forward_backward_sel_linear_model.R

vikaskamath

2021-03-20

```
### In this part we will do:
### (1) All subset selections with no interaction
### (2) Forward selection with two way interactions
### (3) Backward selection with two way interactions

### Upload the data:

wine = read.csv("white-wine.csv", header=TRUE)

# Install packages for sub set selection
# install.packages("leaps")
library(leaps)

### The regsubsets function is the main function in this package that
###  allows you to do variable selection
### Setting nvmax = 11, will consider all the possible subsets
### When the summary() on the results is run, it computes the best as the lowest RSS

all.subsets.models = regsubsets(quality ~ .,     data = wine, nvmax = 11)
forward.models     = regsubsets(quality ~ (.)^2, data = wine, nvmax = 66, method = "forward")
backward.models    = regsubsets(quality ~ (.)^2, data = wine, nvmax = 66, method = "backward")

summary.all.subsets.models = summary(all.subsets.models)
summary.all.subsets.models
```

```
## Subset selection object
## Call: regsubsets.formula(quality ~ ., data = wine, nvmax = 11)
## 11 Variables  (and intercept)
##                       Forced in Forced out
## fixed.acidity             FALSE      FALSE
## volatile.acidity          FALSE      FALSE
## citric.acid               FALSE      FALSE
## residual.sugar            FALSE      FALSE
## chlorides                 FALSE      FALSE
## free.sulfur.dioxide       FALSE      FALSE
## total.sulfur.dioxide      FALSE      FALSE
## density                   FALSE      FALSE
## pH                        FALSE      FALSE
## sulphates                 FALSE      FALSE
## alcohol                   FALSE      FALSE
## 1 subsets of each size up to 11
## Selection Algorithm: exhaustive
```

```
##           fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1  ( 1 )  " "           " "              " "         " "             " "
## 2  ( 1 )  " "           "*"              " "         " "             " "
## 3  ( 1 )  " "           "*"              " "         "*"             " "
## 4  ( 1 )  " "           "*"              " "         "*"             " "
## 