Problem Set 7, Winter 2022

Wendy Christensen

# Load any packages, if any, that you use as part of your answers here  
# For example:   
  
library(mlbench)

## Warning: package 'mlbench' was built under R version 4.1.2

library(glmnet)

## Warning: package 'glmnet' was built under R version 4.1.2

## Loading required package: Matrix

## Loaded glmnet 4.1-3

CONTEXT - HOUSE VALUES IN BOSTON, CIRCA 1970

This dataset was obtained through the mlbench package, which contains a subset of data sets available through the UCI Machine Learning Repository. From the help file:

Housing data for 506 census tracts of Boston from the 1970 census. The dataframe BostonHousing contains the original data by Harrison and Rubinfeld (1979).

The original data are 506 observations on 14 variables.

Continuous variables:

crim per capita crime rate by town zn proportion of residential land zoned for lots over 25,000 sq.ft  
indus proportion of non-retail business acres per town nox nitric oxides concentration (parts per 10 million) rm average number of rooms per dwelling age proportion of owner-occupied units built prior to 1940 dis weighted distances to five Boston employment centres rad index of accessibility to radial highways tax full-value property-tax rate per USD 10,000 ptratio pupil-teacher ratio by town b 1000(B - 0.63)^2 where B is the proportion of blacks by town lstat percentage of lower status of the population medv median value of owner-occupied homes in USD 1000’s

Categorical variables:

chas Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)

## Question 1 - 10 points

The BostonHousing data is contained inside of an R package, so you’ll load the data into memory a little differently than usual. Run the following code chunk, confirm that the data in loaded into memory, and ensure that your variables are of the proper type (they should be)

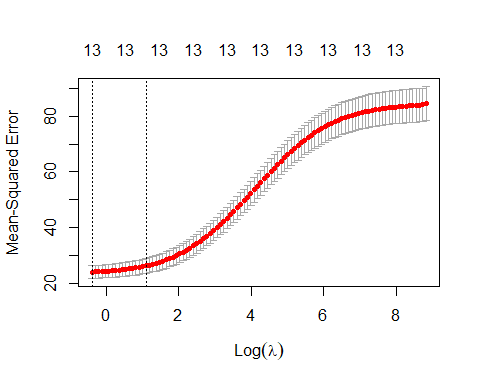
data(BostonHousing) # loads the BostonHousing dataset into memory from the mlbench package  
  
str(BostonHousing)

## 'data.frame': 506 obs. of 14 variables:  
## $ crim : num 0.00632 0.02731 0.02729 0.03237 0.06905 ...  
## $ zn : num 18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...  
## $ indus : num 2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...  
## $ chas : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...  
## $ nox : num 0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...  
## $ rm : num 6.58 6.42 7.18 7 7.15 ...  
## $ age : num 65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...  
## $ dis : num 4.09 4.97 4.97 6.06 6.06 ...  
## $ rad : num 1 2 2 3 3 3 5 5 5 5 ...  
## $ tax : num 296 242 242 222 222 222 311 311 311 311 ...  
## $ ptratio: num 15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...  
## $ b : num 397 397 393 395 397 ...  
## $ lstat : num 4.98 9.14 4.03 2.94 5.33 ...  
## $ medv : num 24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...

For this question, conduct a cross-validated ridge regression. Use medv as the outcome and all of the other variables in the data set as the predictors. Do NOT split your data into training and test sets for this question; conduct the analysis on the whole data set.

First, conduct the cross-validated ridge regression. Be sure to use the set.seed() provided to make your analysis reproducible, and also be sure to run it right before you run the line containing cvfit.house.ridge.

# Your code to get the data into the proper form to conduct cross-validated ridge regression  
  
X<-model.matrix(medv~.,data=BostonHousing)  
X<-X[,-1]  
y<-BostonHousing$medv  
  
# Your code to conduct cross-validated ridge regression  
  
set.seed(1000)  
  
cvfit.house.ridge <- cv.glmnet(x=X, y=y,alpha=0)# Complete this line  
  
plot(cvfit.house.ridge)



Next, display the value for lambda.min *and* the coefficients associated with it. Make sure these are visible in your knitted document.

