

FTAP HW 7

Zachary Fogelson

July 21, 2015

Lasso Homework

Problem 1

1.

```
train <- read.csv("train.csv")
trainMeaningFull <-subset(train, select=-c(id, member_id))
trainMeaningFull <- trainMeaningFull[complete.cases(trainMeaningFull),]
pandoc.table(head(train))
```

```
##
## -----
##  id      member_id    funded_amnt    term_m    int_rate
## -----
##  54734      80364         25000         36      0.1189
##
##  55742     114426          7000         36      0.1071
##
##  57245     138150          1200         36      0.1311
##
##  57416     139635         10800         36      0.1357
##
##  58915     153417          5025         36      0.1008
##
##  59006     154254          3000         36      0.1426
## -----
##
## Table: Table continues below
##
## -----
##  installment    emp_length    annual_inc    dti    revol_util
## -----
##      829.1          0         85000    19.48     0.521
##
##      228.2          0         65000    14.29     0.767
##
##      40.5         15         54000     5.47     0.404
##
##      366.9          6         32000    11.63     0.256
##
##      162.3          3         85000     8.1      0.732
##
##      102.9          3         80800    14.97     0.395
## -----
```

```
pandoc.table(rbind(summary(train), sapply(train, sd)))
```

```
##
## -----
##      id      member_id      funded_amnt      term_m
## -----
## Min.   : 54734    Min.   : 70699    Min.   : 500    Min.   :36.00
## 1st Qu.:496560    1st Qu.: 635543    1st Qu.: 5000    1st Qu.:36.00
## Median :623258    Median : 798198    Median : 9250    Median :36.00
## Mean   :626300    Mean   : 783617    Mean   :10654    Mean   :42.22
## 3rd Qu.:767345    3rd Qu.: 967828    3rd Qu.:15000    3rd Qu.:60.00
## Max.   :975902    Max.   :1198245    Max.   :35000    Max.   :60.00
##
##      NA      NA      NA      NA
##
## 170023.183351294 225109.217961916 6974.75731097762 10.5166250234748
## -----
##
## Table: Table continues below
##
## -----
##      int_rate      installment      emp_length      annual_inc
## -----
## Min.   :0.0542    Min.   : 15.69    Min.   : 0.000    Min.   : 4000
## 1st Qu.:0.0925    1st Qu.: 163.67    1st Qu.: 2.000    1st Qu.: 40002
## Median :0.1171    Median : 272.80    Median : 4.000    Median : 59000
## Mean   :0.1191    Mean   : 317.15    Mean   : 6.044    Mean   : 69152
## 3rd Qu.:0.1435    3rd Qu.: 417.36    3rd Qu.: 9.000    3rd Qu.: 82500
## Max.   :0.2459    Max.   :1305.19    Max.   :15.000    Max.   :6000000
##
##      NA      NA      NA's :868      NA
##
## 0.0362426792330949 205.332202918801      NA      66645.1319511594
## -----
##
## Table: Table continues below
##
## -----
##      dti      revol_util
## -----
## Min.   : 0.00    Min.   :0.0000
```

```
##
## 1st Qu.: 8.00    1st Qu.:0.2420
##
## Median :13.28   Median :0.4760
##
## Mean :13.18     Mean :0.4774
##
## 3rd Qu.:18.49   3rd Qu.:0.7120
##
## Max. :29.99     Max. :0.9990
##
##      NA      NA's :44
##
## 6.69912095939332      NA
## -----
```

2.

a)

```
ols <- lm(int_rate ~ ., trainMeaningFull)
olsSum <- summary(ols)
olsSum
```

```
##
## Call:
## lm(formula = int_rate ~ ., data = trainMeaningFull)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.103692 -0.017971 -0.001579  0.017266  0.118674
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.761e-02  1.017e-03 -27.139  < 2e-16 ***
## funded_amnt -9.413e-06  1.261e-07 -74.653  < 2e-16 ***
## term_m      2.914e-03  2.455e-05 118.687  < 2e-16 ***
## installment 3.411e-04  4.087e-06  83.461  < 2e-16 ***
## emp_length  -3.513e-04  2.648e-05 -13.268  < 2e-16 ***
## annual_inc  -1.519e-08  2.172e-09  -6.993 2.75e-12 ***
## dti          -2.441e-04  2.194e-05 -11.126  < 2e-16 ***
## revol_util   4.657e-02  5.251e-04  88.676  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02525 on 33112 degrees of freedom
## Multiple R-squared:  0.512, Adjusted R-squared:  0.5119
## F-statistic: 4964 on 7 and 33112 DF, p-value: < 2.2e-16
```

