­

Ch6. Linear model selection & Regularization

-Lab & Exercises #1,2,5,9,11

­­

**학번 : 182STG12**

**이름 : 오혜윤**

c

Problem Description

In Chapter 6 lab, we learn about linear model selection methods. Using selection methods, we fit the model and compare the test errors. There are selection methods such as best subset regression, forward selection method and backward selection method. By using these selection methods, we can compare the test errors and find the best model. Also, if we want to choose the number of components, we use ridge regression, lasso, PCR and PLS. Comparing these methods, we can choose the model with the optimal number of components.

We split the training set and the test dataset. Then, we fit the models and calculate the test errors and cv errors. Comparing the errors, we can find the best subset model with the optimal number of components.

In Chapter 6 exercises, we compare the results of many subset models. In the first question, we think about the theory of selection methods. In the second question, we think about the theory of lasso and ridge regression methods. In the third question, we think about the ridge regression optimization. Finally, in the fourth and the fifth questions, we use the real data and fit the models. We compare the errors of models and find the best model.

Results

**CH6. Lab Review**

In this lab, it was mostly about linear model selection. First, using Hitters dataset, we learn about model selection approach to estimate the test error rates and find the best linear model. There are the model selection approaches such as best subset selection, forward stepwise selection and backward stepwise selection. We begin by using the regsubsets() function to perform selection. Then, by comparing R square, Cp or MSE, we can select the best model with less error.

After that, we learn about the validation set approach. First, we split the training set and test dataset. Then, by fitting the model selection methods we learned before, we do validation and compare the cv errors.

Finally, we learn about ridge regression, lasso, PCR and PLS. By using these methods, we can choose the number of components. We can find the number of components when the lowest cross-validation error occurs.

By using model selection methods, we can choose the best model which represents dataset the best.

**CH6 Exercises**

1. **We perform best subset, forward stepwise, and backward stepwise selection on a single data set. For each approach, we obtain p + 1 models, containing 0, 1, 2, . . . , p predictors. Explain your answers:**
2. **Which of the three models with k predictors has the smallest training RSS?**

Best subset selection model has the largest probability that has the smallest training RSS given k predictors. This is because the best selection model will be chosen after considering all the possible models with k parameters for best subset. When performing forward stepwise selection, the model with k predictors is the model with the smallest RSS among the p−k models which augment the predictors in with one additional predictor. When performing backward stepwise selection, the model with k predictors is the model with the smallest RSS among the k models which contains all but one of the predictors in . So, the model with k predictors which has the smallest training RSS is the one obtained from best subset selection as it is the one selected among all k predictors models.

1. **Which of the three models with k predictors has the smallest test RSS?**

Difficult to answer: best subset selection may have the smallest test RSS because it describes more models than the other methods. However, the other methods can also pick a model with smaller test RSS.

**(c) True or False:**

**i. The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the (k+1)-variable model identified by forward stepwise selection.**

TRUE: The model with (k+1) predictors is obtained by augmenting the predictors in the model with k predictors with one additional predictor.

**ii. The predictors in the k-variable model identified by backward stepwise are a subset of the predictors in the (k + 1)-variable model identified by backward stepwise selection.**

TRUE: The model with k predictors is obtained by removing one predictor from the model with (k+1) predictors.

**iii. The predictors in the k-variable model identified by backward stepwise are a subset of the predictors in the (k + 1)- variable model identified by forward stepwise selection.**

FALSE: There is no direct link between the models obtained from forward and backward selection.

**iv. The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the (k+1)-variable model identified by backward stepwise selection.**

FALSE: There is no direct link between the models obtained from forward and backward selection.

**v. The predictors in the k-variable model identified by best subset are a subset of the predictors in the (k + 1)-variable model identified by best subset selection.**

FALSE: The model with (k+1) predictors is obtained by selecting among all possible models with (k+1) predictors, and so does not necessarily contain all the predictors selected for the k-variable model.

**2. For parts (a) through (c), indicate which of i. through iv. is correct. Justify your answer.**

**(a) The lasso, relative to least squares, is:**

**i. More flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.**

**ii. More flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.**

**iii. Less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.**

**iv. Less flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.**

iii is the correct answer. The lasso is less flexible and will give an improved prediction accuracy when its increase in bias is less than its decrease in variance. This is because the lasso is a more restrictive model, and thus it has the possibility of reducing overfitting and variance in predictions. If it does not result in too high of a bias due to its added constraints, it will outperform least squares which might be fitting spurious parameters.

