Advanced Statistical Methods Homework 6

Brandon Hosley

University of Illinois - Springfield

Advanced Statistical Methods Homework 6

Introduction to Statistical Learning
Chapter 6.8 : Problem 11

We will now try to predict per capita crime rate in the ***Boston*** *data set.*
Prepare the data set:

```
library(MASS)
attach(Boston)
library(leaps)
```

(a)

Try out some of the regression methods explored in this chapter, such as best subset selection, the lasso, ridge regression, and PCR. Present and discuss results for the approaches that you consider.

Best Subset Selection. Determine the order of subset values:

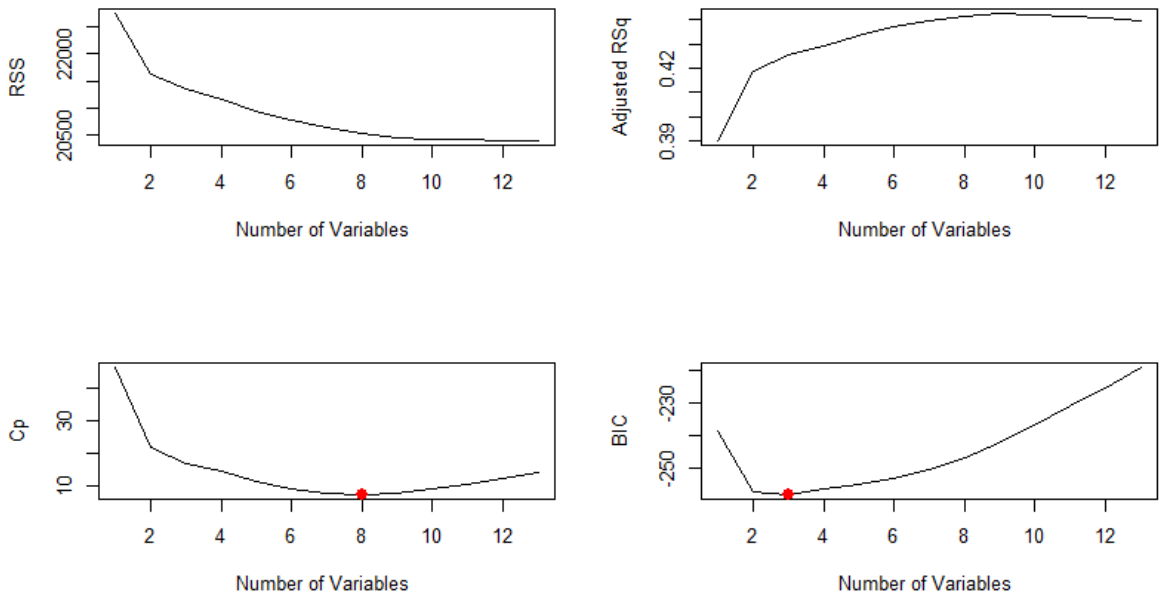
```
regfit.full=regsubsets(crim~.,Boston)
summary(regfit.full)
```

Providing the following results:

```
Subset selection object
Call: regsubsets.formula(crim ~ ., Boston)
13 Variables  (and intercept)
Forced in Forced out
zn          FALSE      FALSE
indus       FALSE      FALSE
chas        FALSE      FALSE
nox         FALSE      FALSE
rm          FALSE      FALSE
age         FALSE      FALSE
dis         FALSE      FALSE
rad         FALSE      FALSE
tax         FALSE      FALSE
ptratio     FALSE      FALSE
black       FALSE      FALSE
lstat       FALSE      FALSE
medv        FALSE      FALSE
1 subsets of each size up to 8
Selection Algorithm: exhaustive
zn  indus chas nox rm  age dis rad tax ptratio black lstat medv
1  ( 1 ) " " " "   " " " " " " " " " " "*" " " " " " "     " " "   " " "   " "
2  ( 1 ) " " " "   " " " " " " " " " " "*" " " " " " "     " " "   "*" "   " "
3  ( 1 ) " " " "   " " " " " " " " " " "*" " " " " " "     "*" "   "*" "   " "
```

```
4  ( 1 ) "*" " "   " "  " " " " " " " " "*" "*" " " " " " "     " "   " "   "*"
5  ( 1 ) "*" " "   " "  " " " " " " " " "*" "*" " " " " " "     "*"   " "   "*"
6  ( 1 ) "*" " "   " "  "*" " " " " " "*" "*" " " " " " "     "*"   " "   "*"
7  ( 1 ) "*" " "   " "  "*" " " " " " "*" "*" " " "*" "     "*"   " "   "*"
8  ( 1 ) "*" " "   " "  "*" " " " " " "*" "*" " " "*" "     "*"   "*"   "*"
```

