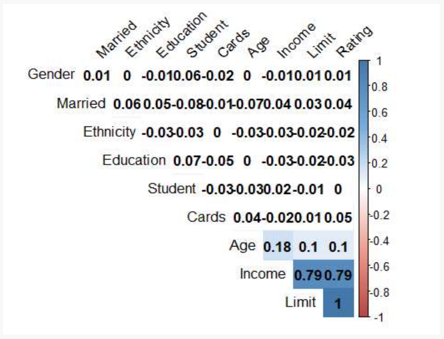
Homework 4

Juan Acosta

November 15, 2018

a) Looking at the file credit.csv we can see that accounting for the dummy variables of Ethnicity we have 11 predictors.

Running a quick correlation matrix we can see that there is clearly some correlation between Age, Income, Limit, and Rating. Therefore, the number of parameters, these correlations, and the amount of interaction terms will make LASSO viable in order to correct for any overparameterization.



```
library(corrplot)

## corrplot 0.84 loaded

cor.mtest <- function(mat, ...) {
  mat <- as.matrix(mat)
  n <- ncol(mat)
  p.mat<- matrix(NA, n, n)
  diag(p.mat) <- 0
  for (i in 1:(n - 1)) {
    for (j in (i + 1):n) {
      tmp <- cor.test(mat[, i], mat[, j], ...)
      p.mat[i, j] <- p.mat[j, i] <- tmp$p.value
    }
  }
  colnames(p.mat) <- rownames(p.mat) <- colnames(mat)
  p.mat
}

data <- read.csv('credit.csv')
df <- subset(data, select = -c(X, Gender, Student, Married, Ethnicity, Balance))
df$Gender <- (data$Gender=="Female")*1
df$Student <- (data$Student=="Yes")*1
df$Married <- (data$Married=="Yes")*1
df$Ethnicity <- as.numeric(data$Ethnicity)
M <- cor(df)
p.mat <- cor.mtest(df)
col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))
corrplot(M, method="color", col=col(200),
         type="upper", order="hclust",
         addCoef.col = "black", # Add coefficient of correlation
         tl.col="black", tl.srt=45, #Text label color and rotation
         # Combine with significance
         p.mat = p.mat, sig.level = 0.05, insig = "blank",
         # hide correlation coefficient on the principal diagonal
         diag=FALSE
)
```

b) We can see that almost all of the coefficients have gone to zero (all except rating) which is consistent with our statement on part a, meaning that the correlations between the predictors and overparameterization is being heavily penalized by the LASSO.

c) MSE = mean_square_error(y, fitted_y) = 0.5041427694026828

```
Index(['Income', 'Limit', 'Rating', 'Cards', 'Age', 'Education', 'Gender',
       'Student', 'Married', 'Asian', 'African American'],
      dtype='object')
```

```
Coefficients: [ 0.          0.          0.36237203  0.         -0.          0.
 -0.          0.         -0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
 -0.         -0.         -0.          0.          0.         -0.
  0.         -0.         -0.         -0.         -0.         -0.
  0.         -0.          0.          0.          0.          0.
  0.         -0.          0.         -0.         -0.          0.
  0.          0.          0.         -0.          0.        ]
Intercept: [-2.18956476e-18]
Fitted_y mean: 4.440892098500626e-18
MSE: 0.5041427694026828
```