# Display your lambda.min here  
cvfit.house.ridge$lambda.min

## [1] 0.6777654

# Display the coefficients associated with lambda.min here  
coef(cvfit.house.ridge, s = "lambda.min")

## 14 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) 28.001475824  
## crim -0.087572712  
## zn 0.032681030  
## indus -0.038003639  
## chas1 2.899781645  
## nox -11.913360479  
## rm 4.011308385  
## age -0.003731470  
## dis -1.118874607  
## rad 0.153730052  
## tax -0.005751054  
## ptratio -0.854984614  
## b 0.009073740  
## lstat -0.472423800

Finally, display the value for lambda.1se *and* the coefficients associated with it. Again, make sure these are visible in your knitted document.

# Display your lambda.1se here  
cvfit.house.ridge$lambda.1se

## [1] 3.002922

# Display the coefficients associated with lambda.1se here  
coef(cvfit.house.ridge, s = "lambda.1se")

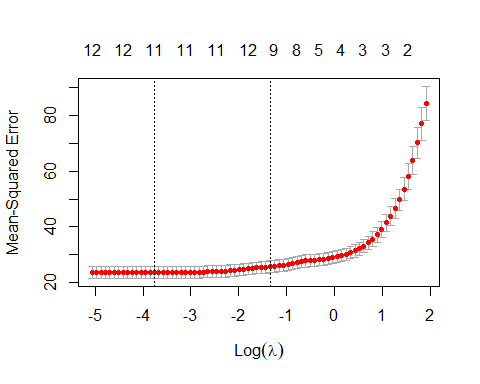
## 14 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) 21.360545973  
## crim -0.070614787  
## zn 0.021786752  
## indus -0.065928119  
## chas1 2.842394465  
## nox -6.204013827  
## rm 3.777295542  
## age -0.007121277  
## dis -0.619584776  
## rad 0.048495418  
## tax -0.003063933  
## ptratio -0.708289190  
## b 0.007996469  
## lstat -0.374059083

## Question 2 - 10 points

For this question, you will use the same outcome (medv) and the same predictors in the as in the last question, but you will instead conduct a cross-validated lasso regression. Do NOT split your data into training and test sets for this question; conduct the analysis on the whole data set.

First, conduct the cross-validated lasso regression. Be sure to use the set.seed() provided to make your analysis reproducible, and also be sure to run it right before you run the line containing cvfit.house.lasso

# Your code to get the data into the proper form to conduct cross-validated lasso regression  
X<-model.matrix(medv~.,data=BostonHousing)  
X<-X[,-1]  
y<-BostonHousing$medv  
  
  
# Your code to conduct cross-validated lasso regression  
  
set.seed(1000)  
  
cvfit.house.lasso <- cv.glmnet(x=X, y=y,alpha=1)# Complete this line  
  
plot(cvfit.house.lasso)



Next, display the value for lambda.min and the coefficients associated with it. Make sure these are visible in your knitted document.

# Display your lambda.min here  
cvfit.house.lasso$lambda.min

## [1] 0.02325053

# Display the coefficients associated with lambda.min here  
coef(cvfit.house.lasso, s = "lambda.min")

## 14 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) 34.741082558  
## crim -0.099983351  
## zn 0.042163911  
## indus .   
## chas1 2.691362171  
## nox -16.484121381  
## rm 3.856387745  
## age .   
## dis -1.412063666  
## rad 0.260319556  
## tax -0.010144690  
## ptratio -0.932724941  
## b 0.009070332  
## lstat -0.522516834

Finally, display the value for lambda.1se and the coefficients associated with it. Again, make sure these are visible in your knitted document.

# Display your lambda.1se here  
cvfit.house.lasso$lambda.1se

## [1] 0.2611788

# Display the coefficients associated with lambda.1se here  
coef(cvfit.house.lasso, s = "lambda.1se")

## 14 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) 21.248597773  
## crim -0.032386551  
## zn 0.008396934  
## indus .   
## chas1 2.244958389  
## nox -7.332177472  
## rm 4.251522556  
## age .   
## dis -0.631213121  
## rad .   
## tax .   
## ptratio -0.815597188  
## b 0.007076584  
## lstat -0.519864563

## Question 3 - 5 points

An important difference between ridge regression and lasso regression is that predictors can be dropped from a model in lasso but not in ridge. The number of predictors set to zero (if any) in lasso depends on the extent of the coefficient shrinkage at a given lambda. Answer the two questions below about the results of your cross-validated lasso models.

