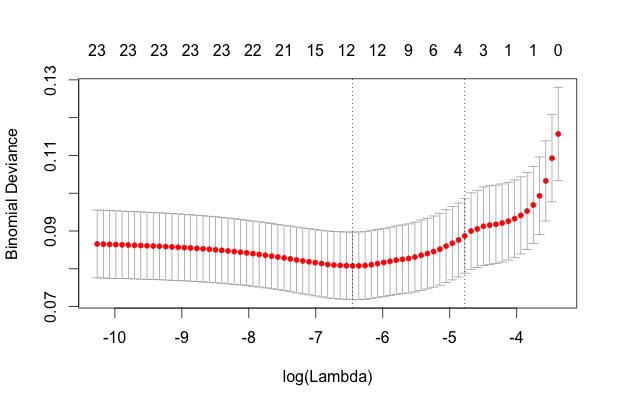
**A Short Introduction To Lasso Regression**

David Goody

1. This explanation assumes the reader has knowledge of generalised linear modelling (GLM), logistic regression and ridge regression.
2. This is an extension of ridge regression that not only shrinks coefficients, but also drops those that are of less value. This highlights which of the data variables have less predictive power and creates a model based solely on the indicators with greatest predictive power. In layperson’s terms this can be thought of as similar to testing the statistical significance of each of the variables within a GLM models and retaining the most important (forward stepwise regression)[[1]](#footnote-1). However like ridge regression the coefficients for the variables are shrunken in order to reduce the variance. The chart below highlights this combination of shrinkage and variable selection – it is the lasso version of the earlier ridge regression chart.

|  |  |
| --- | --- |
|  | Image sourced from:  [https://onlinecourses.science.psu.edu/](https://onlinecourses.science.psu.edu/stat857/node/158)  stat857/node/158 |

1. As with ridge regression the initial model fitting gives you a range of models based on the different tuning parameters implied. Again you would typically seek to pick the model with the lowest mean square predicted error calculated using cross-validation. In the chart below the number of variables included goes from 0 on the right hand side to 23 on the left hand side. The lowest mean square predicted error occurs with 12 variables included. You could also pick a more sparse version of the model based selecting the point at which you are within 1 standard error of the mean square error. This is the dotted line on the right which occurs with 3 variables. The more sparse model can guard against over-fitting and can highlight what the core factors within a model are.



**Further Reading**

Wikipedia - <http://en.wikipedia.org/wiki/Least_squares#Lasso_method>

Penn State University - <https://onlinecourses.science.psu.edu/stat857/node/158>

**Annex A – Undertaking lasso regression in R**

#EXAMPLE OF APPLYING LASSO REGRESSION IN R

#This example uses is based on the passenger list from the Titanic. We are trying to

#predict who survived and who died based on their age (Age), the fare they paid (Fare)

#gender (Female), embarkation point (South), how many siblings/spouses in each party (SibSp)

#how many parents/children in each passengers party (Parch)

#The dataset is used is publically available. It is the basis of a machine learning

#example on Kaggle (https://www.kaggle.com/c/titanic-gettingStarted).

##########################################################################

#SET-UP

#Install the ridge package (note can also do ridge regression in GLMNET package)

install.packages("glmnet")

require("glmnet")

#Set the working directory where you will be loading and saving data from

setwd("/Users/datascientist3/Desktop/R Packages/Kaggle Titanic")

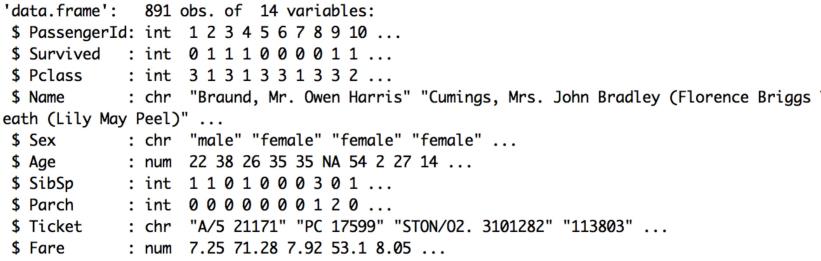
#Load Titanic dataset that we will train the data on

Titanic.Train <- read.csv("titanic\_train\_kaggle.csv", stringsAsFactors=FALSE)

#HAVE A LOOK AT THE DATA YOU ARE GOING TO USE

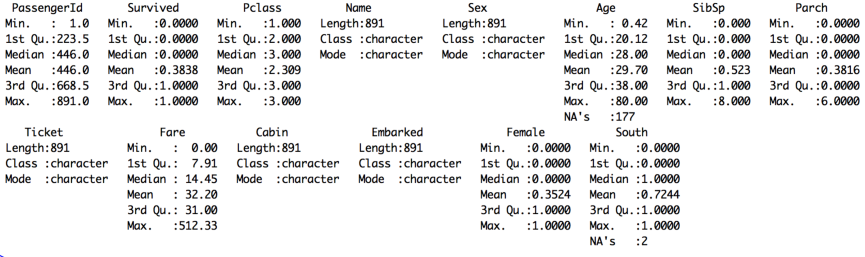
#Review structure of the data (shows field type and outputs form first few cases)

str(Titanic.Train)



#Look at a summary of the data (shows distribution of data for numeric fields)

summary(Titanic.Train)



