Computational Statistics II

Lab 6 Homework

# Suraj Rimal (856489906)

1. (Question 9, Page 263) In this exercise, we will predict the number of applications received using the other variables in the College data set.
2. Split the data set into a training set and a test set.

***library(ISLR)***

***set.seed(2)***

***train = sample(c(TRUE, FALSE), nrow(College), rep=TRUE)***

***test = (!train)***

1. Fit a linear model using least squares on the training set and report the test error obtained.

***library(leaps)***

***regfit.best = regsubsets(Apps~., data=College[train,], nvmax = 18)***

***test.mat = model.matrix(Apps~., data = College[test,])***

***# Vector to store errors for different models***

***val.errors = rep(NA, 18)***

***for(i in 1:17){***

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

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

***val.errors[i] = mean((College$Apps[test]-pred)^2)***

***}***

***# Determining the minimum value for errors in vector***

***min\_error = which.min(val.errors)***

***cat("Error using linear regression is ", val.errors[min\_error])***



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

***library(glmnet)***

***x = model.matrix(Apps~., College)[,-1]***

***y = College$Apps***

***# Getting the best lambda using cross validation***

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

***best\_lambda = cv.out$lambda.min***

***grid=10^seq(10,-2, length =50)***

***ridge.mod = glmnet(x[train,], y[train], alpha = 0, lambda = best\_lambda)***

***ridge.pred = predict(ridge.mod, s= best\_lambda, newx = x[test,])***

***error = mean((ridge.pred - y[test])^2)***

***cat("Error using ridge regression is", error)***



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.

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

***best\_lambda = cv.out$lambda.min***

***lasso.mod = glmnet(x[train,], y[train], alpha = 1, lambda = best\_lambda)***

***out = glmnet(x, y, alpha = 1, lambda = best\_lambda)***

***lasso.coef = predict(out, type="coefficients", s= best\_lambda)[1:18,]***

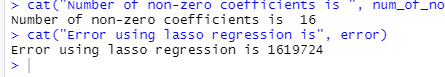
***num\_of\_non\_zero\_coefficients = length(lasso.coef[lasso.coef != 0])***

***lasso.pred = predict(lasso.mod, s=best\_lambda, newx=x[test,])***

***error = mean((lasso.pred-y[test])^2)***

***cat("Number of non-zero coefficients is ", num\_of\_non\_zero\_coefficients)***

***cat("Error using lasso regression is", error)***



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.

***library(pls)***

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

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

***# We see that the lowest cross validation error occurs when M = 17***

***pcr.pred = predict(pcr.fit, x[test,], ncomp = 17)***

***error = mean ((pcr.pred-y[test])^2)***

***cat("Error using pcr is", error)***

***Chart, line chart

Description automatically generated***



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.

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

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

***# We see that the lowest cross validation error occurs when M = 17***

***pls.pred = predict(pls.fit, x[test,], ncomp = 17)***

***error = mean((pls.pred-y[test])^2)***

***cat("Error using pcr is", error)***

***Chart

Description automatically generated with medium confidence***



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?

|  |  |
| --- | --- |
| **Regression Method** | **Mean Squared Error** |
| Linear Regression | 1556804 |
| Ridge Regression | 2455040 |
| Lasso Regression | 1619724 |
| PCR | 1556804 |
| PLS | 1556804 |

We can see that the Linear regression, PCR and PLS have the same accuracy, giving the test result of 1556804. Ridge regression appears to be the last on the list giving the highest amount of test error.