Model Selection Methods

Aymen Rumi

5/20/2020

Overview

We will run regression analysis on the College. The goal in this report is to try out different model selection techniques for regression analysis such as variable selection, regularization & dimensionality reduction techniques, in aims to pick the best predictive model for our dataset(lowest error rate on testing data)

College Dataset

We want to predict the number of applicants using all the variables available

Standard Regression Analysis

```r
summary(lm(data=College,Apps~.))
```

```
Call:
lm(formula = Apps ~ ., data = College)

Residuals:
    Min      1Q  Median      3Q     Max
-4908.8  -430.2   -29.5   322.3  7852.5

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -445.08413  408.32855  -1.090 0.276053
PrivateYes  -494.14897  137.81191  -3.586 0.000358 ***
Accept         1.58581    0.04074  38.924  < 2e-16 ***
Enroll        -0.88069    0.18596  -4.736 2.60e-06 ***
Top10perc     49.92628    5.57824   8.950  < 2e-16 ***
Top25perc    -14.23448    4.47914  -3.178 0.001543 **
F.Undergrad    0.05739    0.03271   1.754 0.079785 .
P.Undergrad    0.04445    0.03214   1.383 0.167114
Outstate      -0.08587    0.01906  -4.506 7.64e-06 ***
Room.Board     0.15103    0.04829   3.127 0.001832 **
Books          0.02090    0.23841   0.088 0.930175
Personal       0.03110    0.06308   0.493 0.622060
PhD           -8.67850    4.63814  -1.871 0.061714 .
Terminal      -3.33066    5.09494  -0.654 0.513492
S.F.Ratio     15.38961   13.00622   1.183 0.237081
perc.alumni    0.17867    4.10230   0.044 0.965273
Expend         0.07790    0.01235   6.308 4.79e-10 ***
Grad.Rate      8.66763    2.94893   2.939 0.003390 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 1041 on 759 degrees of freedom
Multiple R-squared:  0.9292,    Adjusted R-squared:  0.9276
F-statistic: 585.9 on 17 and 759 DF,  p-value: < 2.2e-16
```

Observations

From the regression function summary, 10 variables seem to have a low p-value indicating significant contribution to predicting the response, these variables are: PrivateYes, Accept,Enroll, Top10perc, Top25perc,Outstate,PhD,Room.Board, Expend,Grad.Rate