1. Among the set of coefficients associated with *lambda.min* in the cross-validated lasso regression, which predictors were set to zero? Please list them.

Your answer here: indus, age

1. Among the set of coefficients associated with *lambda.1se* in the cross-validated lasso regression, which predictors were set to zero? Please list them.

Your answer here: indus, age, rad, tax

1. Which of these - lambda.min or lambda.1se - had more coefficients set to zero?

Your answer here: lambda.lse

CONTEXT - NATIONAL INDONESIA CONTRACEPTIVE PRELAVENCE SURVEY (1987)

This dataset was obtained from the UCI Machine Learning Repository. From the description on <https://archive.ics.uci.edu/ml/datasets/Contraceptive+Method+Choice>:

This dataset is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The samples are married women who were either not pregnant or do not know if they were at the time of interview. The problem is to predict the current contraceptive method choice (no use, long-term methods, or short-term methods) of a woman based on her demographic and socio-economic characteristics.

Continuous variables

Wife’s age (w.age) Number of children ever born (children)

Categorical variables:

Wife’s education (w.edu) 1=low, 2, 3, 4=high Husband’s education (h.edu) 1=low, 2, 3, 4=high Wife’s religion (w.relig) 0=Non-Islam, 1=Islam Wife’s now working? (w.work) 0=Yes, 1=No Husband’s occupation (h.occ) 1, 2, 3, 4 Standard-of-living index (sol.index) 1=low, 2, 3, 4=high Media exposure (media) 0=Good, 1=Not good Contraceptive method used (contra) 1=No-use, 2=Long-term, 3=Short-term

## Question 4

First, load the data set into memory and change variables into their proper type.

contraception <- read.csv("contra.csv", header=TRUE, sep=",")  
  
str(contraception)

## 'data.frame': 1473 obs. of 10 variables:  
## $ w.age : int 24 45 43 42 36 19 38 21 27 45 ...  
## $ w.edu : int 2 1 2 3 3 4 2 3 2 1 ...  
## $ h.edu : int 3 3 3 2 3 4 3 3 3 1 ...  
## $ children : int 3 10 7 9 8 0 6 1 3 8 ...  
## $ w.relig : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ w.work : int 1 1 1 1 1 1 1 0 1 1 ...  
## $ h.occ : int 2 3 3 3 3 3 3 3 3 2 ...  
## $ sol.index: int 3 4 4 3 2 3 2 2 4 2 ...  
## $ media : int 0 0 0 0 0 0 0 0 0 1 ...  
## $ contra : int 1 1 1 1 1 1 1 1 1 1 ...

Next, re-code the contra variable such that non-use of contraception (contra=1) is equal to zero and use of any contraception (contra=2 or 3) is equal to one.

contraception$contra\_use <- contraception$contra # Complete this line  
contraception$contra\_use[contraception$contra==1] <- 0  
contraception$contra\_use[contraception$contra==3] <- 1  
contraception$contra\_use[contraception$contra==2] <- 1

Now, run the code chunk below to use the table() function to verify that your recoded variable is correct and answer the question below it.

table(contraception$contra)

##   
## 1 2 3   
## 629 333 511

table(contraception$contra\_use)

##   
## 0 1   
## 629 844

1. Does the number of 2’s and 3’s in the original contra variable match the number of 1’s in your recoded outcome variable? (Hint: if not, you have a mistake in your recoding)

Your answer here (yes/no): Yes

One more thing to do: your variables probably all came in as integer, but some of them are categorical variables (included your recoded outcome). Change your variables to the appropriate type

# Code to change variable types here  
contraception$w.age<- as.numeric(contraception$w.age)  
contraception$children<- as.numeric(contraception$children)  
contraception$w.edu<- as.factor(contraception$w.edu)  
contraception$h.edu<- as.factor(contraception$h.edu)  
contraception$w.relig<- as.factor(contraception$w.relig)  
contraception$w.work<- as.factor(contraception$w.work)  
contraception$h.occ<- as.factor(contraception$h.occ)  
contraception$sol.index<- as.factor(contraception$sol.index)  
contraception$media<- as.factor(contraception$media)  
contraception$contra<- as.factor(contraception$contra)   
contraception$contra\_use<- as.numeric(contraception$contra\_use)  
  