5  ( 1 )  " "           "*"              " "         "*"             " "
## 6  ( 1 )  " "           "*"              " "         "*"             " "
## 7  ( 1 )  " "           "*"              " "         "*"             " "
## 8  ( 1 )  "*"           "*"              " "         "*"             " "
## 9  ( 1 )  "*"           "*"              " "         "*"             " "
## 10  ( 1 ) "*"           "*"              " "         "*"             "*"
## 11  ( 1 ) "*"           "*"              "*"         "*"             "*"
##           free.sulfur.dioxide total.sulfur.dioxide density pH  sulphates
## 1  ( 1 )  " "                 " "                 " "     " " " " "
## 2  ( 1 )  " "                 " "                 " "     " " " " "
## 3  ( 1 )  " "                 " "                 " "     " " " " "
## 4  ( 1 )  "*"                 " "                 " "     " " " " "
## 5  ( 1 )  " "                 " "                 "*"     "*" " " "
## 6  ( 1 )  " "                 " "                 "*"     "*" "*"
## 7  ( 1 )  "*"                 " "                 "*"     "*" "*"
## 8  ( 1 )  "*"                 " "                 "*"     "*" "*"
## 9  ( 1 )  "*"                 "*"                 "*"     "*" "*"
## 10  ( 1 ) "*"                 "*"                 "*"     "*" "*"
## 11  ( 1 ) "*"                 "*"                 "*"     "*" "*"
##           alcohol
## 1  ( 1 )  "*"
## 2  ( 1 )  "*"
## 3  ( 1 )  "*"
## 4  ( 1 )  "*"
## 5  ( 1 )  "*"
## 6  ( 1 )  "*"
## 7  ( 1 )  "*"
## 8  ( 1 )  "*"
## 9  ( 1 )  "*"
## 10  ( 1 ) "*"
## 11  ( 1 ) "*"
```

```
###
### Among 1 variable models, the model with only alcohol had the lowest RSS.
### Among 2 variable models, the model with alcohol and volatile.acidity has the lowest RSS.
### Among 3 variable models, the model with alcohol and volatile.acidity and residual.sugar has the low
###

### What additional information is contained in the summary?

names(summary.all.subsets.models)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

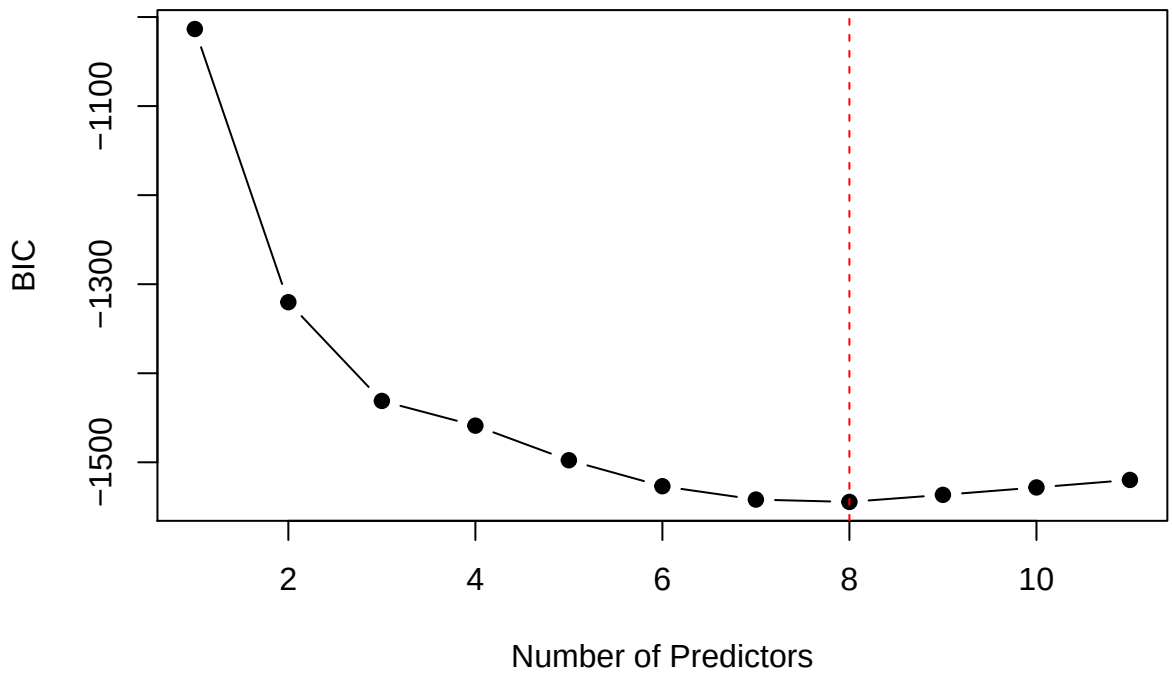
```
###
### Extract the BIC values.  The lower the BIC, the better the model.
###
```