Based on the OLS regression, all of the columns which are not IDs appear to be significant

b)

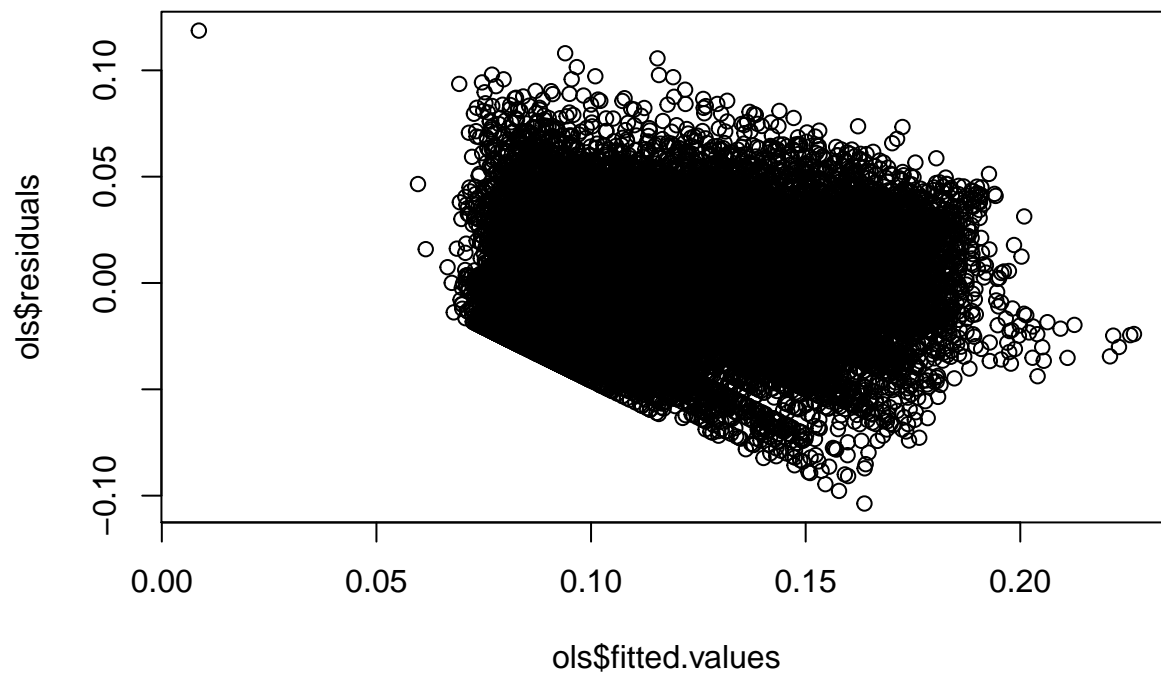
```
olsSum$r.squared
```

```
## [1] 0.5120423
```

The baseline model has an R^2 of about .51. It is hard to judge if the baseline model is successful because, we do not have anything to compare it to. But using all of the available information we can explain 39% of the variance which is not terrible.

c)

```
plot(ols$fitted.values, ols$residuals)
```



d)

```
rmse <- mean((ols$residuals)^2)  
rmse
```

```
## [1] 0.0006375506
```

e)

```
test <- read.csv("test.csv")  
test <- subset(test, select=c(id, member_id))  
test <- test[complete.cases(test),]
```

```
predOLS <- predict(ols, test, interval = "none")  
rmseOLS <- mean((test$int_rate - predOLS)^2)  
cat("RMSE OLS: ", rmseOLS)
```

```
## RMSE OLS: 0.0007511019
```