# Display your data using the str() function to demonstrate that variables have been recoded correctly  
str(contraception)

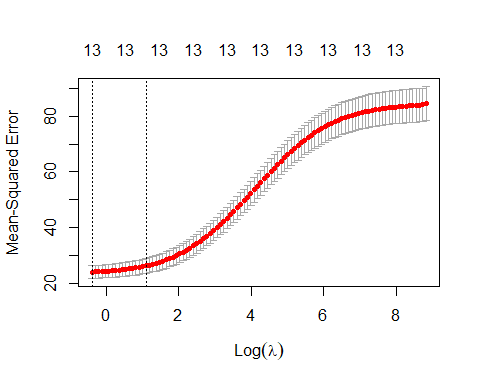
## 'data.frame': 1473 obs. of 11 variables:  
## $ w.age : num 24 45 43 42 36 19 38 21 27 45 ...  
## $ w.edu : Factor w/ 4 levels "1","2","3","4": 2 1 2 3 3 4 2 3 2 1 ...  
## $ h.edu : Factor w/ 4 levels "1","2","3","4": 3 3 3 2 3 4 3 3 3 1 ...  
## $ children : num 3 10 7 9 8 0 6 1 3 8 ...  
## $ w.relig : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...  
## $ w.work : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 1 2 2 ...  
## $ h.occ : Factor w/ 4 levels "1","2","3","4": 2 3 3 3 3 3 3 3 3 2 ...  
## $ sol.index : Factor w/ 4 levels "1","2","3","4": 3 4 4 3 2 3 2 2 4 2 ...  
## $ media : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...  
## $ contra : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...  
## $ contra\_use: num 0 0 0 0 0 0 0 0 0 0 ...

Now that you’ve recoded your outcome variable, you can now move on to the analysis!

Conduct a cross-validated ridge regression. Use your recoded contraception use variable as the outcome and all of the other variables (except contra) in the data set as the predictors. Do NOT split your data into training and test sets for this question; conduct the analysis on the whole data set.

First, conduct the cross-validated ridge regression. Be sure to use the set.seed() provided to make your analysis reproducible, and also be sure to run it right before you run the line containing cvfit.contra.ridge

# Your code to get the data into the proper form to conduct cross-validated ridge regression  
a<-model.matrix(contra\_use~w.age + w.edu + h.edu + children + w.relig + w.work + h.occ + sol.index + media,data=contraception)  
a <- a[,-1]  
b<-contraception$contra\_use  
  
  
# Your code to conduct cross-validated ridge regression  
  
set.seed(1000)  
  
cvfit.contra.ridge<- cv.glmnet(x=a, y=b,alpha=0)# Complete this line  
  
plot(cvfit.house.ridge)



Next, display the value for lambda.min and the coefficients associated with it. Make sure these are visible in your knitted document.

# Display your lambda.min here  
cvfit.contra.ridge$lambda.min

## [1] 0.009928821

# Display the coefficients associated with lambda.min here  
coef(cvfit.contra.ridge, s = "lambda.min")

## 18 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) 0.63954542  
## w.age -0.01639015  
## w.edu2 0.03235201  
## w.edu3 0.12767779  
## w.edu4 0.28290306  
## h.edu2 0.04599988  
## h.edu3 0.07525462  
## h.edu4 0.04503636  
## children 0.06574262  
## w.relig1 -0.09156743  
## w.work1 0.02885199  
## h.occ2 -0.03920432  
## h.occ3 0.01291358  
## h.occ4 0.08235210  
## sol.index2 0.07259440  
## sol.index3 0.11103510  
## sol.index4 0.15841830  
## media1 -0.11872671

Finally, display the value for lambda.1se and the coefficients associated with it. Again, make sure these are visible in your knitted document.