**(b) Repeat (a) for ridge regression relative to least squares.**

iii is the correct answer. Ridge regression is less flexible than the least squares method and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

**(c) Repeat (a) for non-linear methods relative to least squares.**

ii is the correct answer. Non-linear methods are generally more flexible than least squares. They perform better when the linearity assumption is strongly broken. These methods will have more variance due to their more sensitive fits to the underlying data, and to perform well will need to have a substantial drop in bias.

**5. It is well-known that ridge regression tends to give similar coefficient values to correlated variables, whereas the lasso may give quite different coefficient values to correlated variables. We will now explore this property in a very simple setting. Suppose that n = 2, p = 2, x11 = x12, x21 = x22. Furthermore, suppose that y1+y2 = 0 and x11+x21 = 0 and x12+x22 = 0, so that the estimate for the intercept in a least squares, ridge regression, or lasso model is zero: = 0.**

**(a) Write out the ridge regression optimization problem in this setting.**

Because the intercept is zero and there are two variables, x1 and x2, the model which the ridge regression tries to minimize is

.

**(b) Argue that in this setting, the ridge coefficient estimates satisfy** =.

By taking the derivatives of the above expression with respect to and and setting them equal to 0, we obtain the two equations below.

By subtracting the two expressions above, we get=.

**(c) Write out the lasso optimization problem in this setting.**

Because the intercept is zero and there are two variables, x1 and x2, the model which the lasso optimization tries to minimize is

.

**(d) Argue that in this setting, the lasso coefficients ˆ β1 and ˆ β2 are not unique—in other words, there are many possible solutions to the optimization problem in (c). Describe these solutions.**

Another formulation of lasso is

subject to .

,and by solving this equation, we can get the coefficients.

Geometrically the lasso constraint takes the form of a diamond centered at the origin of the plane (,) which intersects the axes at a distance s from the origin. By using the setting of this problem ==, ==, =0 and =0, we must minimize the expression below.

.

This optimization problem has a simple solution: =/. Geometrically, this is a line parallel to the edge of the diamond of the constraints. Now, solutions to the lasso optimization problem are contours of the function that intersects the diamond of the constraints. So, the entire edge =s (as is the edge =−s) is a potential solution to the lasso optimization problem. Thus, the lasso optimization problem has a whole set of solutions instead of a unique one.

**9. In this exercise, we will predict the number of applications received using the other variables in the College data set.**

**(a) Split the data set into a training set and a test set.**

|  |
| --- |
| R codes: |
| >set.seed(1)  >train=sample(c(TRUE,FALSE),nrow(College),rep=TRUE)  >test=(!train)  >College.train=College[train,,drop=F]  >College.test=College[test,,drop=F] |

**(b) Fit a linear model using least squares on the training set, and report the test error obtained.**

The test error is 1520331.

1. **Fit a ridge regression model on the training set, with λ chosen by cross-validation. Report the test error obtained.**

The value of chosen λ is 28.4804. Also, the test error is 1664103.

1. **Fit a lasso model on the training set, with λ chosen by cross validation. Report the test error obtained, along with the number of non-zero coefficient estimates.**

The value of chosen λ is 14.17474. Also, the test error is 1615301.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Coefficients: | Intercept | PrivateYes | Accept | Top10perc |
|  | -8.7602e+02 | -4.89E+02 | 1.15E+00 | 4.27E+01 |

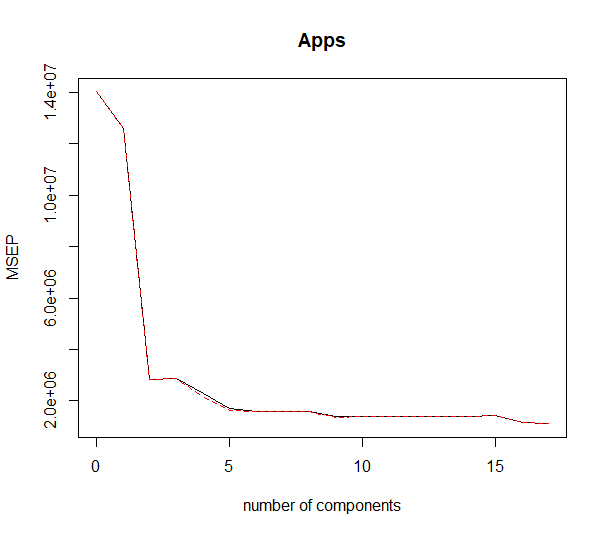
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Coefficients: | Top25perc | F.Undergrad | P.Undergrad | Outstate |
|  | -8.73E+00 | 6.99E-02 | 2.06E-02 | -2.52E-03 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Coefficients: | Room.Board | Books | Personal | PhD |
|  | 1.03E-01 | 1.11E-01 | 1.43E-01 | -6.86E+00 |