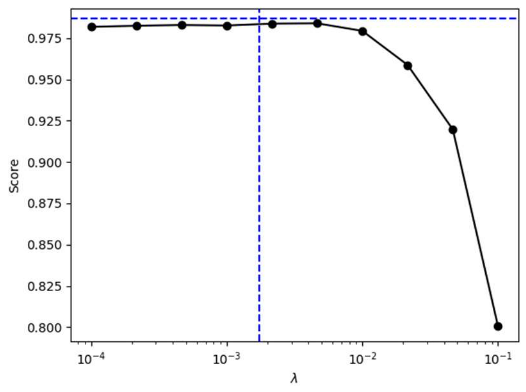
d)Running a 5-fold CV with lambda = 0.5 we can see that most coefficients disappear and we have pretty high MSE's with the 5th fold having the smallest at 0.398 yet the avg MSE = 0.50414276940268285 which is no different than the original MSE

```
Fold 1, Coefficients:
[ 0.          0.          0.37343086  0.         -0.          0.
 -0.          0.         -0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
 -0.         -0.          0.          0.          0.          0.
  0.         -0.         -0.         -0.         -0.         -0.
  0.         -0.          0.          0.          0.          0.
  0.         -0.          0.         -0.         -0.          0.
  0.          0.          0.         -0.          0.        ]
Intercept: [-0.01468451], MSE: 0.4722425120824699

Fold 2, Coefficients:
[ 0.          0.          0.34406554  0.         -0.         -0.
 -0.          0.         -0.         -0.         -0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
 -0.         -0.          0.          0.          0.          0.
  0.         -0.         -0.         -0.         -0.         -0.
  0.         -0.          0.         -0.          0.          0.
  0.         -0.          0.         -0.         -0.         -0.
  0.          0.          0.         -0.         -0.        ]
Intercept: [0.01302494], MSE: 0.6364941785309387

Fold 3, Coefficients:
[ 0.          0.35220304  0.00491833  0.         -0.          0.
 -0.          0.         -0.          0.         -0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.         -0.          0.          0.          0.
  0.          0.          0.          0.          0.         -0.
  0.          0.          0.          0.          0.          0.
 -0.         -0.          0.         -0.          0.         -0.
  0.          0.         -0.         -0.         -0.         -0.
  0.         -0.          0.          0.          0.          0.
  0.         -0.          0.         -0.         -0.          0.
  0.          0.          0.         -0.          0.        ]
Intercept: [-0.01372523], MSE: 0.5502034856879371
```

```
Fold 4, Coefficients:
[ 0.          0.          0.37449949  0.         -0.          0.
 -0.          0.          0.          0.         -0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
 -0.         -0.         -0.         -0.          0.         -0.
  0.         -0.         -0.         -0.         -0.         -0.
  0.         -0.          0.         -0.          0.          0.
  0.         -0.          0.         -0.         -0.          0.
  0.          0.          0.         -0.          0.        ]
Intercept: [0.03725784], MSE: 0.46378729373580835

Fold 5, Coefficients:
[ 0.          0.          0.35871558  0.         -0.         -0.
 -0.          0.         -0.         -0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.         -0.          0.          0.          0.          0.
  0.          0.          0.          0.         -0.          0.
  0.          0.          0.          0.          0.          0.
 -0.         -0.         -0.          0.          0.         -0.
  0.         -0.         -0.         -0.         -0.         -0.
 -0.         -0.          0.         -0.          0.         -0.
  0.         -0.          0.         -0.         -0.          0.
  0.          0.          0.         -0.          0.        ]
Intercept: [-0.02198376], MSE: 0.3979863769762602
```

e) We can see how between the LASSO in (c) and the 5-fold LASSO CVf the CV Lasso in (d) the MSE of (c) and the avg MSE of (d) are the same. Nonetheless, we can see several folds with lower coefficient values and lower MSE. This is consistent with the model "simplification", as it approaches a line all the coefficients will tend to 0 and the model will be lost. It is important to see that only because Rating has a high relationship with Balance its MSE is lower than the original LASSO, yet some of the other folds have higher MSE's which is again consistent with the simplification of the model, but the line being generated is a "worse" fit.

f) Running my code (appended at the end) we can see that the optimal lambda is 0.002154434690031882 which is consistent with the graph below.



g) From the Graph we can see how the OLS is quite powerful in this example and has a high score, yet by using the 5-fold CV LASSO we can achieve a minor improvement in score by setting our lambda to be 0.002. This is consistent with the results from parts (c) and (d) and what we studied in class.