```
bic.all.subsets.models = summary.all.subsets.models$bic
bic.all.subsets.models
```

```
##  [1] -1013.458 -1320.192 -1431.081 -1458.839 -1497.680 -1526.853 -1541.854
##  [8] -1544.476 -1536.548 -1528.237 -1519.794
```

```
###
### Plot the BIC values versus the number of predictors to see the "best" model:
###

predictors = 1:11
plot(bic.all.subsets.models ~ predictors, ylab="BIC", xlab="Number of Predictors", type="b", pch=19)
abline(v = 8, lty = 2, col = "red")
```




```
### Note something interesting: BIC keeps on decreasing but after some point it starts going up; models
### more predictors start to perform worse.

### 8 appears to be the lowest one.  Which ones are they?

summary.all.subsets.models
```

```
## Subset selection object
## Call: regsubsets.formula(quality ~ ., data = wine, nvmax = 11)
## 11 Variables  (and intercept)
```

```
##                       Forced in Forced out
## fixed.acidity             FALSE      FALSE
## volatile.acidity          FALSE      FALSE
## citric.acid               FALSE      FALSE
## residual.sugar            FALSE      FALSE
## chlorides                 FALSE      FALSE
## free.sulfur.dioxide       FALSE      FALSE
## total.sulfur.dioxide      FALSE      FALSE
## density                   FALSE      FALSE
## pH                        FALSE      FALSE
## sulphates                 FALSE      FALSE
## alcohol                   FALSE      FALSE
## 1 subsets of each size up to 11
## Selection Algorithm: exhaustive
##           fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1  ( 1 )  " "           " "              " "          " "             " "
## 2  ( 1 )  " "           "*"              " "          " "             " "
## 3  ( 1 )  " "           "*"              " "          "*"             " "
## 4  ( 1 )  " "           "*"              " "          "*"             " "
## 5  ( 1 )  " "           "*"              " "          "*"             " "
## 6  ( 1 )  " "           "*"              " "          "*"             " "
## 7  ( 1 )  " "           "*"              " "          "*"             " "
## 8  ( 1 )  "*"           "*"              " "          "*"             " "
## 9  ( 1 )  "*"           "*"              " "          "*"             " "
## 10  ( 1 ) "*"           "*"              " "          "*"             "*"
## 11  ( 1 ) "*"           "*"              "*"          "*"             "*"
##           free.sulfur.dioxide total.sulfur.dioxide density pH  sulphates
## 1  ( 1 )  " "                 " "                  " "     " " " "
## 2  ( 1 )  " "                 " "                  " "     " " " "
## 3  ( 1 )  " "                 " "                  " "     " " " "
## 4  ( 1 )  "*"                 " "                  " "     " " " "
## 5  ( 1 )  " "                 " "                  "*"     "*" " "
## 6  ( 1 )  " "                 " "                  "*"     "*" "*"
## 7  ( 1 )  "*"                 " "                  "*"     "*" "*"
## 8  ( 1 )  "*"                 " "                  "*"     "*" "*"
## 9  ( 1 )  "*"                 "*"                  "*"     "*" "*"
## 10  ( 1 ) "*"                 "*"                  "*"     "*" "*"
## 11  ( 1 ) "*"                 "*"                  "*"     "*" "*"
##           alcohol
## 1  ( 1 )  "*"
## 2  ( 1 )  "*"
## 3  ( 1 )  "*"
## 4  ( 1 )  "*"
## 5  ( 1 )  "*"
## 6  ( 1 )  "*"
## 7  ( 1 )  "*"
## 8  ( 1 )  "*"
## 9  ( 1 )  "*"
## 10  ( 1 ) "*"
## 11  ( 1 ) "*"
```

```
coef(all.subsets.models,8)
```

```
##         (Intercept)       fixed.acidity    volatile.acidity      residual.sugar
```

```
##        1.541062e+02        6.810394e-02       -1.888140e+00        8.284724e-02
## free.sulfur.dioxide             density                  pH           sulphates
##        3.349015e-03       -1.542913e+02        6.942135e-01        6.285081e-01
##             alcohol
##        1.931628e-01
```

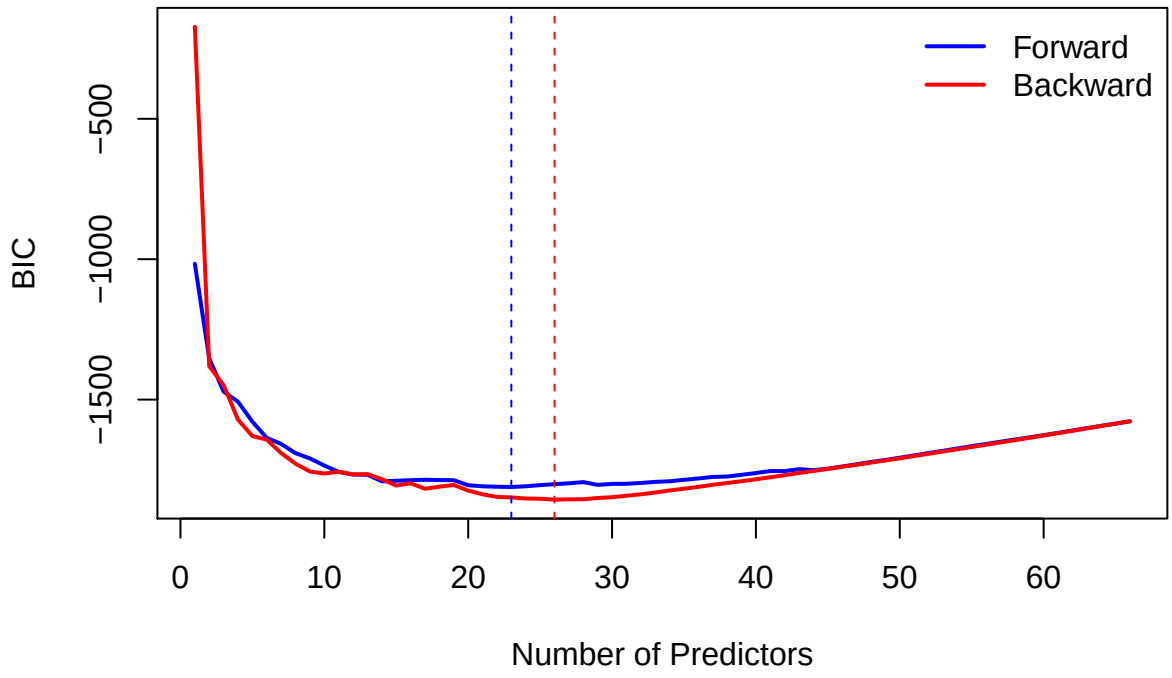
```
### They are: fixed.acidity + volatile.acidity + residual.sugar + free.sulfur.dioxide + density + pH +