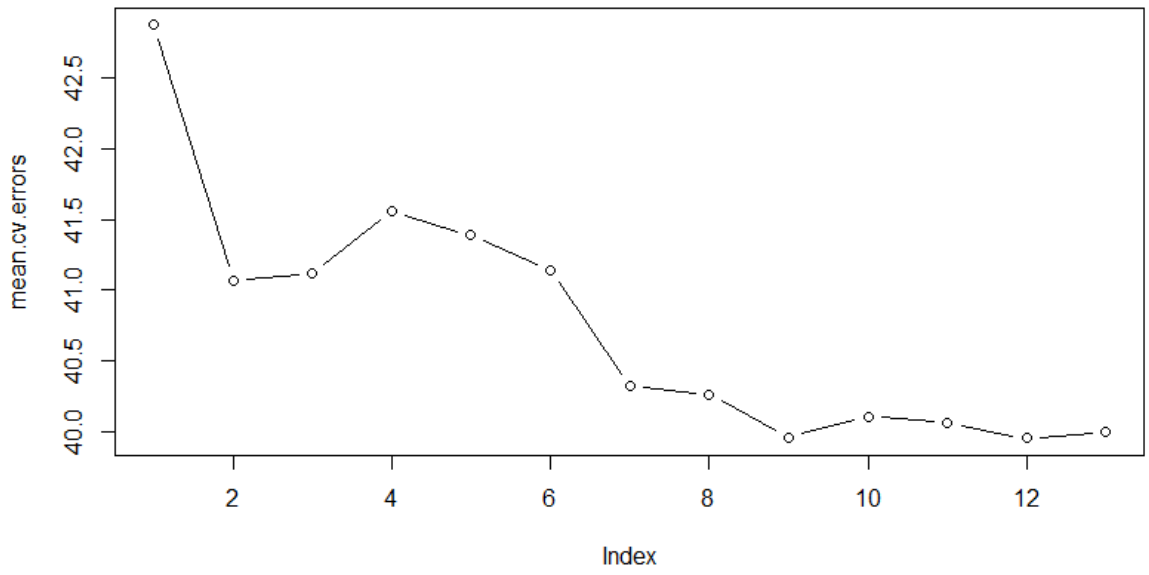
```
regfit.full=regsubsets (crim~.,data=Boston ,nvmax=13)
reg.summary =summary (regfit.full)

par(mfrow=c(2,2))
plot(reg.summary$rss ,xlab="Number of Variables ",ylab="RSS", type="l")
plot(reg.summary$adjr2 ,xlab="Number of Variables ", ylab="Adjusted RSq",
       type="l")
which.max(reg.summary$adjr2)
points (8,reg.summary$adjr2[8], col="red",cex=2,pch =20)
plot(reg.summary$cp ,xlab="Number of Variables ",ylab="Cp", type='l')
which.min(reg.summary$cp)
points (8,reg.summary$cp [8], col ="red",cex=2,pch =20)
plot(reg.summary$bic ,xlab="Number of Variables ",ylab="BIC", type='l')
which.min(reg.summary$bic )
points (3,reg.summary$bic [3],col="red",cex=2,pch =20)
```



Determine the best subset using Cross-validation as the evaluation metric.

```
set.seed(1)
train=sample(c(TRUE ,FALSE), nrow(Boston),rep=TRUE)
test=(!train)
regfit.best=regsubsets(crim~.,data=Boston[train ,], nvmax=13)
test.mat=model.matrix(crim~.,data=Boston[test ,])
val.errors =rep(NA ,13)
```