3.

```
interactDF <- as.data.frame(model.matrix(~(.)^2,subset(trainMeaningFull, select=-c(int_rate))))  
interactTest <- as.data.frame(model.matrix(~(.)^2,subset(test, select=-c(int_rate))))
```

a)

```
smallest = lm(trainMeaningFull$int_rate ~ 1, interactDF)  
biggest = as.formula(lm(trainMeaningFull$int_rate ~ ., interactDF))  
stepForwardAIC <- step(smallest, scope = biggest, k = 2, direction = "forward", trace = F)  
stepForwardBIC <- step(smallest, scope = biggest, k = log(length(trainMeaningFull$int_rate)), direction = "forward", trace = F)  
cat("R^2 AIC: ", summary(stepForwardAIC)$r.squared, "R^2 BIC: ", summary(stepForwardBIC)$r.squared)
```

```
## R^2 AIC: 0.8377169 R^2 BIC: 0.8377169
```

b)

```
cat("AIC: ", AIC(stepForwardAIC), "BIC: ", BIC(stepForwardBIC))
```

```
## AIC: -186107.8 BIC: -185872.3
```

c)

```
rmseAIC <- mean((stepForwardAIC$residuals)^2)  
rmseBIC <- mean((stepForwardBIC$residuals)^2)  
cat("RMSE AIC: ", rmseAIC, "RMSE BIC: ", rmseBIC)
```

```
## RMSE AIC: 0.0002120341 RMSE BIC: 0.0002120341
```

d)

```
predAIC <- predict(stepForwardAIC,interactTest, interval = "none")  
predBIC <- predict(stepForwardBIC,interactTest, interval = "none")  
rmseAIC <- mean((test$int_rate - predAIC)^2)  
rmseBIC <- mean((test$int_rate - predBIC)^2)  
cat("RMSE AIC: ", rmseAIC, "RMSE BIC: ", rmseBIC)
```

```
## RMSE AIC: 0.0002731841 RMSE BIC: 0.0002731841
```

4. a)

```
biggest = lm(trainMeaningFull$int_rate ~ ., interactDF)  
smallest = as.formula(lm(trainMeaningFull$int_rate ~ 1, interactDF))  
stepBackwardAIC <- step(biggest, scope = smallest, k = 2, direction = "backward", trace = F)  
stepBackwardBIC <- step(biggest, scope = smallest, k = log(length(trainMeaningFull$int_rate)), direction = "backward", trace = F)  
cat("R^2 AIC: ", summary(stepBackwardAIC)$r.squared, "R^2 BIC: ", summary(stepBackwardBIC)$r.squared)
```

```
## R^2 AIC: 0.8376995 R^2 BIC: 0.8376154
```

b)

```
cat("AIC: ", AIC(stepBackwardAIC), "BIC: ", BIC(stepBackwardBIC))
```

```
## AIC: -186108.2 BIC: -185893.3
```

c)

```
rmseBackAIC <- mean((stepBackwardAIC$residuals)^2)
rmseBackBIC <- mean((stepBackwardBIC$residuals)^2)
cat("RMSE AIC: ", rmseBackAIC, "RMSE BIC: ", rmseBackBIC)
```

```
## RMSE AIC: 0.0002120568 RMSE BIC: 0.0002121667
```

d)

```
predBackAIC <- predict(stepBackwardAIC, interactTest, interval = "none")
predBackBIC <- predict(stepBackwardBIC, interactTest, interval = "none")
rmseBackAIC <- mean((test$int_rate - predBackAIC)^2)
rmseBackBIC <- mean((test$int_rate - predBackBIC)^2)
cat("RMSE AIC: ", rmseBackAIC, "RMSE BIC: ", rmseBackBIC)
```

```
## RMSE AIC: 0.0002732995 RMSE BIC: 0.0002731435
```

5.