Selecting Best Subset of Predictors

We will perform forward stepwise variable selection, & plot graphs for statistics such as Cp,BIC,RSS, & Rsquared as the # of variables increases

```r
College<-College

regfit.full=regsubsets(Apps~.,data=College,nvmax=17,method="forward")
reg.summary=summary(regfit.full)

reg.summary
```

```
Subset selection object
Call: regsubsets.formula(Apps ~ ., data = College, nvmax = 17, method = "forward")
17 Variables  (and intercept)
            Forced in Forced out
PrivateYes      FALSE      FALSE
Accept          FALSE      FALSE
Enroll          FALSE      FALSE
Top10perc       FALSE      FALSE
Top25perc       FALSE      FALSE
F.Undergrad     FALSE      FALSE
P.Undergrad     FALSE      FALSE
Outstate        FALSE      FALSE
Room.Board      FALSE      FALSE
Books           FALSE      FALSE
Personal        FALSE      FALSE
PhD             FALSE      FALSE
Terminal        FALSE      FALSE
S.F.Ratio       FALSE      FALSE
perc.alumni     FALSE      FALSE
Expend          FALSE      FALSE
Grad.Rate       FALSE      FALSE
1 subsets of each size up to 17
Selection Algorithm: forward
          PrivateYes Accept Enroll Top10perc Top25perc F.Undergrad
1  ( 1 )  " "        "*"    " "    " "       " "       " "
2  ( 1 )  " "        "*"    " "    "*"       " "       " "
3  ( 1 )  " "        "*"    " "    "*"       " "       " "
4  ( 1 )  " "        "*"    " "    "*"       " "       " "
5  ( 1 )  " "        "*"    "*"    "*"       " "       " "
```

```
6  ( 1 ) " "        "*"    "*"    "*"       " "       " "
7  ( 1 ) " "        "*"    "*"    "*"       "*"       " "
8  ( 1 ) "*"        "*"    "*"    "*"       "*"       " "
9  ( 1 ) "*"        "*"    "*"    "*"       "*"       " "
10  ( 1 ) "*"        "*"    "*"    "*"       "*"       " "
11  ( 1 ) "*"        "*"    "*"    "*"       "*"       "*"
12  ( 1 ) "*"        "*"    "*"    "*"       "*"       "*"
13  ( 1 ) "*"        "*"    "*"    "*"       "*"       "*"
14  ( 1 ) "*"        "*"    "*"    "*"       "*"       "*"
15  ( 1 ) "*"        "*"    "*"    "*"       "*"       "*"
16  ( 1 ) "*"        "*"    "*"    "*"       "*"       "*"
17  ( 1 ) "*"        "*"    "*"    "*"       "*"       "*"
          P.Undergrad Outstate Room.Board Books Personal PhD Terminal
1  ( 1 )  " "         " "      " "        " "   " "      " " " "
2  ( 1 )  " "         " "      " "        " "   " "      " " " "
3  ( 1 )  " "         " "      " "        " "   " "      " " " "
4  ( 1 )  " "         "*"      " "        " "   " "      " " " "
5  ( 1 )  " "         "*"      " "        " "   " "      " " " "
6  ( 1 )  " "         "*"      "*"        " "   " "      " " " "
7  ( 1 )  " "         "*"      "*"        " "   " "      " " " "
8  ( 1 )  " "         "*"      "*"        " "   " "      " " " "
9  ( 1 )  " "         "*"      "*"        " "   " "      "*" " "
10  ( 1 ) " "         "*"      "*"        " "   " "      "*" " "
11  ( 1 ) " "         "*"      "*"        " "   " "      "*" " "
12  ( 1 ) "*"         "*"      "*"        " "   " "      "*" " "
13  ( 1 ) "*"         "*"      "*"        " "   " "      "*" " "
14  ( 1 ) "*"         "*"      "*"        " "   " "      "*" "*"
15  ( 1 ) "*"         "*"      "*"        " "   "*"      "*" "*"
16  ( 1 ) "*"         "*"      "*"        "*"   "*"      "*" "*"
17  ( 1 ) "*"         "*"      "*"        "*"   "*"      "*" "*"
          S.F.Ratio perc.alumni Expend Grad.Rate
1  ( 1 )  " "       " "         " "    " "
2  ( 1 )  " "       " "         " "    " "
3  ( 1 )  " "       " "         "*"    " "
4  ( 1 )  " "       " "         "*"    " "
5  ( 1 )  " "       " "         "*"    " "
6  ( 1 )  " "       " "         "*"    " "
7  ( 1 )  " "       " "         "*"    " "
8  ( 1 )  " "       " "         "*"    " "
9  ( 1 )  " "       " "         "*"    " "
10  ( 1 ) " "       " "         "*"    "*"
11  ( 1 ) " "       " "         "*"    "*"
12  ( 1 ) " "       " "         "*"    "*"
13  ( 1 ) "*"       " "         "*"    "*"
14  ( 1 ) "*"       " "         "*"    "*"
15  ( 1 ) "*"       " "         "*"    "*"
16  ( 1 ) "*"       " "         "*"    "*"
17  ( 1 ) "*"       "*"         "*"    "*"
```

```r
par(mfrow=c(2,2))

which.max(reg.summary$adjr2)
```

```
[1] 13
```

```r
plot(reg.summary$adjr2,xlab="Number of Variables",
      ylab="Adjusted RSq",type="l")
points(which.max(reg.summary$adjr2),reg.summary$adjr2[which.max(reg.summary$adjr2)], col="red",cex=2,pc

which.min(reg.summary$rss)
```

```
[1] 17
```