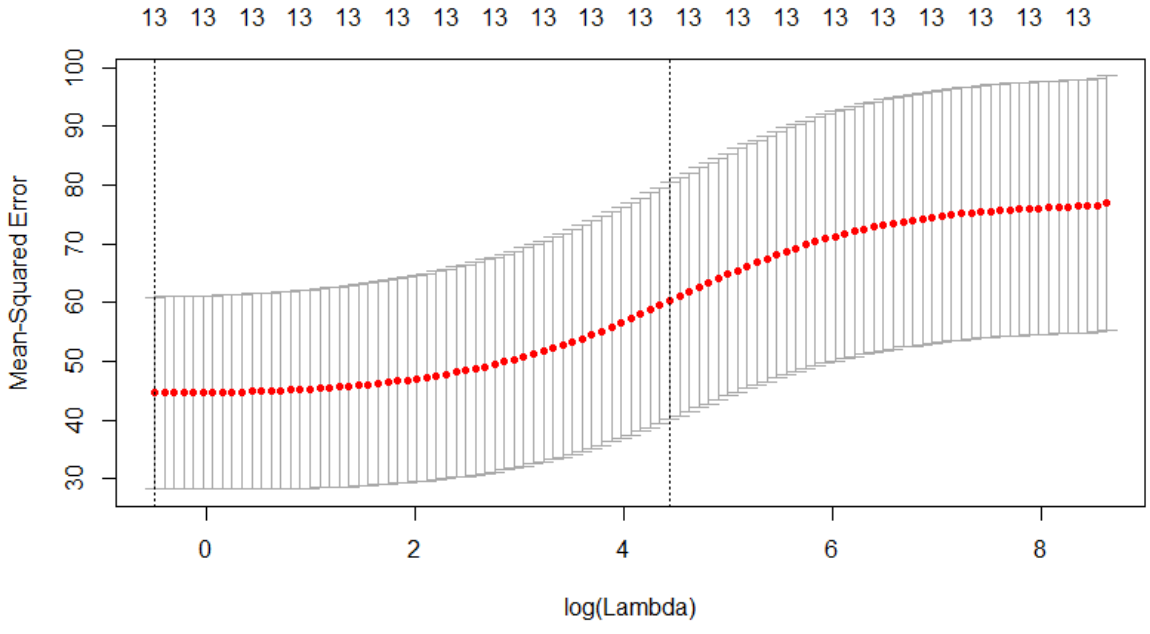
```
for(i in 1:13){coefi=coef(regfit.best ,id=i)
pred=test.mat[,names(coefi)]%*%coefi
val.errors[i]=mean((Boston$crim[test]-pred)^2)
}
## Cross-Validation
predict.regsubsets = function(object , newdata ,id ,...){
form=as.formula (object$call [[2]])
mat=model.matrix(form ,newdata )
coefi=coef(object ,id=id)
xvars=names(coefi)
mat[,xvars]%*%coefi
}
k <- 10
folds=sample (1:k,nrow(Boston),replace=TRUE)
cv.errors=matrix (NA,k,13, dimnames=list(NULL, paste (1:13)))
for(j in 1:k){
best.fit <- regsubsets(crim~., data=Boston[folds!=j,],nvmax=13)
for(i in 1:13){
pred <- predict(best.fit ,Boston[folds ==j,], id=i)
cv.errors[j,i] <- mean(( Boston$crim[folds==j]-pred)^2)
}
}
mean.cv.errors=apply(cv.errors ,2, mean)
mean.cv.errors
par(mfrow=c(1,1))
plot(mean.cv.errors ,type='b')
reg.best=regsubsets (crim~.,data=Boston , nvmax=13)
coef(reg.best ,12)
```



Ridge Regression. .

```
library(glmnet)

x=model.matrix(crim~.,Boston )[,-1]
y=Boston$crim

grid=10^seq(10,-2, length =100)
ridge.mod=glmnet (x,y,alpha=0, lambda=grid)
dim(coef(ridge.mod))

train=sample (1: nrow(x), nrow(x)/2)
test=(-train)
y.test=y[test]
## Cross-validation
cv.out=cv.glmnet(x[train ,],y[ train],alpha=0)
plot(cv.out)
bestlam =cv.out$lambda.min
bestlam
ridge.pred=predict (ridge.mod ,s=bestlam ,newx=x[test ,])
mean((ridge.pred -y.test)^2)
```

This provides an MSE of 40.62748



The Lasso. .