|  |  |  |  |
| --- | --- | --- | --- |
| Coefficients: | perc.alumni | Expend | Grad.Rate |
|  | -3.47E+00 | 5.58E-02 | 5.19E+00 |

1. **Fit a PCR model on the training set, with M chosen by cross validation. Report the test error obtained, along with the value of M selected by cross-validation.**

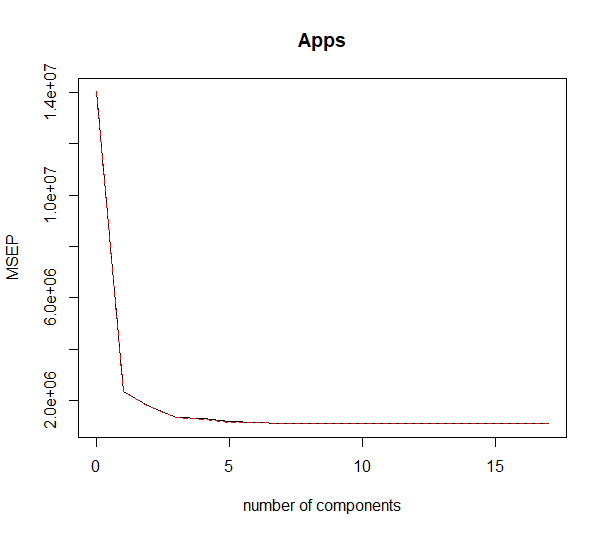
The graph below is the PCR validation plot which represents the best number of components.



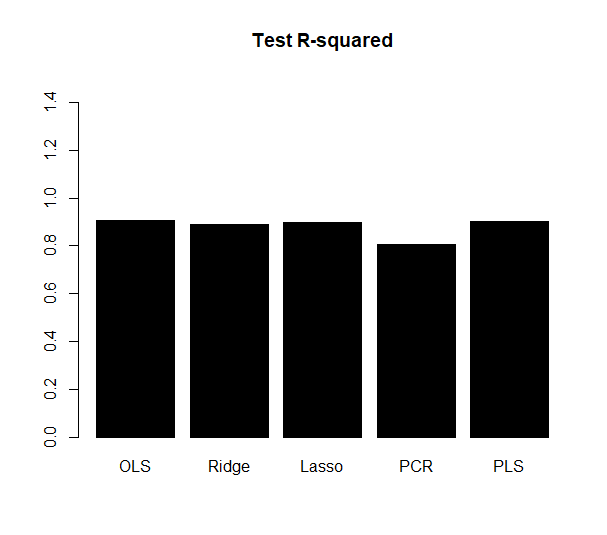
We can choose 10 as the number of components. When we chose 10 as the number of components, we obtain the test error, 3142295.

1. **Fit a PLS model on the training set, with M chosen by cross validation. Report the test error obtained, along with the value of M selected by cross-validation.**

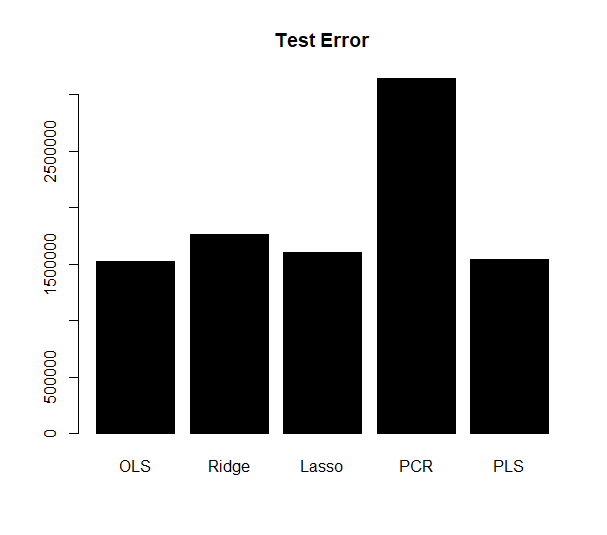
The graph below is the PLS validation plot which represents the best number of components.