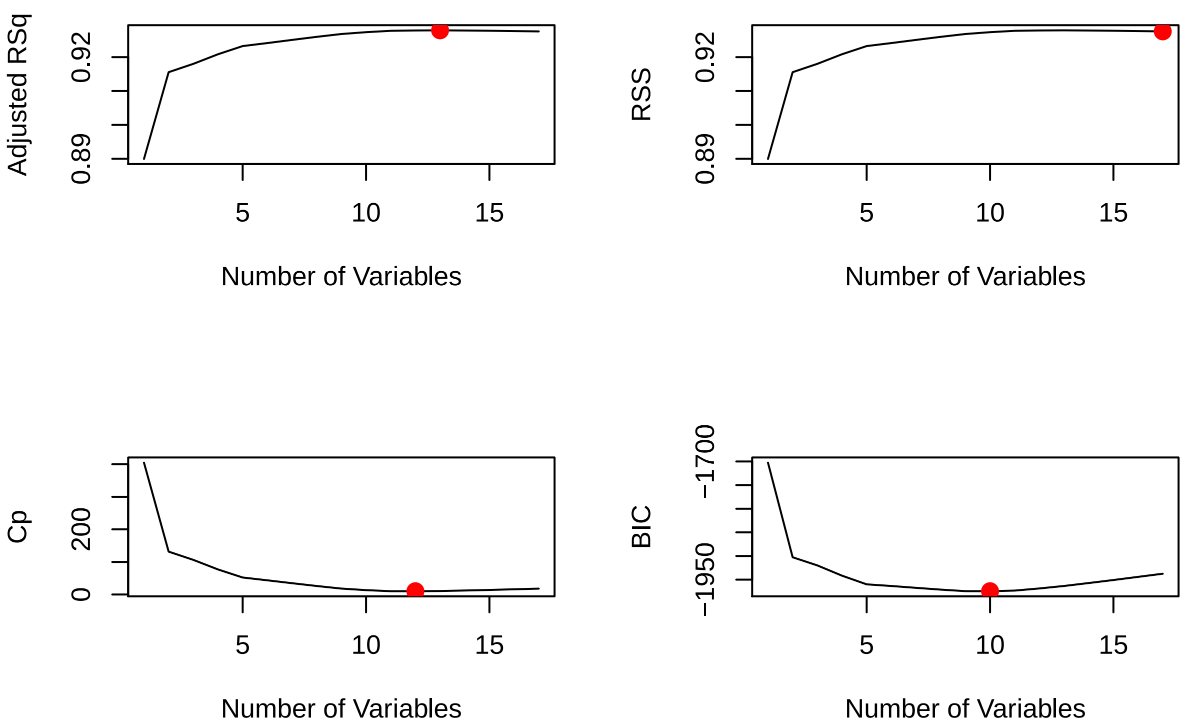
```r
plot(reg.summary$adjr2,xlab="Number of Variables",
      ylab="RSS",type="l")
points(which.min(reg.summary$rss),reg.summary$adjr2[which.min(reg.summary$rss)], col="red",cex=2,pch=20


which.min(reg.summary$cp)
```

```
[1] 12
```

```r
plot(reg.summary$cp,xlab="Number of Variables",
      ylab="Cp",type="l")
points(which.min(reg.summary$cp),reg.summary$cp[which.min(reg.summary$cp)], col="red",cex=2,pch=20)


which.min(reg.summary$bic)
```

```
[1] 10
```

```r
plot(reg.summary$bic,xlab="Number of Variables",
      ylab="BIC",type="l")
points(which.min(reg.summary$bic),reg.summary$bic[which.min(reg.summary$bic)], col="red",cex=2,pch=20)
```



Observations

As we can see there is still no clear evidence of the optimal number of variables to use but BIC had the same result as the regression model with 10 variables: PrivateYes, Accept, Enroll, Top10perc, Top25perc,Outstate, Room.Board, PhD,Expend, Grad.Rate.

The goal of choosing the best subset of variables is to optimize predictive accuracy on future unseen dataset, so let's analyze performance of these subsets by calculating test error rate using K-Fold Cross Validation