a)

```
model=as.formula(paste("~", paste(names(interactDF)[-1], collapse= "+")))
x=model.matrix(model,interactDF);
lassoFit <- glmnet(x,trainMeaningFull$int_rate)
outLasso <- predict(lassoFit, newx = as.matrix(interactDF), s=.0001)
cat("R^2: ", var(outLasso)/var(ols$residuals+ols$fitted.values))
```

```
## R^2: 0.6976909
```

```
coef(lassoFit, s=.0001)
```

```
## 30 x 1 sparse Matrix of class "dgCMatrix"
##              1
## (Intercept)  2.539734e-02
## (Intercept)  .
## funded_amnt -2.693844e-05
## term_m      1.572056e-03
## installment 3.912436e-04
## emp_length  -4.509901e-04
## annual_inc  -1.140216e-08
## dti         -1.850803e-04
## revol_util   5.564668e-02
## funded_amnt:term_m  .
## funded_amnt:installment -1.266817e-09
```

```
## funded_amnt:emp_length 1.304751e-08
## funded_amnt:annual_inc .
## funded_amnt:dti .
## funded_amnt:revol_util -4.823517e-06
## term_m:installment 1.392770e-05
## term_m:emp_length .
## term_m:annual_inc .
## term_m:dti .
## term_m:revol_util .
## installment:emp_length 1.015833e-07
## installment:annual_inc 7.992317e-12
## installment:dti 4.234663e-07
## installment:revol_util 6.463018e-05
## emp_length:annual_inc 5.289935e-10
## emp_length:dti .
## emp_length:revol_util 1.753555e-05
## annual_inc:dti -1.428128e-09
## annual_inc:revol_util .
## dti:revol_util .
```

The R^2 is calculated based on the R^2 calculation in `lasso_class`; however, prof. Russell did mention that there is no R^2 for the lasso, so I printed the coefficients of the lambda model instead.

b)

```
insamplePred <- predict(lassoFit, newx = as.matrix(interactDF), s=.0001)
lassoRMSE <- mean((insamplePred - trainMeaningFull$int_rate)^2)
cat("RMSE Lasso: ", lassoRMSE)
```

```
## RMSE Lasso: 0.0002951606
```

c)

```
predLasso <- predict(lassoFit, newx = as.matrix(interactTest), s=.0001)
lassoRMSE <- mean((test$int_rate - predLasso)^2)
cat("RMSE Lasso: ", lassoRMSE)
```

```
## RMSE Lasso: 0.0003651544
```

6.

a)

```
lassoCVFit <- cv.glmnet(x, trainMeaningFull$int_rate, nfolds = 10)
lassoCVFit$lambda.min
```

```
## [1] 1.462956e-05
```

b)

```
insampleCVPred <- predict(lassoCVFit, newx = as.matrix(interactDF), s="lambda.min")
lassoCVRMSE <- mean((insampleCVPred - trainMeaningFull$int_rate)^2)
cat("RMSE CVLasso: ", lassoCVRMSE)
```

```
## RMSE CVLasso: 0.0002278089
```

c)

```
predCVLasso <- predict(lassoCVFit, newx = as.matrix(interactTest), s="lambda.min")
lassoCVRMSE <- mean((test$int_rate - predCVLasso)^2)
cat("RMSE CVLasso: ", lassoCVRMSE)
```

```
## RMSE CVLasso: 0.0002958154
```

Logit Homework

```
trades <- read.csv("detailed_trades_est.csv")
```

1)

```
logit <- glm(PCHANGE~LASK+LBID+RETURN+SIGN_VOL, family=binomial, data=trades)
cat("Lasso AIC: ", AIC(logit), "Lasso BIC: ", BIC(logit))
```

```
## Lasso AIC: 3754.378 Lasso BIC: 3784.334
```

2)

```
interactions <- subset(trades, select = -c(PCHANGE, PCHANGE0))
interactions <- as.data.frame(model.matrix(~(.)^2, interactions))
```

3)

```
trades.pca <- prcomp(interactions)
myModel <- lm(trades$PCHANGE~trades.pca$x[,1]+trades.pca$x[,2]+trades.pca$x[,3])
```

4)

```
error <- trades$PCHANGE-(as.numeric(myModel$fitted.values > .5))
mean(abs(error))
```

```
## [1] 0.3431472
```