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

plot(cv.out)
```

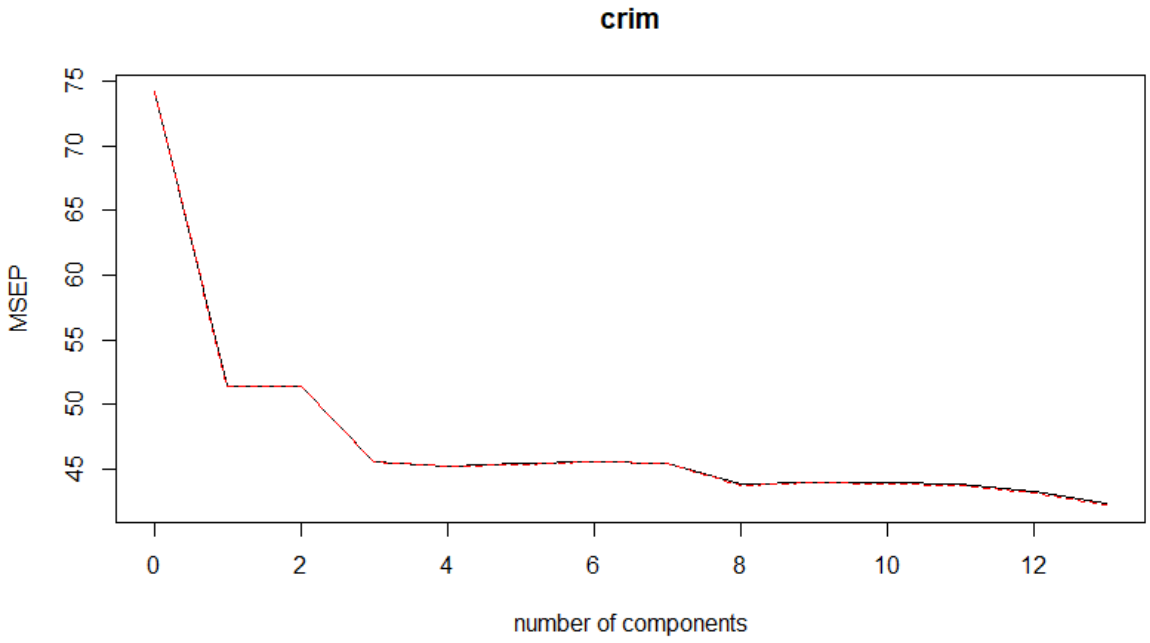
```
bestlam =cv.out$lambda.min
lasso.pred=predict (lasso.mod ,s=bestlam ,newx=x[test ,])
mean((lasso.pred -y.test)^2)

out=glmnet (x,y,alpha=1, lambda=grid)
lasso.coef=predict (out ,type="coefficients",s= bestlam) [1:14,]
lasso.coef
```

The Lasso provided an MSE of 42.21187 and eliminated 2 of the 13 variables.

PCR. *Principle Components Regression.*

```
library(pls)
set.seed(1)
pcr.fit=pcr(crim~., data=Boston, scale=TRUE, validation ="CV")
validationplot(pcr.fit ,val.type="MSEP")

pcr.pred=predict (pcr.fit ,x[test ,],ncomp =7)
mean((pcr.pred -y.test)^2)
```

PCR provides an MSE of 43.6911.



(b)

Propose a model (or set of models) that seem to perform well on this data set, and justify your answer. Make sure that you are evaluating model performance using validation set error, cross-validation, or some other reasonable alternative, as opposed to using training error.

Of the above models the subset selection provided option with the lowest MSE.

(c)

Does your chosen model involve all of the features in the data set? Why or why not?

The two lowest MSE subsets were the 12 feature and the 9 feature subsets. The while the 12 feature set provides marginal improvement over the 9 feature set, if there is an issue with over-fitting the 12 feature set is more likely to be over-fit. For this model the efficiency is not really a concern, but if it needed to be applied on a large scale, the 9 feature set will be significantly more efficient in calculating predictions.