We can choose 10 as the number of components. When we chose 10 as the number of components, we obtain the test error, 1538778.

1. **Comment on the results obtained. How accurately can we predict the number of college applications received? Is there much difference among the test errors resulting from these five approaches?**

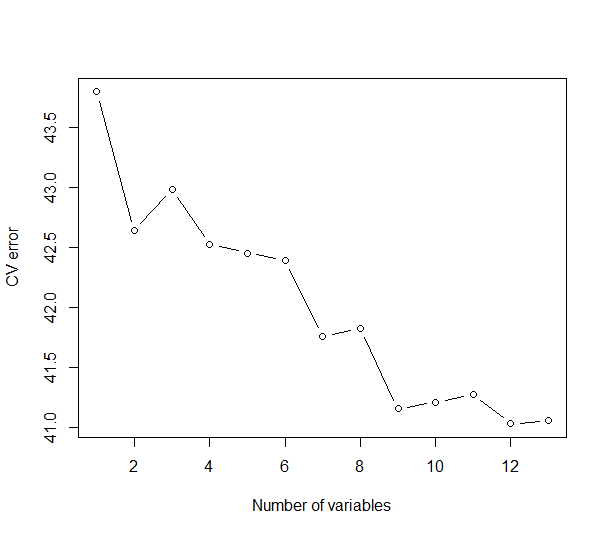
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test R-squared | | | | |
| "OLS” | "Ridge" | "Lasso" | "PCR" | "PLS" |
| 0.9052 | 0.8902 | 0.8999 | 0.8040 | 0.9040 |

The values of R-squared of the models are very high except for PCR model. The PCR model has 80% test R-squared value. Other models such as OLS, Ridge, Lasso and PLS have about 90% test R-squared values.

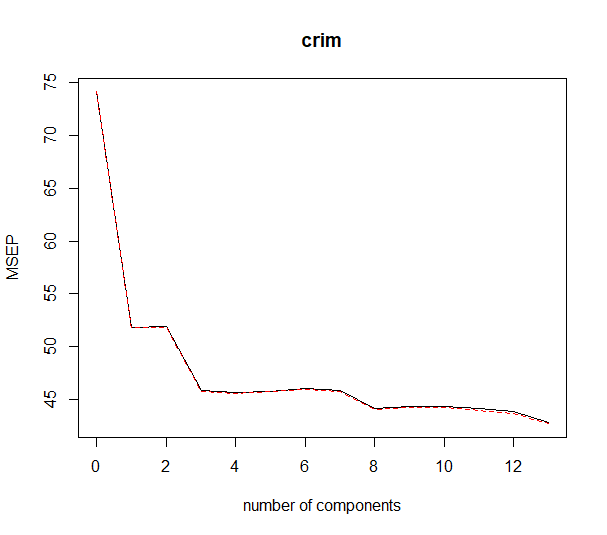
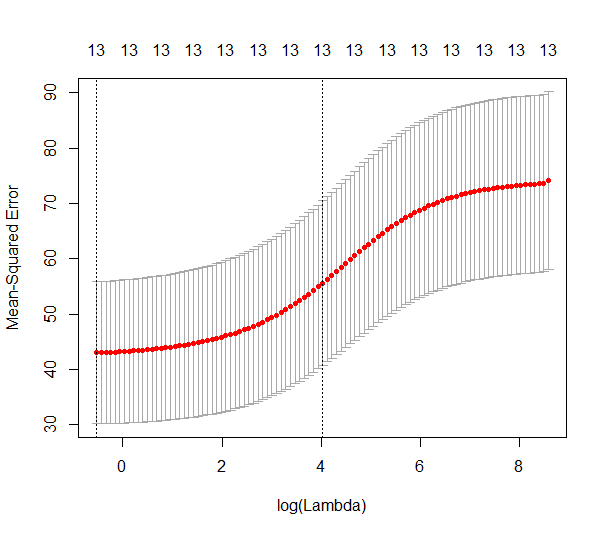
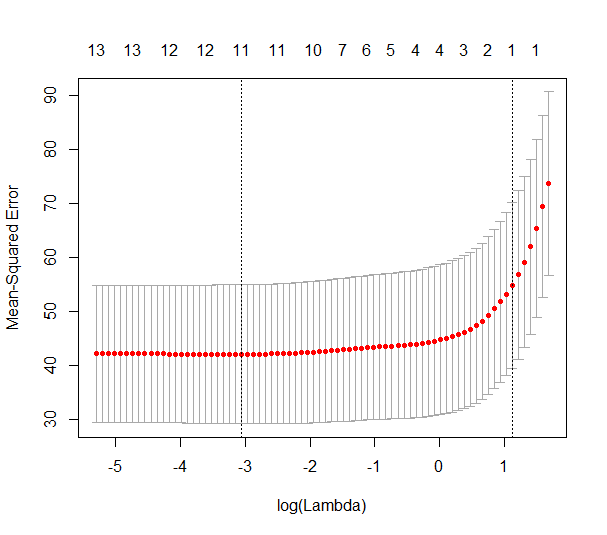
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Error | | | | |
| "OLS” | "Ridge" | "Lasso" | "PCR" | "PLS" |
| 1520331 | 1664103 | 1615301 | 3142295 | 1538778 |