#Convert text fields to dummy variables with 1/0 flags for use in equation

#Convert gender field into a 1/0 flag for female status

Titanic.Train$Female[Titanic.Train$Sex=='female'] <- 1

Titanic.Train$Female[Titanic.Train$Sex=='male'] <- 0

#Convert embarkation point 1/0 flag for those boarding at Southampton

Titanic.Train$South[Titanic.Train$Embarked=='S'] <- 1

Titanic.Train$South[Titanic.Train$Embarked!='S'] <- 0

#Over-write NA value for age with average as Lasso package can handle NA values

Titanic.Train$Age[is.na(Titanic.Train$Age)] <- 29.70

#FIT THE MODEL

#Apply Lasso model to predict whether passengers survived based on the 6 variables

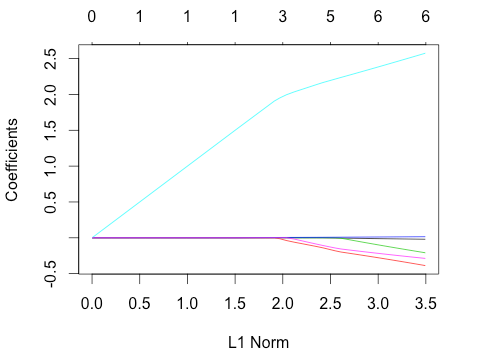
#These are variables in columns 6, 7, 8, 10, 13 and 14 in the dataset

Titanic.Train.Lasso.model <- glmnet(as.matrix(Titanic.Train[,c(6,7,8,10,13,14)]),Titanic.Train$Survived,

family="binomial", alpha = 1)

#Show a plot of the shrinkage applied to the coefficients

plot(Titanic.Train.Lasso.model)

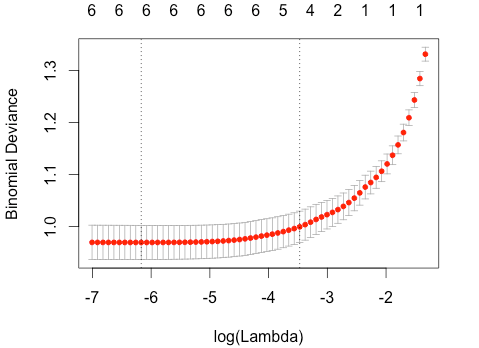


#Run cross-validation of the results to see which level of shrinkage is most effective

cv.Titanic.Train.Lasso.model <- cv.glmnet(as.matrix(Titanic.Train[,c(6,7,8,10,13,14)]),

Titanic.Train$Survived, family="binomial")

plot(cv.Titanic.Train.Lasso.model)



#Shows the coefficients from the model for them minimum mean square error

predict(Titanic.Train.Lasso.model,as.matrix(Titanic.Train[,c(6,7,8,10,13,14)]),

s=cv.Titanic.Train.Lasso.model$lambda.min,type="coefficients")

#Shows the coefficients from the model for a sparser model

predict(Titanic.Train.Lasso.model,as.matrix(Titanic.Train[,c(6,7,8,10,13,14)]),

s=cv.Titanic.Train.Lasso.model$lambda.1se,type="coefficients")

|  |  |
| --- | --- |
| **Coefficients with minimum mean square error** | **Coefficients for sparser model** |

#Shows the coefficients for each of the different points shown on cross-validation plot

coef(Titanic.Train.Lasso.model)

#APPLY THE MODEL

#We can take the model we've fitted and apply it to our training data

#This will give us a percentage value for each record.

#This is our prediction of how likely they were to survive

#The predict function is of the form predict([model name],[dataset name], [output required])

#In this case response gives us the % chance that they survive according to our model

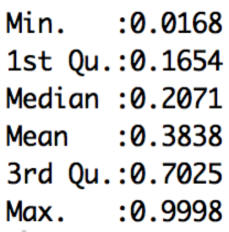
Titanic.Train.Lasso.Preds <- predict(Titanic.Train.Lasso.model,as.matrix(Titanic.Train[,c(6,7,8,10,13,14)]),

s=cv.Titanic.Train.Lasso.model$lambda.min,type="response") #Minimum lambda

#Look at a summary of our predictions. The % chance of survival varies from 2% to 100%

#There are also 179 cases where the model can predict due to missing data

summary(Titanic.Train.Lasso.Preds)



#Match the predictions back onto the main dataset to create a new file

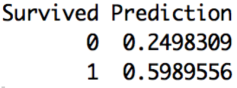
Titanic.Train.With.Predictions <- data.frame(Titanic.Train, Prediction = Titanic.Train.Lasso.Preds)

#Show the average prediction chance of survival for those who did live or die (Survived = 1 or 0)

#Model gives an average prediction of 27% for those who died and 61% for those who lived

aggregate(Prediction ~ Survived, data=Titanic.Train.With.Predictions,

FUN=function(x) {sum(x)/length(x)})



#Look at an individual result - the fifth case in the dataset

#Mr William Henry Allen did not survive (Survived = 0)

#Our model gave him a 15% chance of survival. Not very good odds! The model is doing OK here.

Titanic.Train.With.Predictions [5,] #The square brackets mean take the 5th row (then comma) and show all columns



#We can save our results to a csv file

write.csv(Titanic.Train.With.Predictions, file="Titanic.Train.With.Predictions.Lasso.csv")

1. <http://statweb.stanford.edu/~tibs/lasso/simple.html> [↑](#footnote-ref-1)