Andrew Hillard

Data 3

Homework 3

**Problem 1**

a) I used the sample function to split the data into a training and test set.

b) The test MSE is 1108531. All possible predictors used in linear model.

c) The test MSE is 1037616. The lambda chosen from cross-validation was 450.7435.

d) The test MSE is 1030941. The number of non-zero coefficients, not including the intercept, is 15. The lambda chosen from cross-validation was 24.62086.

e) The test MSE is 1166897. The value of M chosen from cross-validation is 13.

f) The test MSE is 1134531. The value of M chosen from cross-validation is 10.

g) The ridge regression and lasso model performed much better than the other models and had a test MSE that was approximately 100000 less. The test MSE’s for the ridge and lasso models were very similar. Since the mean test error converted to the original units should be approximately 1000 applications, and the lasso and ridge regression are methods that help to control variability, the predictions based on the these models should be fairly accurate.

**Problem 2**

a) I created the training and test set using the same method as shown on page 248 of Lab 5.

b) The test MSE is 59.36511.

c) The test MSE is 59.3044. The number of variables chosen from cross-validation was 12.

d) The test MSE is 58.26644. The lambda chosen from cross-validation was 0.5412185.

e) The test MSE is 56.26661. The number of non-zero coefficients, not including the intercept, is 6. The lambda chosen from cross-validation was 0.2512114. The following variables have parameters estimated to be zero: indus, chas, nox, age, tax, ptratio, and black.

f) The test MSE is 56.59548. The value of M chosen from cross-validation is 8, which, while it doesn’t yield the lowest test MSE from the 10-fold CV of the training set, is the number of components at which the test MSE levels out.

g) The test MSE is 56.24453. The value of M chosen from cross-validation is 4, which, while it doesn’t yield the lowest test MSE from the 10-fold CV of the training set, is the number of components at which the test MSE levels out.

h) Based on the test MSE, the predictions for per capita crime rate shouldn’t be very accurate. The mean of the crime rate is 3.61, the median is 0.256510, and the 75th percentile is 3.677083. The test MSE, which is approximately 60, translates to an approximate error of 7.5 in the original units. That is a large error to make if the average observation is 3.61 and 75% of the data is below 3.677. There is a difference between test error rates but not a large one. The ridge regression, PCR, and PLS performed the best. Based on the Lasso and the best subsets method, the most important predictors are zn, rm, dis, rad, lstat, and medv. If I were analyzing this data again, I would want to choose M for PCR and PLS using the one standard error rule instead of eyeballing it.

**Problem 3**

a) The training and test set are created using the technique on page 248 of Lab 5.

* Ridge Regression: The test MSE is 18.35763. The lambda value chosen from the 5-fold cross validation on the training set is 5.83721. All the variables are used in the ridge regression.
* Lasso Regression: The test MSE is 18.51826. The lambda value chosen from the 5-fold cross validation on the training set is 0.61151. The only non-zero coefficients left in the model are are Medu, failures, and higheryes.
* PCR: The test MSE is 16.60168. The M value chosen from the 5-fold cross validation on the training set is 25. All the variables are weighted in the components.
* PLS: The test MSE is 18.78685. The M value chosen from the 5-fold cross validation on the training set is 1. All the variables are weighted in this component.

Of the models analyzed, the model that performed best on the test dataset was the PLS model. The test MSE for PLS was substantially different and lower than the other models.

b) The training and test set are created using the technique on page 248 of Lab 5.

* Ridge Regression: The test MSE is 4.392806. The lambda value chosen from the 5-fold cross validation on the training set is 0.4556321. All the variables are used in the ridge regression.
* Lasso Regression: The test MSE is 3.466583. The lambda value chosen from the 5-fold cross validation on the training set is 0.1755791. The only non-zero coefficients left in the model are schoolMS, reasonhome, failures, paidyes, nurseryyes, famrel, absences, G1, and G2.
* PCR: The test MSE is 3.265469. The M value chosen from the 5-fold cross validation on the training set is 39, which is where the CV test MSE finally levels out. All the variables are used and weighted in the components.
* PLS: The test MSE is 3.381085. The M value chosen from the 5-fold cross validation on the training set is 5, which is where the CV test MSE starts to level out. All the variables are weighted in this component.