The values of test error of the models are about 1600000 except for PCR model. The PCR model has 3142295 test error value. Other models such as OLS, Ridge, Lasso and PLS have 1600000 test error values. So, we can conclude that except for PCR model, all models have high accuracy.

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

**(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.**

By using regsubsets() function, we can obtain the number of variables which makes the CV error the lowest. The plot above represents the relationship of the CV errors and the number of variables. The number of variables when the CV error is the lowest (41.0346) is 12.



The plots above indicate the CV mse plot of lasso, ridge regression and PCR. First, cross-validation of lasso model selects a λ equal to 0.0467. We have a CV estimate for the test MSE equal to 42.1343. Second, cross-validation of ridge regression model selects a λ equal to 0.5375. We have a CV estimate for the test MSE equal to 42.9835. Last, cross-validation selects M to be equal to 14 (so, no dimension reduction). We have a CV estimate for the test MSE equal to 45.6936.

|  |  |  |
| --- | --- | --- |
| Cross-validation error | | |
| “Best subset selection” | "Lasso" | “Ridge” |
| **41.0346** | 42.1343 | 42.9835 |

**(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.**

As computed above, the model with the lowest cross-validation error is the one chosen by the best subset selection method which has 41.0346 cross-validation error.

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

No, the model chosen by the best subset selection method has only 13 predictors.

river?

Discussion

We applied various validation approaches to many models. In the College dataset, we split dataset into the training set and the test set. Then, we fitted the five selection methods ("OLS”,"Ridge","Lasso","PCR","PLS") into the datasets. Comparing the test errors, we could find the best model. In the Boston dataset, we split the observations into the training set and the validation set. Then, we fitted the selection models into the dataset. Comparing the prediction values and the validation set, we calculated the cross-validation error. In the Boston dataset, by using cross validation methods, we could find that the best subset selection method has the least cross validation error.

Appendix (R

**R codes**

# Chapter 6 Lab: Subset Selection Methods

# Best Subset Selection

library(ISLR)

fix(Hitters);names(Hitters);dim(Hitters)

sum(is.na(Hitters$Salary))

Hitters=na.omit(Hitters);dim(Hitters);sum(is.na(Hitters))

library(leaps)

regfit.full=regsubsets(Salary~.,Hitters);summary(regfit.full)

regfit.full=regsubsets(Salary~.,data=Hitters,nvmax=19);reg.summary=summary(regfit.full)

names(reg.summary);reg.summary$rsq

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$adjr2);points(11,reg.summary$adjr2[11], col="red",cex=2,pch=20)

plot(reg.summary$cp,xlab="Number of Variables",ylab="Cp",type='l')

which.min(reg.summary$cp);points(10,reg.summary$cp[10],col="red",cex=2,pch=20)

which.min(reg.summary$bic);plot(reg.summary$bic,xlab="Number of Variables",ylab="BIC",type='l')

points(6,reg.summary$bic[6],col="red",cex=2,pch=20)

plot(regfit.full,scale="r2")

plot(regfit.full,scale="adjr2")

plot(regfit.full,scale="Cp")

plot(regfit.full,scale="bic")

coef(regfit.full,6)

# Forward and Backward Stepwise Selection

regfit.fwd=regsubsets(Salary~.,data=Hitters,nvmax=19,method="forward");summary(regfit.fwd)

regfit.bwd=regsubsets(Salary~.,data=Hitters,nvmax=19,method="backward");summary(regfit.bwd)

coef(regfit.full,7);coef(regfit.fwd,7);coef(regfit.bwd,7)

