REgularization Methods (Ridge,Lasso and Elastic Net)

Vazgen Tadevosyan

March 26, 2019

```
For this Homework, you are required to submit both Markdown and HTML files with your answers and code i

Use movies_data.xls dataset uploaded on Moodle to analyze the relationship between the target variable a
```

Problem 1. 3 pt.

```
a. Fit a linear model using least squares on the training set, and report the test error obtained.
b. What is regularization? Why we need it?
```

```r
library(readxl)
library(dplyr)
library(plyr)
library(zoo)
library(Metrics)
```

a.

```r
data<-read_excel("movies_data.xls",na=c('#DIV/0',"#DIV/0!","#NAME?", 'NA',"N/A",""," "))
data$genre_first<-factor(data$genre_first)
data$Country<-factor(data$Country)
data$Rated<-factor(data$Rated)
data$Rated<-revalue(data$Rated, c("NOT RATED"="UNRATED"))
data$genre_first<-factor(data$genre_first)
data$Production<-factor(data$Production)
data$OscarWon<-factor(data$OscarWon)
data$Oscar_binary<-ifelse(data$OscarNom==0,0,1)
data<-data%>% mutate_if(is.numeric , na.aggregate)
data<-na.omit(data)
df<-data %>% dplyr::select(-c(Writer,Production,index,Country,OscarWon,Oscar_binary,OscarNom,Director,P
```

```r
set.seed(42)
index<-sample(nrow(df),nrow(df)*.75,replace = F)
train<-df[index,]
test<-df[-index,]
options(scipen=999)
model<-lm(gross_adjusted~genre_first+imdbRating+year+budget_adjusted+reviews,data = train)
summary(model)
```

```
##
## Call:
## lm(formula = gross_adjusted ~ genre_first + imdbRating + year +
##     budget_adjusted + reviews, data = train)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -244524427  -34691091   -8725934   20388302  422225578
##
## Coefficients:
##                                  Estimate       Std. Error t value
## (Intercept)             4826568298.23499  370718743.24039  13.019
```

```
## genre_firstAdventure     21822181.00362    6312194.09966   3.457
## genre_firstAnimation     40673007.38534   16207309.80680   2.510
## genre_firstBiography    -12378162.93018    8606042.14727  -1.438
## genre_firstComedy        19030341.54305    5056354.20495   3.764
## genre_firstCrime        -14400836.17116    7533400.01185  -1.912
## genre_firstDocumentary   -3514077.15475   22338336.41871  -0.157
## genre_firstDrama         -6399694.59752    5719735.47211  -1.119
## genre_firstFamily        -9893496.70593   65179082.52149  -0.152
## genre_firstFantasy        3426532.44251   16839974.69870   0.203
## genre_firstHorror        10480212.39278    8985334.28632   1.166
## genre_firstMusical     -151153093.00054   66531078.97807  -2.272
## genre_firstMystery       -8275722.11141   21026618.66668  -0.394
## genre_firstRomance       -6178375.80188   65169612.79549  -0.095
## genre_firstSci-Fi       -45454483.70235   46219658.85949  -0.983
## imdbRating               11358461.88757    2012005.43697   5.645
## year                     -2444153.43635     183550.00479 -13.316
## budget_adjusted                 0.90438          0.04429  20.421
## reviews                     39927.98760       4662.96754   8.563
##                                    Pr(>|t|)
## (Intercept)            < 0.0000000000000002 ***
## genre_firstAdventure               0.000561 ***
## genre_firstAnimation               0.012189 *
## genre_firstBiography               0.150546
## genre_firstComedy                  0.000174 ***
## genre_firstCrime                   0.056111 .
## genre_firstDocumentary             0.875020
## genre_firstDrama                   0.263364
## genre_firstFamily                  0.879373
## genre_firstFantasy                 0.838789
## genre_firstHorror                  0.243644
## genre_firstMusical                 0.023227 *
## genre_firstMystery                 0.693943
## genre_firstRomance                 0.924482
## genre_firstSci-Fi                  0.325541
## imdbRating                     0.0000000195 ***
## year                   < 0.0000000000000002 ***
## budget_adjusted        < 0.0000000000000002 ***
## reviews                < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 65030000 on 1561 degrees of freedom
## Multiple R-squared:  0.4311, Adjusted R-squared:  0.4246
## F-statistic: 65.72 on 18 and 1561 DF,  p-value: < 0.00000000000000022
```

```r
pred<-predict(model,test)
actual<-test$gross_adjusted
rmse(actual,pred)
```

```
## [1] 66139797
```

b. *Regularization is a technique that is used to avoid overfitting. It introduces small amount of bias into how new line is fit to the data.But in return for that we get a significant drop in Variance. Rmse for the test is 66139797*

Problem 2. 5 pt.

```
a. Fit a ridge regression model on the training set, with lambda chosen by cross-validation. Report the

b. Describe the main idea of ridge regression.
```

a.

```
library(glmnet)
```

```
## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-16

##
## Attaching package: 'glmnet'

## The following object is masked from 'package:Metrics':
##
##     auc
```

```
set.seed(42)
y_train<-train$gross_adjusted
x_train<-model.matrix(~.,subset(train,select =-gross_adjusted))
y_test<-test$gross_adjusted
x_test<-model.matrix(~.,subset(test,select =-gross_adjusted))
lambdas<-seq(from=0,to = 10,by = 0.001)
x <- model.matrix( ~ .-gross_adjusted, df)
cros_ridge<-cv.glmnet(x =x,y=df$gross_adjusted,alpha=0,lambda = lambdas,nfolds = 10)
ridge<-glmnet(x = x_train,y=y_train,lambda=cros_ridge$lambda.min,alpha = 0)
pred_r<-predict(ridge,newx = x_test )
rmse(y_test,pred_r)
```

```
## [1] 66139237
```

b.