Validation Set

```
set.seed(4)

# split the dataset

sample <- sample.split(College$Apps, SplitRatio = .70)

train <- subset(College, sample == TRUE)
test  <- subset(College, sample == FALSE)

regfit.fwd=regsubsets(Apps~.,data=train, nvmax=17,method="forward")

test.mat=model.matrix(Apps~.,data=test)

validation.errors=rep(NA,17)


for(i in 1:17){
  coefi=coef(regfit.fwd,id=i)
  pred=test.mat[,names(coefi)]%*%coefi
  validation.errors[i]=mean((test$Apps-pred)^2)

}

validation.errors
```

```
 [1] 1820930 1583796 1705665 1688576 1651695 1657382 1870889 1922150
 [9] 1952357 1939974 1870591 1852722 1854180 1853134 1850381 1849124
[17] 1855638
```

```
which.min(validation.errors)
```

```
[1] 2
```

```
coef(regfit.fwd,id=which.min(validation.errors))
```

```
(Intercept)      Accept    Top10perc
-824.439455    1.362166    38.437432
```

K-Fold Cross Validation

```
set.seed(1)
k=10
folds=sample(1:k,nrow(College),replace=TRUE)
cv.errors=matrix(NA,k,17, dimnames=list(NULL, paste(1:17)))


predict.regsubsets=function(object,newdata,id,...){
  form=as.formula(object$call[[2]])
 mat=model.matrix(form,newdata)
 coefi=coef(object,id=id)
 xvars=names(coefi)
 mat[,xvars]%*%coefi
}


for(j in 1:k)
{
  best.fit=regsubsets(Apps~.,data=College[folds!=j,],nvmax=17)

  for (i in 1:17)
  {
    pred=predict.regsubsets(best.fit,College[folds==j,],id=i)
    cv.errors[j,i]=mean((College$Apps[folds==j]-pred)^2)
  }
}

mean.cv.errors=apply(cv.errors,2,mean)

which.min(mean.cv.errors)
```

```
4
4
```

Observations

If we plot minimum instance of test error we can see that different variables are chosen at different iterations, confirming a high level of overfitting to test data, best method to tackle this will be using regularization

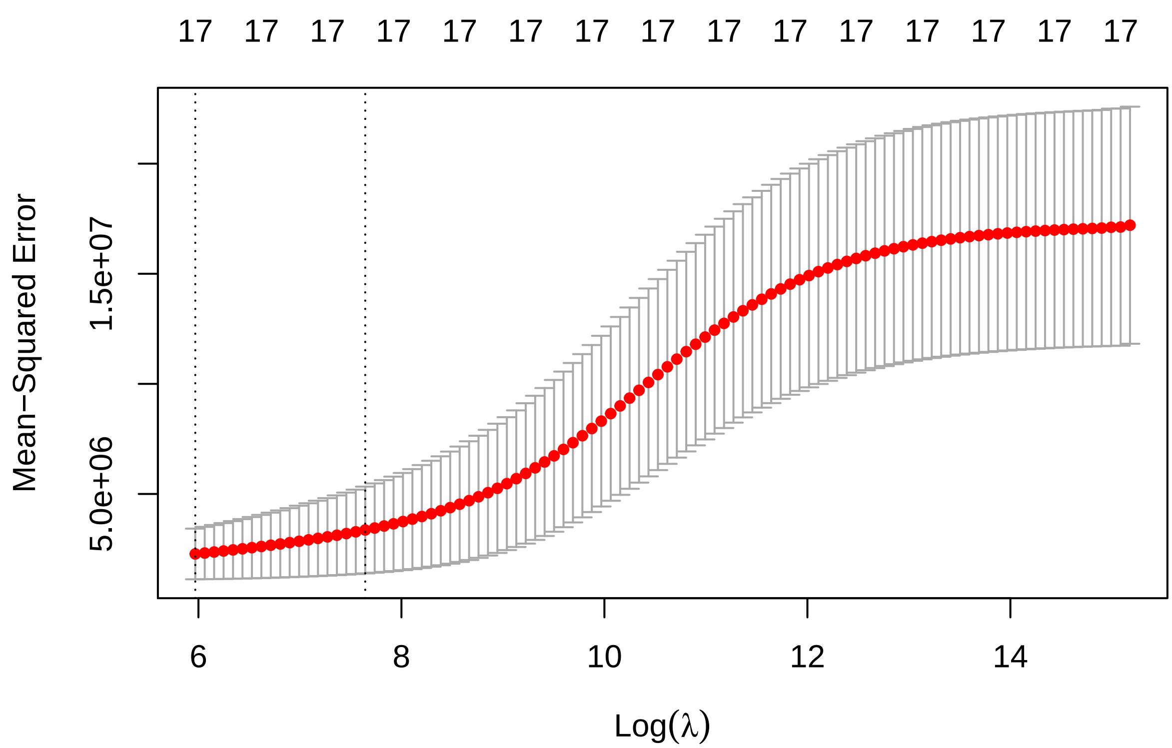
L1-Lasso Regularization & L2-Ridge Regularization

```
x=model.matrix(Apps~.,College)[,-1]
y=College$Apps

# Ridge Regression

grid=10^seq(10,-2,length=100)

ridge.mod=glmnet(x,y,alpha=0,lambda=grid)
```

```r
# Train & Test

train=sample(1:nrow(x),nrow(x)/2)

test=(-train)

y.test=y[test]

ridge.mod=glmnet(x[train,],y[train],alpha=0,lambda=grid,thresh=1e-12)

ridge.pred=predict(ridge.mod,s=4,newx=x[test,])

mean((ridge.pred-y.test)^2)
```

```
[1] 1162752
```

```r
# cross validation

set.seed(1)

cv.out=cv.glmnet(x[train,],y[train],alpha=0)
plot(cv.out)
```