# Choosing Among Models

set.seed(1)

train=sample(c(TRUE,FALSE), nrow(Hitters),rep=TRUE)

test=(!train)

regfit.best=regsubsets(Salary~.,data=Hitters[train,],nvmax=19)

test.mat=model.matrix(Salary~.,data=Hitters[test,])

val.errors=rep(NA,19)

for(i in 1:19){

coefi=coef(regfit.best,id=i)

pred=test.mat[,names(coefi)]%\*%coefi

val.errors[i]=mean((Hitters$Salary[test]-pred)^2)

}

val.errors

which.min(val.errors)

coef(regfit.best,10)

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

}

regfit.best=regsubsets(Salary~.,data=Hitters,nvmax=19);coef(regfit.best,10)

k=10

set.seed(1)

folds=sample(1:k,nrow(Hitters),replace=TRUE)

cv.errors=matrix(NA,k,19, dimnames=list(NULL, paste(1:19)))

for(j in 1:k){

best.fit=regsubsets(Salary~.,data=Hitters[folds!=j,],nvmax=19)

for(i in 1:19){

pred=predict(best.fit,Hitters[folds==j,],id=i)

cv.errors[j,i]=mean( (Hitters$Salary[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(Salary~.,data=Hitters, nvmax=19)

coef(reg.best,11)

# Ridge Regression and the Lasso

x=model.matrix(Salary~.,Hitters)[,-1];y=Hitters$Salary

# Ridge Regression

library(glmnet)

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

ridge.mod=glmnet(x,y,alpha=0,lambda=grid);dim(coef(ridge.mod));ridge.mod$lambda[50];

coef(ridge.mod)[,50];sqrt(sum(coef(ridge.mod)[-1,50]^2));ridge.mod$lambda[60]

coef(ridge.mod)[,60];sqrt(sum(coef(ridge.mod)[-1,60]^2))

predict(ridge.mod,s=50,type="coefficients")[1:20,]

set.seed(1)

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);mean((mean(y[train])-y.test)^2)

ridge.pred=predict(ridge.mod,s=1e10,newx=x[test,]);mean((ridge.pred-y.test)^2)

ridge.pred=predict(ridge.mod,s=0,newx=x[test,],exact=T,x=x[train,],y=y[train]);mean((ridge.pred-y.test)^2)

lm(y~x, subset=train)

predict(ridge.mod,s=0,exact=T,type="coefficients",x=x[train,],y=y[train])[1:20,]

set.seed(1)

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)

out=glmnet(x,y,alpha=0);predict(out,type="coefficients",s=bestlam)[1:20,]

# The Lasso

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

set.seed(1)

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)[1:20,];lasso.coef

lasso.coef[lasso.coef!=0]

# PCR and PLS Regression

# Principal Components Regression

library(pls)

set.seed(2)

pcr.fit=pcr(Salary~., data=Hitters,scale=TRUE,validation="CV");summary(pcr.fit)

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

set.seed(1)

pcr.fit=pcr(Salary~., data=Hitters,subset=train,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.fit=pcr(y~x,scale=TRUE,ncomp=7);summary(pcr.fit)

# Partial Least Squares

set.seed(1)

pls.fit=plsr(Salary~., data=Hitters,subset=train,scale=TRUE, validation="CV");summary(pls.fit)

validationplot(pls.fit,val.type="MSEP")

pls.pred=predict(pls.fit,x[test,],ncomp=2);mean((pls.pred-y.test)^2)

pls.fit=plsr(Salary~., data=Hitters,scale=TRUE,ncomp=2);summary(pls.fit)

#exercises 1,2,5,9,11

##9-----------------------------------------------------------------------

library(ISLR);data(College)

###a)---------------------------------------------------------------------

set.seed(1)

train=sample(c(TRUE,FALSE),nrow(College),rep=TRUE);test=(!train)

College.train=College[train,,drop=F];College.test=College[test,,drop=F]

###b)---------------------------------------------------------------------

lm.fit <- lm(Apps ~ ., data = College.train); pred.lm <- predict(lm.fit, College.test)

mean((pred.lm - College.test$Apps)^2)