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda*\sum_{j=1}^{k}\hat{\beta_j}^2=$$

$$=SSR+\lambda*\sum_{j=1}^{k}\hat{\beta_j}^2$$

Ridge regression is adding a penalty term to our loss function. $\lambda*\sum_{j=1}^{k}\hat{\beta_j}^2$ *is a shrinkage penalty. where*$\lambda\geq 0$*is a tuning parameter. This has the effect of shrinking large values of the coefficients towards zero.So our dependent variable becomes less sensitive to predictors.As* λ *increases the flexibility of the ridge regression fit decreases, leading to decreased variance but increased bias.Thus,we should get optimal value of* λ*. Rmse for the test is 66139237*

Problem 3. 5 pt.

```
a. Fit a lasso model on the training set, with lambda chosen by cross-validation. Report the test error

b. What is the difference between ridge and lasso regression?
```

```
lambdas<-seq(from=0,to = 10,by = 0.001)
cros_lasso<-cv.glmnet(x =x,y=df$gross_adjusted,alpha=1,lambda = lambdas,nfolds = 10)
lasso<-glmnet(x = x_train,y=y_train,lambda=cros_lasso$lambda.min,alpha = 1)
pred_l<-predict(lasso,newx = x_test )
rmse(y_test,pred_l)
```

```
## [1] 66139237
```

```r
sum(lasso$beta!=0)
```

```
## [1] 18
```

Number of nonzero coeficients are 18 for lasso regression. b.*Ridge regression's penalty term will never force any of the coefficients to be exactly zero. Thus the final model will include all variables, whick makes it harder to interpret. Meanwhile the LASSO make some coefficients end up being set to exactly zero. The LASSO works in a similar way as Ridge,except it uses a different penalty term.* $SSR + \lambda * \sum_{j=1}^{k} |\hat{\beta}_j|$ *where*$\lambda \geq 0$ * is a tuning parameter. So, the Lasso Regression can exclude uselss variables from equation. Rmse for the test is 66139237 *

Problem 4. 7 pt.

```
a. Find the best elastic net regression with alpha chosen by cross-validation.
b. Is there much difference among the test errors resulting from these four approaches. Which one is the
```

```r
alphas=seq(from=0,to=1,length.out = 11)^3
library(glmnetUtils)
```

```
## Warning: package 'glmnetUtils' was built under R version 3.5.3

##
## Attaching package: 'glmnetUtils'

## The following objects are masked from 'package:glmnet':
##
##     cv.glmnet, glmnet
```

```r
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```r
# make some dummy data
output.of.cva.glmnet <- cva.glmnet(x =x,y=df$gross_adjusted,alpha = alphas)
number.of.alphas.tested <- length(output.of.cva.glmnet$alpha)
cv.glmnet.dt <- data.table()
for (i in 1:number.of.alphas.tested){
  glmnet.model <- output.of.cva.glmnet$modlist[[i]]
  min.mse <-  min(glmnet.model$cvm)
  min.lambda <- glmnet.model$lambda.min
  alpha.value <- output.of.cva.glmnet$alpha[i]
  new.cv.glmnet.dt <- data.table(alpha=alpha.value,min_mse=min.mse,min_lambda=min.lambda)
  cv.glmnet.dt <- rbind(cv.glmnet.dt,new.cv.glmnet.dt)
}
best.params <- cv.glmnet.dt[which.min(cv.glmnet.dt$min_mse)]
elast<-glmnet(x = x_train,y=y_train,lambda=best.params$min_lambda,alpha = best.params$alpha)
pred_l<-predict(elast,newx = x_test)
rmse(y_test,pred_l)
```

```
## [1] 66124701
```

Rmse for the test is 66124701

There is not much differences among the test errors resulting from these four approaches, however the best model according to RMSE is elastic net with alpha=0.729 and lambda=0.008

Bonus 2 pt.

`Is there any relationship between the variance of ridge estimation and the variance of OLS estimation? S`

$Var(\hat{\beta}_{ols}) = \sigma^2(X^TX)^{-1}$ $Var(\hat{\beta}_{ridge}) = \sigma^2(X^TX+\lambda I)^{-1}X^TX(X^TX+\lambda I)^{-1}$ Note that $\lambda \geq 0$ From this equations we can state $Var(\hat{\beta}_{ridge}) \leq Var(\hat{\beta}_{ols})$ as $X^TX(X^TX+\lambda I)^{-1} \geq \sigma^2(X^TX+\lambda I)^{-1}$ $X^TX(X^TX+\lambda I)^{-1}$ is in denominator.So as big is λ less is variance.