# Display your lambda.1se here  
cvfit.contra.ridge$lambda.1se

## [1] 0.1115328

# Display the coefficients associated with lambda.1se here  
coef(cvfit.contra.ridge, s = "lambda.1se")

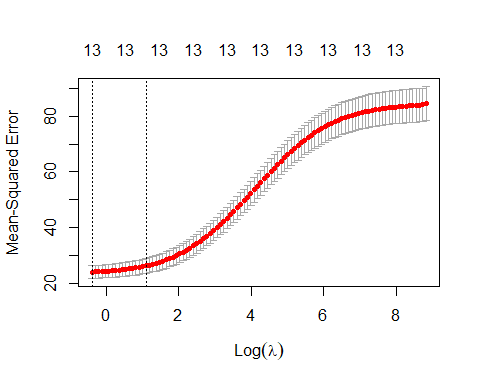
## 18 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) 0.684569142  
## w.age -0.010921010  
## w.edu2 -0.028040613  
## w.edu3 0.046940849  
## w.edu4 0.164963194  
## h.edu2 -0.005882661  
## h.edu3 0.037123532  
## h.edu4 0.034301316  
## children 0.044524603  
## w.relig1 -0.069667136  
## w.work1 0.029029619  
## h.occ2 -0.035184619  
## h.occ3 0.006866822  
## h.occ4 0.036732703  
## sol.index2 0.009382919  
## sol.index3 0.042187505  
## sol.index4 0.083778043  
## media1 -0.139463570

## Question 5

For this question, you will use the same outcome (your binary contraception use variable) and the same predictors in the as in the last question, but you will instead conduct a cross-validated lasso regression. Do NOT split your data into training and test sets for this question; conduct the analysis on the whole data set.

First, conduct the cross-validated lasso regression and display the output the function produces. Be sure to use the set.seed() provided to make your analysis reproducible, and also be sure to run it right before you run the line containing cvfit.contra.lasso

# Your code to get the data into the proper form to conduct cross-validated lasso regression  
a<-model.matrix(contra\_use~w.age + w.edu + h.edu + children + w.relig + w.work + h.occ + sol.index + media,data=contraception)  
a <- a[,-1]  
b<-contraception$contra\_use  
  
  
# Your code to conduct cross-validated lasso regression  
  
set.seed(1000)  
  
cvfit.contra.lasso <- cv.glmnet(x=a, y=b,alpha=1)# Complete this line  
  
plot(cvfit.house.ridge)



Next, display the value for lambda.min and the coefficients associated with it. Make sure these are visible in your knitted document.

# Display your lambda.min here  
cvfit.contra.lasso$lambda.min

## [1] 0.002894304

# Display the coefficients associated with lambda.min here  
coef(cvfit.contra.lasso, s = "lambda.min")

## 18 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) 0.738903757  
## w.age -0.016615157  
## w.edu2 0.012760956  
## w.edu3 0.111542706  
## w.edu4 0.268478216  
## h.edu2 .   
## h.edu3 0.027689679  
## h.edu4 .   
## children 0.065805940  
## w.relig1 -0.086598641  
## w.work1 0.021566586  
## h.occ2 -0.040380309  
## h.occ3 0.003337904  
## h.occ4 0.047549239  
## sol.index2 0.040162926  
## sol.index3 0.081394953  
## sol.index4 0.129988662  
## media1 -0.123023453

Finally, display the value for lambda.1se and the coefficients associated with it. Again, make sure these are visible in your knitted document.

# Display your lambda.1se here  
cvfit.contra.lasso$lambda.1se

## [1] 0.0186048

# Display the coefficients associated with lambda.1se here  
coef(cvfit.contra.lasso, s = "lambda.1se")

## 18 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) 0.718084491  
## w.age -0.011336056  
## w.edu2 .   
## w.edu3 0.044909486  
## w.edu4 0.199155087  
## h.edu2 .   
## h.edu3 .   
## h.edu4 .   
## children 0.046943792  
## w.relig1 -0.033627375  
## w.work1 .   
## h.occ2 -0.006558213  
## h.occ3 .   
## h.occ4 .   
## sol.index2 .   
## sol.index3 .   
## sol.index4 0.042414772  
## media1 -0.122434257

## Question 6 - 5 points

Answer the two questions below about the results of your cross-validated lasso models for the contraception data.

1. Among the set of coefficients associated with lambda.min in the cross-validated lasso regression, which predictors were set to zero? Please list them.

Your answer here: h.edu2, h.edu4

1. Among the set of coefficients associated with lambda.1se in the cross-validated lasso regression, which predictors were set to zero? Please list them.

Your answer here: w.edu2, h.edu2, h.edu3, h.edu4, w.work1, h.occ3, h.occ4, sol.index2, sol.index3

1. Which of these - lambda.min or lambda.1se - had more coefficients set to zero?

Your answer here: lambda.lse