###c)---------------------------------------------------------------------

library(glmnet)

train.mat <- model.matrix(Apps ~ ., data = College.train);test.mat <- model.matrix(Apps ~ ., data = College.test)

grid <- 10 ^ seq(4, -2, length = 100)

ridge.fit <- glmnet(train.mat, College.train$Apps, alpha = 0, lambda = grid, thresh = 1e-12)

cv.ridge <- cv.glmnet(train.mat, College.train$Apps, alpha = 0, lambda = grid, thresh = 1e-12)

bestlam.ridge <- cv.ridge$lambda.min;bestlam.ridge

pred.ridge <- predict(ridge.fit, s = bestlam.ridge, newx = test.mat)

mean((pred.ridge - College.test$Apps)^2)

###d)---------------------------------------------------------------------

lasso.fit <- glmnet(train.mat, College.train$Apps, alpha = 1, lambda = grid, thresh = 1e-12)

cv.lasso <- cv.glmnet(train.mat, College.train$Apps, alpha = 1, lambda = grid, thresh = 1e-12)

bestlam.lasso <- cv.lasso$lambda.min; bestlam.lasso

pred.lasso <- predict(lasso.fit, s = bestlam.lasso, newx = test.mat);mean((pred.lasso - College.test$Apps)^2)

predict(lasso.fit, s = bestlam.lasso, type = "coefficients")

###e)---------------------------------------------------------------------

library(pls)

pcr.fit <- pcr(Apps ~ ., data = College.train, scale = TRUE, validation = "CV") ; summary(pcr.fit)

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

pred.pcr <- predict(pcr.fit, College.test, ncomp = 10);mean((pred.pcr - College.test$Apps)^2)

###f)---------------------------------------------------------------------

pls.fit <- plsr(Apps ~ ., data = College.train, scale = TRUE, validation = "CV");summary(pls.fit)

validationplot(pls.fit, val.type = "MSEP")

pred.pls <- predict(pls.fit, College.test, ncomp = 10);mean((pred.pls - College.test$Apps)^2)

###g)---------------------------------------------------------------------

#r2

test.avg <- mean(College.test$Apps)

lm.r2 <- 1 - mean((pred.lm - College.test$Apps)^2) / mean((test.avg - College.test$Apps)^2)

ridge.r2 <- 1 - mean((pred.ridge - College.test$Apps)^2) / mean((test.avg - College.test$Apps)^2)

lasso.r2 <- 1 - mean((pred.lasso - College.test$Apps)^2) / mean((test.avg - College.test$Apps)^2)

pcr.r2 <- 1 - mean((pred.pcr - College.test$Apps)^2) / mean((test.avg - College.test$Apps)^2)

pls.r2 <- 1 - mean((pred.pls - College.test$Apps)^2) / mean((test.avg - College.test$Apps)^2)

barplot(c(lm.r2, ridge.r2, lasso.r2, pcr.r2, pls.r2), col="black", names.arg=c("OLS", "Ridge", "Lasso", "PCR", "PLS"), main="Test R-squared",ylim = c(0,1.5))

#test error

lm.test.err<-mean((pred.lm - College.test$Apps)^2) ;ridge.test.err<-mean((pred.ridge - College.test$Apps)^2)

lasso.test.err<-mean((pred.lasso - College.test$Apps)^2);pcr.test.err<-mean((pred.pcr - College.test$Apps)^2)

pls.test.err<-mean((pred.pls - College.test$Apps)^2)

barplot(c(lm.test.err, ridge.test.err, lasso.test.err, pcr.test.err, pls.test.err), col="black", names.arg=c("OLS", "Ridge", "Lasso", "PCR", "PLS"), main="Test Error")

##11-----------------------------------------------------------------------

library(MASS);library(leaps);data(Boston);set.seed(1)

###a)---------------------------------------------------------------------

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);plot(mean.cv.errors, type = "b", xlab = "Number of variables", ylab = "CV error")

which.min(mean.cv.errors) ;mean.cv.errors[which.min(mean.cv.errors)]

x = model.matrix(crim ~ . - 1, data = Boston);y = Boston$crim

cv.lasso = cv.glmnet(x, y, type.measure = "mse");plot(cv.lasso) ;coef(cv.lasso) ;

cv.ridge = cv.glmnet(x, y, type.measure = "mse", alpha = 0);plot(cv.ridge); coef(cv.ridge);

pcr.crime = pcr(crim ~ ., data = Boston, scale = TRUE, validation = "CV");validationplot(pcr.crime, val.type = "MSEP")