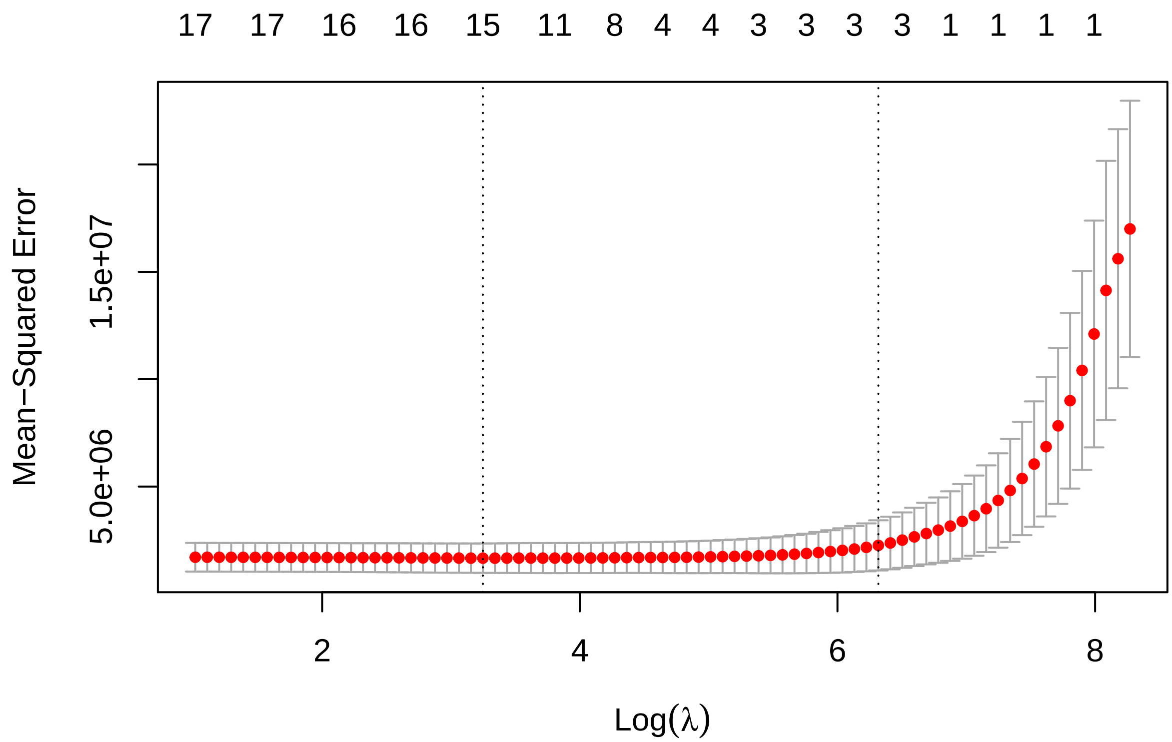

```r
bestlam=cv.out$lambda.min

bestlam
```

```
[1] 390.979
```

```
ridge.pred=predict(ridge.mod,s=bestlam,newx=x[test,])
mean((ridge.pred-y.test)^2)
```

```
[1] 1062563
```

```
# lasso regression

cv.out=cv.glmnet(x[train,],y[train],alpha=1)
plot(cv.out)
```




```
bestlam=cv.out$lambda.min

bestlam
```

```
[1] 25.72381
```

```
ridge.pred=predict(ridge.mod,s=bestlam,newx=x[test,])
mean((ridge.pred-y.test)^2)
```