### Let's look at the results of the forward and backward too

summary.forward.models  = summary(forward.models)
summary.backward.models = summary(backward.models)

bic.values.forward  = summary.forward.models$bic
bic.values.backward = summary.backward.models$bic

predictors = 1:66
matplot(x = predictors , y = cbind(bic.values.forward, bic.values.backward),type = "l", lty = 1,
        ylab = "BIC", xlab="Number of Predictors", col = c("blue","red"), lwd = 2 )
abline(v = which.min(bic.values.forward),   lty = 2, col = "blue")
abline(v = which.min(bic.values.backward),  lty = 2, col = "red")
legend("topright", c("Forward","Backward"), lty = 1, col = c("blue","red"), lwd = 2, bty = "n")
```




```
### The BIC based best models must be different.
```

```
### So what?  What have we done? Which model should we go with?

which.min(bic.values.forward)
```

```
## [1] 23
```

```
which.min(bic.values.backward)
```

```
## [1] 26
```

```
coef(forward.models,  23)
```

```
##                                (Intercept)
##                               1.603629e+01
##                           volatile.acidity
##                              -1.537212e+02
##                         free.sulfur.dioxide
##                              -2.435197e+00
##                        total.sulfur.dioxide
##                               1.214100e-02
##                                    alcohol
##                               2.541716e+01
##             fixed.acidity:volatile.acidity
##                              -9.474352e-02
##               fixed.acidity:residual.sugar
##                               3.876906e-03
##                    fixed.acidity:chlorides
##                               1.357402e+00
##          fixed.acidity:free.sulfur.dioxide
##                               1.777602e-03
##                  volatile.acidity:chlorides
##                              -5.365445e+00
##                    volatile.acidity:density
##                               1.458530e+02
##                    volatile.acidity:alcohol
##                               7.414368e-01
##         residual.sugar:free.sulfur.dioxide
##                              -1.383180e-03
##                      residual.sugar:alcohol
##                               9.493015e-03
##                          chlorides:alcohol
##                              -8.295223e-01
## free.sulfur.dioxide:total.sulfur.dioxide
##                              -9.688053e-05
##                free.sulfur.dioxide:density
##                               2.401000e+00
##              free.sulfur.dioxide:sulphates
##                               3.095014e-02
##                free.sulfur.dioxide:alcohol
##                               4.756285e-03
##             total.sulfur.dioxide:sulphates
##                              -1.832049e-02
```

```
##                             density:pH
##                          -2.882743e+00
##                        density:alcohol
##                          -2.686009e+01
##                           pH:sulphates
##                           6.153656e-01
##                             pH:alcohol
##                           3.135028e-01
```