Of the models analyzed, the PCR model appears to perform the best on the test set of data. However, the test MSE’s are very close for Lasso, PCR, and PLS, and it’s hard to say if the PCR model is overall better, or just better for this test set.

c) I am unsure why I am splitting the data set into a training and test dataset. Why wouldn’t I use cross-validation to estimate the test MSE rather than using the test set? I am also unsure the best way to decide the number of M components in the PCR and PLS. I am deciding by using the graph and seeing where the CV value levels off. Besides that, I can’t think of other difficulties I had.

d)

setwd("/Users/ahillard/Documents/2016 Fall Semester/Data 3")

p3 = read.table("student-mat.csv", sep=";", header=TRUE)

head(p3)

#Ridge Regression

set.seed(1)

train = sample(c(TRUE ,FALSE), nrow(p3),rep=TRUE)

test = (!train)

x <- model.matrix(G3~., data=p3)[,-c(1,41:42)]

y <- p3$G3

set.seed(1)

ridge.mod = glmnet(x[train,], y[train], alpha=0)

set.seed(1)

cv.out = cv.glmnet(x[train,], y[train], alpha=0, nfolds=5)

bestlam=cv.out$lambda.min

bestlam

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

mean((ridge.pred-y[test])^2) #18.35763 Test MSE

#Lasso Regression

set.seed(1)

lasso.mod = glmnet(x[train,], y[train], alpha=1)

set.seed(1)

cv.out = cv.glmnet(x[train,], y[train], alpha=1, nfolds=5)

bestlam=cv.out$lambda.min

bestlam

lasso.pred = predict(lasso.mod, s=bestlam, newx=x[test,])

mean((lasso.pred-y[test])^2) #18.51826 Test MSE

lasso.mod1 = glmnet(x[train,], y[train], alpha=1, lambda=bestlam)

coefficients(lasso.mod1)

#PC Regression

p3a <- as.data.frame(cbind(y,x))

names(p3a)[1] <- "G3"

set.seed(1)

pcr.fit = pcr(G3~., data=p3a, scale=TRUE, validation="CV", segments=5)

validationplot(pcr.fit, val.type="MSEP") #25 comps, 4.335

summary(pcr.fit)

4.335^2 #18.7922 CV Test MSE

pcr.pred=predict(pcr.fit, x[test,], ncomp=25)

mean((pcr.pred - y[test])^2) #16.60168 Test MSE

#PLS Regression

p3a <- as.data.frame(cbind(y,x))

names(p3a)[1] <- "G3"

set.seed(1)

pls.fit = plsr(G3~., data=p3a, scale=TRUE, validation="CV", segments=5)

validationplot(pls.fit, val.type="MSEP") #1 Comp, 4.333

summary(pls.fit)

4.333^2 #18.77489 CV Test MSE

pcr.pred=predict(pcr.fit, x[test,], ncomp=1)

mean((pcr.pred - y[test])^2) #18.78685 Test MSE

######Problem 3, Part b#########

#Ridge Regression

set.seed(1)

train = sample(c(TRUE ,FALSE), nrow(p3),rep=TRUE)

test = (!train)

x <- model.matrix(G3~., data=p3)[,-1]

y <- p3$G3

#Ridge Regression

ridge.mod = glmnet(x[train,], y[train], alpha=0)

set.seed(1)

cv.out = cv.glmnet(x[train,], y[train], alpha=0, nfolds=5)

bestlam=cv.out$lambda.min

bestlam

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

mean((ridge.pred-y[test])^2) #4.392806 Test MSE

#Lasso Regression

set.seed(1)

lasso.mod = glmnet(x[train,], y[train], alpha=1)

set.seed(1)

cv.out = cv.glmnet(x[train,], y[train], alpha=1, nfolds=5)

bestlam=cv.out$lambda.min

bestlam

lasso.pred = predict(lasso.mod, s=bestlam, newx=x[test,])

mean((lasso.pred-y[test])^2) #3.466583 Test MSE

lasso.mod1 = glmnet(x[train,], y[train], alpha=1, lambda=bestlam)

coefficients(lasso.mod1)

#PC Regression

p3a <- as.data.frame(cbind(y,x))

names(p3a)[1] <- "G3"

set.seed(1)

pcr.fit = pcr(G3~., data=p3a, scale=TRUE, validation="CV", segments=5)

validationplot(pcr.fit, val.type="MSEP") #Levels out at 39 comps, 2.061

summary(pcr.fit)

2.077^2 #4.313929 CV Test MSE

pcr.pred=predict(pcr.fit, x[test,], ncomp=39)

mean((pcr.pred - y[test])^2) #3.265469 Test MSE

#PLS Regression

p3a <- as.data.frame(cbind(y,x))

names(p3a)[1] <- "G3"

set.seed(1)

pls.fit = plsr(G3~., data=p3a, scale=TRUE, validation="CV", segments=5)

validationplot(pls.fit, val.type="MSEP") #5 Comp levels out 2.0...

summary(pls.fit)

2.090^2 #4.3681 Test MSE

pls.pred=predict(pls.fit, x[test,], ncomp=5)

mean((pls.pred - y[test])^2) #Test MSE