```
(Intercept)       AtBat        Hits        Runs       Walks
    135.519      -2.056       7.506      -1.797       6.062
      CRuns        CRBI      CWalks     LeagueN   DivisionW
      1.559       0.778      -0.835      39.087    -112.644
    Assists
      0.243
```

This is your final model that you'd deploy to predict the salary of baseball players.

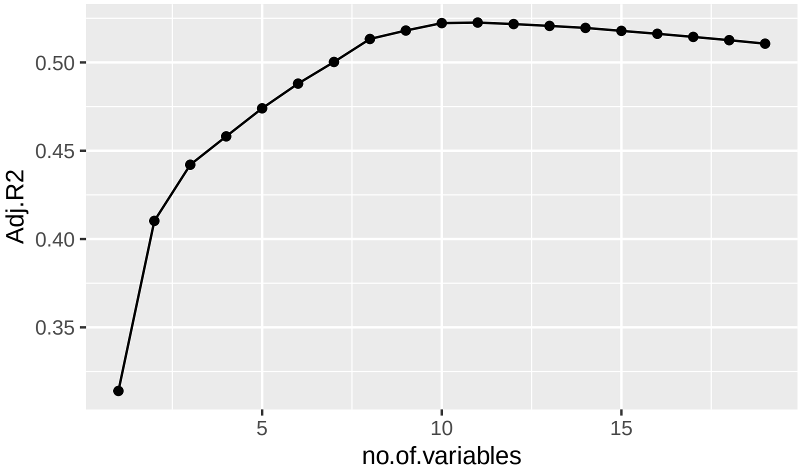
Backward Stepwise Selection

```r
regfit.bwd=regsubsets(Salary~.,data=hitters,
                      nvmax=19,method="backward")
bwd.sum <- summary(regfit.bwd)
bwd.model <- which.max(bwd.sum$adjr2)
bwd.model
```

```
[1] 11
```

```
coef(regfit.bwd, id=bwd.model)
```

```
(Intercept)       AtBat        Hits       Walks      CAtBat
    135.751      -2.128       6.924       5.620      -0.139
       CRBI      CWalks     LeagueN   DivisionW     PutOuts
      0.785      -0.823      43.112    -111.146       0.289
```


Adj.R2
0.50
0.45
0.40
0.35
5
10
15
no.of.variables

Validation set approach

```
nvars=19
regfit.bwd=regsubsets(Salary~.,data=hit_training_data,
                      nvmax=nvars,method="backward")

validation.mat=model.matrix(Salary~.,
                      data=hit_validation_data)

bwd.val.errors = numeric(nvars)
for(each in 1:nvars){
    coefi = coef(regfit.bwd,id=each)
    pred = validation.mat[,names(coefi)]%*%coefi
    bwd.val.errors[each]=
      mean((hit_validation_data$Salary-pred)^2)
}

which.min(bwd.val.errors)
```

```
[1] 13
```

K-fold cross validation

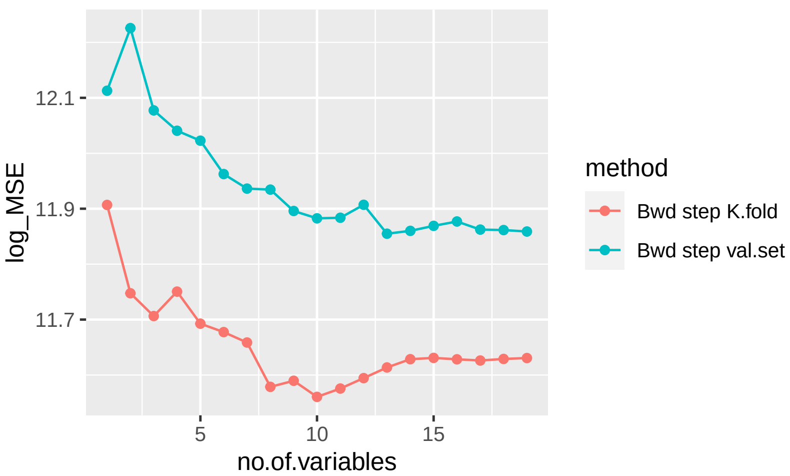
```
nvars = 19
nfold = 10
# Create folds
fold.list <- createFolds(rownames(hitters),nfold)
# Empty vector to store the resulting MSEs
cv.errors =matrix(0,nfold,nvars,
                dimnames =list(NULL,paste (1:nvars)))

for(each in 1:nfold){
 train <- hitters[-fold.list[[each]],]
 validate <- hitters[fold.list[[each]],]

 best.fit=regsubsets(Salary~.,data=train,nvmax =19,
                     method = "backward")
 validation.mat=model.matrix(Salary~.,data=validate)

}
```

..continued from before

```
 for(i in 1:nvars){
   coefi = coef(regfit.best,id=i)
   pred = validation.mat[,names(coefi)]%*%coefi
   cv.errors[each,i] = mean( (validate$Salary-pred)^2)
   }
}
```

```r
mean.bwd.cv.errors=apply(bwd.cv.errors ,2, mean)
best.bwd.cv.model <- which.min(mean.bwd.cv.errors)
best.bwd.cv.model
```

```
10
10
```



log_MSE
12.1
11.9
11.7
5
10
15
no.of.variables
method
Bwd step K.fold
Bwd step val.set

To obtain the final model we perform backward stepwise selection on the full data set and obtain the 10‘-variable model.

```
best.bwd.fit=regsubsets(Salary~.,data=hitters,nvmax =19,
                         method = "backward")
coef(best.bwd.fit,best.bwd.cv.model)
```

```
(Intercept)       AtBat        Hits       Walks      CAtBat
    162.535      -2.169       6.918       5.773      -0.130
       CRBI      CWalks   DivisionW     PutOuts     Assists
      0.774      -0.831    -112.380       0.297       0.283
```

This is your final model that you’d deploy to predict the salary of baseball players.

Let's compare the test error estimates from all approaches