```
coef(backward.models, 26)
```

```
##                             (Intercept)
##                           4.456253e+00
##                      volatile.acidity
##                          -4.642154e+02
##                        residual.sugar
##                           2.530556e-01
##                               chlorides
##                          -1.628748e+03
##                     free.sulfur.dioxide
##                          -5.051230e+00
##                                      pH
##                           1.347262e+02
##                                alcohol
##                           1.202098e+01
##          fixed.acidity:volatile.acidity
##                          -5.141489e-01
##               fixed.acidity:citric.acid
##                          -5.028716e-01
##                        fixed.acidity:pH
##                           1.448748e-01
##         volatile.acidity:residual.sugar
##                          -1.450347e-01
##                volatile.acidity:density
##                           4.581726e+02
##                volatile.acidity:alcohol
##                           1.090409e+00
##                     citric.acid:density
##                           3.457918e+00
##                residual.sugar:chlorides
##                          -1.144411e+00
##       residual.sugar:free.sulfur.dioxide
##                          -2.228968e-03
##                      chlorides:density
##                           1.685777e+03
##                           chlorides:pH
##                          -1.352663e+01
## free.sulfur.dioxide:total.sulfur.dioxide
##                          -1.021969e-04
##             free.sulfur.dioxide:density
##                           5.020584e+00
##           free.sulfur.dioxide:sulphates
##                           2.597139e-02
##             free.sulfur.dioxide:alcohol
```

```
##                              7.883666e-03
##             total.sulfur.dioxide:density
##                              1.224371e-02
##           total.sulfur.dioxide:sulphates
##                             -1.775517e-02
##                                density:pH
##                             -1.353385e+02
##                           density:alcohol
##                             -1.257575e+01
##                              pH:sulphates
##                              6.545284e-01
```

```
### Some ugly string processing

u1               = toString(names(coef(forward.models, 23))[-1])
u2               = gsub(pattern = ", ",  replacement = " + ", x = toString(u1))
forward.formula  = as.formula(paste("quality ~ ", u2, sep = ""))

v1               = toString(names(coef(backward.models, 26))[-1])
v2               = gsub(pattern = ", ",  replacement = " + ", x = toString(v1))
backward.formula = as.formula(paste("quality ~ ", v2, sep = ""))

### Does it perform better when doing prediction?
### Just for fun let's inculde a model with only alcohol

no.of.folds  = 10
set.seed(666)
index.values = sample(1:no.of.folds, size = dim(wine)[1], replace = TRUE)

mse = matrix(NA, nrow = no.of.folds, ncol = 4)
colnames(mse) = c("Alcohol","All","Forward","Backward")

for (i in 1:no.of.folds)
{
   index.out               = which(index.values == i)
   left.out.data           = wine[  index.out, ]
   left.in.data            = wine[ -index.out, ]

   alcohol          = lm(quality ~ alcohol, data = left.in.data)
   all              = lm(quality ~ fixed.acidity + volatile.acidity + residual.sugar + free.sulfur.dioxi
                                   density + pH + sulphates + alcohol, data = left.in.data)
   forward          = lm(forward.formula,  data = left.in.data)
   backward         = lm(backward.formula, data = left.in.data)

   y =  left.out.data[,12]

   mse[i,1] = mean((y - predict(alcohol,  newdata = left.out.data))^2)
   mse[i,2] = mean((y - predict(all,      newdata = left.out.data))^2)
   mse[i,3] = mean((y - predict(forward,  newdata = left.out.data))^2)
   mse[i,4] = mean((y - predict(backward, newdata = left.out.data))^2)

}

### Now let's look at the RMSE values.
```

```r
head(mse)
```

```
##        Alcohol       All   Forward  Backward
## [1,] 0.6848250 0.6011510 0.5406846 0.5498007
## [2,] 0.6380729 0.5546313 0.5168927 0.5166337
## [3,] 0.6374817 0.5585944 0.5339785 0.5264624
## [4,] 0.6249472 0.5920126 0.5927749 0.5287814
## [5,] 0.6385844 0.5705884 0.5243246 0.5200230
## [6,] 0.6281525 0.5933352 0.5687950 0.5506213
```

```r
sqrt(apply(mse, MARGIN = 2, FUN = mean))
```

```
##   Alcohol       All   Forward  Backward
## 0.7973652 0.7534422 0.7355584 0.7270722
```

```r
################################################################################
```