Code in python

```
import numpy as np  #general library, will come in handy later
import pandas as pd  #another nice library for storing matrices, it rely's on
numpy
import matplotlib.pyplot as plt  #this library is for graphing things
from sklearn.model_selection import train_test_split
from sklearn.linear_model import Lasso  #These libraries have the necessary
models
from sklearn.metrics import mean_squared_error
from sklearn.linear_model import LassoCV
#load data
raw_df = pd.read_csv('credit.csv')
output = pd.DataFrame(raw_df['Balance'])
output = (output - output.mean()) / output.std()
y = output.values
df = raw_df.drop(['Unnamed: 0', 'Balance', 'Ethnicity', 'Gender', 'Student',
'Married'], axis=1)
df = (df-df.mean())/df.std()

m = {'Male' : 1, 'Female' : 0}
df['Gender'] = raw_df['Gender'].map(m)
m = {'Yes' : 1, 'No' : 0}
df['Student'] = raw_df['Student'].map(m)
m = {'Yes' : 1, 'No' : 0}
df['Married'] = raw_df['Married'].map(m)
m = {'Asian' : 1, 'Caucasian' : 0, 'African American' : 0}
df['Asian'] = raw_df['Ethnicity'].map(m)
m = {'African American' : 1, 'Caucasian' : 0, 'Asian' : 0}
df['African American'] = raw_df['Ethnicity'].map(m)

#get the variable names in a list
column_names = df.columns  #select the columns we want
print(column_names)
df_extended = df.copy()  #make a copy to edit
count = 0
for i in range(len(column_names)-2):
  column = str(column_names[i])
  j = 0
  if count > 5:
    j=1
  for interact in range(i+j, len(column_names)):
    interact = str(column_names[interact])
    interaction_name = column+'*'+interact
    df_extended[interaction_name] = df_extended[column] * df_extended[interact]
  count = count + 1
print(df_extended.describe())
X = df_extended


#fit the lasso to it, notice the second parameter is 0
#why do you think that is?
lasso = Lasso(alpha=.5)  #note that alpha corresponds to lambda
lasso.fit(X, y)
print(lasso.coef_)
print(lasso.intercept_)

#prepare fitted data to compare using MSE function

fitted_y = lasso.predict(X)
print(fitted_y.mean())

MSE = mean_squared_error(y, fitted_y)

print(MSE)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,
random_state=0)

alphas = np.logspace(-4, -1, 10)
scores = np.empty_like(alphas)
for i,a in enumerate(alphas):
    lasso = Lasso()
    lasso.set_params(alpha=a)
    lasso.fit(X_train, y_train)
    scores[i] = lasso.score(X_test, y_test)
    print('Lambda: ',a,' Coefs: ', lasso.coef_)

lassocv = LassoCV(cv=5)
lassocv.fit(X, y)
lassocv_score = lassocv.score(X, y)
lassocv_alpha = lassocv.alpha_
print('CV', lassocv.coef_)

plt.plot(alphas, scores, '-ko')
plt.axhline(lassocv_score, color='b', ls='--')
plt.axvline(lassocv_alpha, color='b', ls='--')
plt.xlabel(r'$\lambda$')
plt.ylabel('Score')
plt.xscale('log')
plt.savefig('my_graph.png')
plt.clf()   # Clear figure

for i in range(5):
    #compute start/end of fold
    start_index = int((80)*i)
    end_index = int((80)*(i + 1))

    #partition data
    X_test = X[start_index:end_index]
    y_test = y[start_index:end_index]

    X_train = np.concatenate((X[0:start_index], X[end_index:]))
    y_train = np.concatenate((y[0:start_index], y[end_index:]))

    #estimate model
    l = Lasso(alpha=.5)
    l.fit(X_train, y_train)
    fitted_y = lasso.predict(X_test)
    MSE = mean_squared_error(y_test, fitted_y)
    print('Fold %s, Coefficients: %s, Intercept: %s, MSE: %s \n' % (i+1, l.coef_,
l.intercept_, MSE))


#from here you can figure out CV_n

#Specify a grid of lambda values to optimize over.
lambda_values = np.logspace(-4, -1, 10)

lass_cv = LassoCV(cv=5, alphas=lambda_values)
lass_cv = lass_cv.fit(X, y)

#The lass gets as close as possible (given our grid)
print('lass_cv.coef_ : ', lass_cv.coef_)
print('lass_cv.intercept_ : ',lass_cv.intercept_)
print('lass_cv.alpha_ : ' ,lass_cv.alpha_)
#iteratively fit model and create a graph
intercepts = []

print(lambda_values)
```

```
for lamb in lambda_values:
    l = Lasso(alpha=lamb)
    l.fit(X, y)
    intercepts.append(lasso.intercept_)
print(intercepts)

#graph result
plt.plot(lambda_values, intercepts)
plt.savefig('Class_graph.png')
```

Code in R (Failed due to memory error)

```
library(corrplot)

library(rJava)

library(xlsx)




cor.mtest <- function(mat, ...) {

  mat <- as.matrix(mat)

  n <- ncol(mat)

  p.mat<- matrix(NA, n, n)

  diag(p.mat) <- 0

  for (i in 1:(n - 1)) {

    for (j in (i + 1):n) {

      tmp <- cor.test(mat[, i], mat[, j], ...)

      p.mat[i, j] <- p.mat[j, i] <- tmp$p.value

    }

  }

  colnames(p.mat) <- rownames(p.mat) <- colnames(mat)

  p.mat

}
```

```
data <- read.csv('credit.csv')
bal <- data$Balance
df <- subset(data, select = -c(X, Gender, Student, Married, Ethnicity, Balance))
dfexp <- df**2
df$Gender <- (data$Gender=="Female")*1
df$Student <- (data$Student=="Yes")*1
df$Married <- (data$Married=="Yes")*1
df$Asian <- (data$Ethnicity=="Asian")*1
df$African <- (data$Ethnicity=="African American")*1

colnames(dfexp) <- c("Income*Income", "Limit*Limit", "Rating*Rating", "Cards*Cards",
           "Age*Age", "Education*Education")

dfinter <- do.call(cbind, combn(colnames(df), 2, FUN=function(x)
  list(setNames(data.frame(df[,x[1]]*df[,x[2]]), paste(x, collapse='*')))))

df1 <- merge(dfinter, dfexp)

write.xlsx(df1, "interactions.xlsx")

M <- cor(df)
p.mat <- cor.mtest(df)
col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))
corrplot(M, method="color", col=col(200),
     type="upper", order="hclust",
     addCoef.col = "black", # Add coefficient of correlation
     tl.col="black", tl.srt=45, #Text label color and rotation
     # Combine with significance
```

```
        p.mat = p.mat, sig.level = 0.05, insig = "blank",

        # hide correlation coefficient on the principal diagonal

        diag=FALSE

)
```

# Running a quick correlation matrix we can see that there is clearly some high correlation between Age, Income, Limit, and Rating.

# Therefore, LASSO could potentially be a beter model for these parameters as it would adjust for this correlation.

```
data <- read.csv('credit.csv')

summary(data)

summary(data.frame(data))

head(data)
```