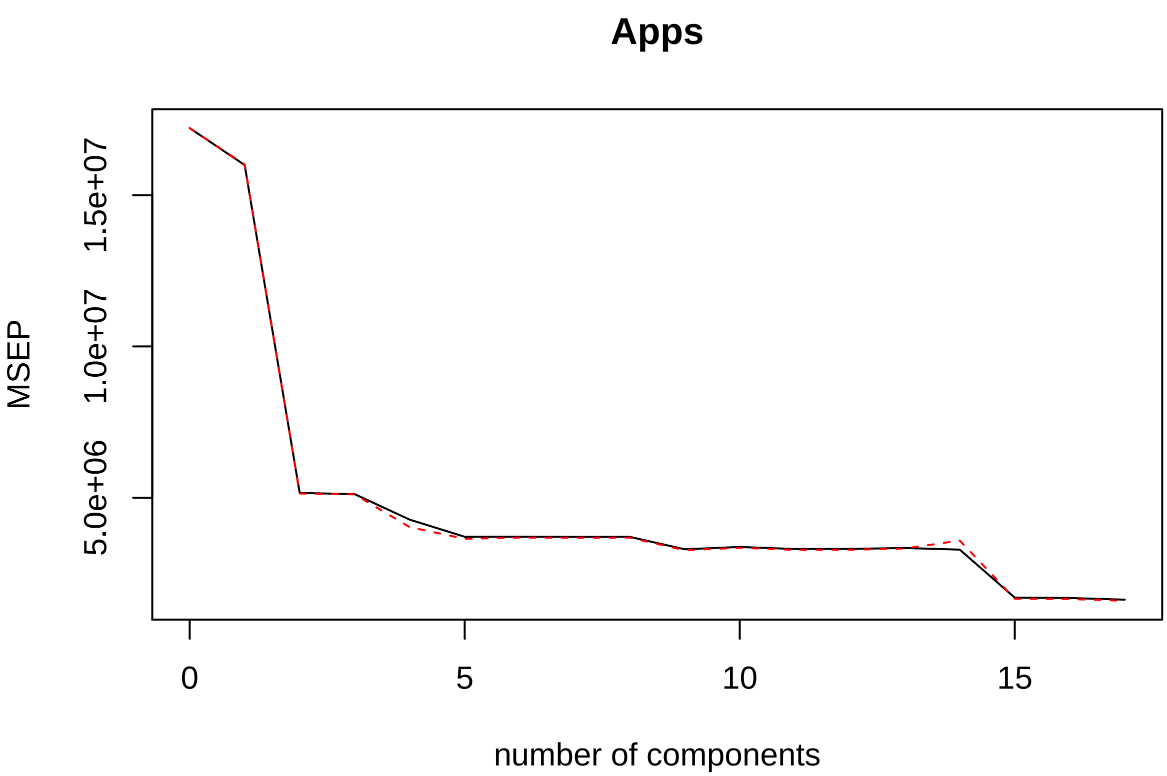
```
[1] 1145142
```

Principal Component Regression

```
set.seed(1)

pcr.fit=pcr(Apps~.,data=College,scale=TRUE,validation="CV")
pcr.fit
```

```
Principal component regression , fitted with the singular value decomposition algorithm.
Cross-validated using 10 random segments.
Call:
pcr(formula = Apps ~ ., data = College, scale = TRUE, validation = "CV")
```

```r
# try on training & testing

pcr.fit=pcr(Apps~.,data=College,subset=train,scale=TRUE,validation="CV")

validationplot(pcr.fit,val.type="MSEP")
```




```r
pcr.pred=predict(pcr.fit,x[test,],ncomp=17)
mean((pcr.pred-y.test)^2)
```

```
[1] 1166368
```