Stat 897 Fall 2017 Data Analysis Assignment 5

Penn State

Due September 24, 2017

(1) In the first part of this assignment we will use the College data found in the ISLR library. Split the data into a training set of size 100 and test set with the rest. Your goal is to predict the number of applications received using the other variables in the data set.

(a) Fit a "best" model obtained from your previous assignment on the training set and report the test error for this model.

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

(c) Fit a lasso model on the training set, with λ chosen by cross-validation. Report the test error obtained.

(d) Compare the result obtained in (a) – (c). How accurately can we predict the number of college applications? Is there much difference among the test errors resulting from different approaches?

(e) Now partition the data into a training set of size 600 and a test set with the rest. Compare the test errors from the "best" linear regression model, ridge and lasso models. Note that, the "best" model here may not be the "best" model obtained before.

(f) Do you see any difference between the two sets of results? Comment.

(2) The file Sp.Rdv contains daily returns for 501 stocks in 2016. It was created as follows.

(You don't need to run this, i.e. leave eval=FALSE)

```
library("quantmod")
sp = read.csv("SP500HistoricalComponents.csv")
Symbols = sp[which(sp$X12.31.2016 == "X"), "Ticker"]
StartDate = '2016-01-01'
EndDate = '2016-12-31'

Stocks = lapply(Symbols, function(sym) {
  print(sym)
  dailyReturn(na.omit(getSymbols(as.character(sym), from = StartDate, to = EndDate,
                                 auto.assign = FALSE, src = "yahoo")))
})

SPreturns = do.call(merge, Stocks)
colnames(SPreturns) = Symbols

#Clean out cols with NA
spreturns = SPreturns[ , colSums(is.na(SPreturns)) == 0]
save(spreturns, file = "spreturns.Rda")
```

Make sure you downloaded the data from the assignment page, and you can load it using:

```
library(quantmod)
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: TTR
```

```
## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```
load("spreturns.Rda")
```

I have constructed a secret long-only portfolio chosen from these stocks. It contains between five and twenty stocks. The daily return of this portfolio for each trading day of 2016 is in the object `portfolioreturns` which you can load from file with:

```
load("portfolioreturnsstatic.Rda")
```

Your goal is to recover the stocks and weights of the secret portfolio.

Note that you can think of a portfolio as a vector of nonnegative weights that sum to one. For simplicity, we are assuming that this portfolio is rebalanced daily at the closing prices. Then if the daily returns vector on date d is r_d and the weight vector is w_d, the daily return for the portfolio is the dot product $r_d \cdot w_d$. If this were a buy-and-hold portfolio, we would have to back into the returns more carefully.

(a) First try to fit an ordinary regression (lm) with `portfolioreturns` as the response and `spreturns` as the predictors. What happens? What problem do you run into?

(b) Now use elasticnet (Lasso, Ridge, or a combination, i.e. glmnet) instead of linear regression to model the secret portfolio. Once get a smaller set of variables from the shrinkage, refit a linear model with only those predictors. [Note, you may want to check out the `plotmo` and `pander` packages for nice plots and outputs for your models.]

(c) Here is how the portfolio was created. We're essentially randomly sampling columns from spreturns to have non-zero coefficients (portfoliowts) and then generating returns from that.

```
t = runif(ncol(spreturns))
thresh = .98
mask = t > thresh
w = runif(ncol(spreturns))
sum(mask) # number of chosen coefficients
portfoliowts = w * mask / sum(w * mask)
myportfolioreturns = spreturns %*% portfoliowts
```

Write a function that takes a threshold as input and produces the total error for estimated weights from lasso to the true weights. To do this you will need (1) a function that generates weights and returns for a given threshold function, (2) a function that takes the the returns and outputs the estimated coefficeints from a lasso, and (3) a function that takes the estimated coefficients and returns the error relative to the true weights. Plot the errors for a variety of different thresholds between 0.5 and 1.