**A Short Introduction To Ridge regression**

David Goody

1. This explanation assumes the reader has knowledge of generalised linear modelling (GLM) and logistic regression.
2. Ridge regression is an extension of GLM that “shrinks” the coefficients to gain a more accurate model. This introduces some bias, but can greatly reduce the variance, resulting in a better mean-squared error. This means it should be able to produce a more accurate model than the basic GLM approach. To apply ridge regression you set up a GLM style equation but introduce a tuning parameter (lambda). The modelling algorithm then fits a range of models based on different values of this tuning parameter. These can vary from 0 (which gives the same result as GLM) to infinity (at which point the variables have all been shrunken to 0 – giving a flat line).
3. The following chart highlights how the parameters are shrunken as the tuning parameter varies (here the x-axis is the level of shrinkage applied). At the extreme right the values shown for each line are the parameter values you get from GLM. As you move to the left these shrink towards 0 – some faster than others. This is the tuning of the GLM model that creates ridge regression.

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

1. A process called generalised cross-validation is typically used to select the most accurate model out of all of those tested based on the smallest mean square error. This tests the models on a series of subsets of the data in order to guard against over-fitting the model. This ridge regression approach can be far more effective than GLM in dealing with multi co-linearity in models (where a number of the underlying variables are closely related to each other).

**Further Reading**

Wikipedia - <http://en.wikipedia.org/wiki/Tikhonov_regularization>

Penn State University - <https://onlinecourses.science.psu.edu/stat857/node/155>

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

#EXAMPLE OF APPLYING RIDGE 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("ridge")

require("ridge")

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

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

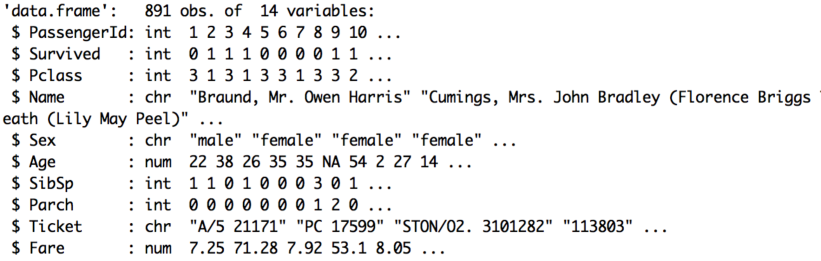
#Load Titanic dataset that we will train the data on

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

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

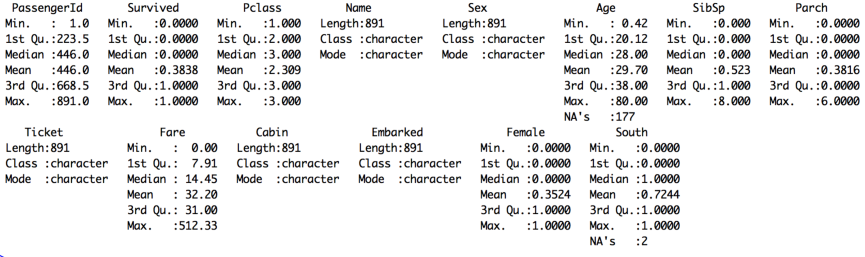
#Review structure of the data (shows field type and outputs form first few cases)

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=='C'] <- 0

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

#FIT THE MODEL

#Apply Ridge model to predict whether passengers survived based on the 5 variables

#This is a basic linear equation (y = a + bx + cz + .....)

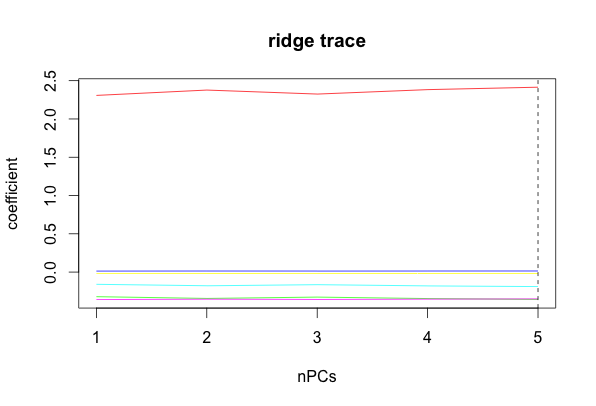
Titanic.Train.Ridge.model <- logisticRidge(Survived ~ Female + Age + SibSp + Parch + Fare + South, #Equation

data=Titanic.Train, #Dataset

lambda="automatic") #Algorithm automatically finds optional lambda

#Trace plot of ridge regression - not a huge amount of variation here

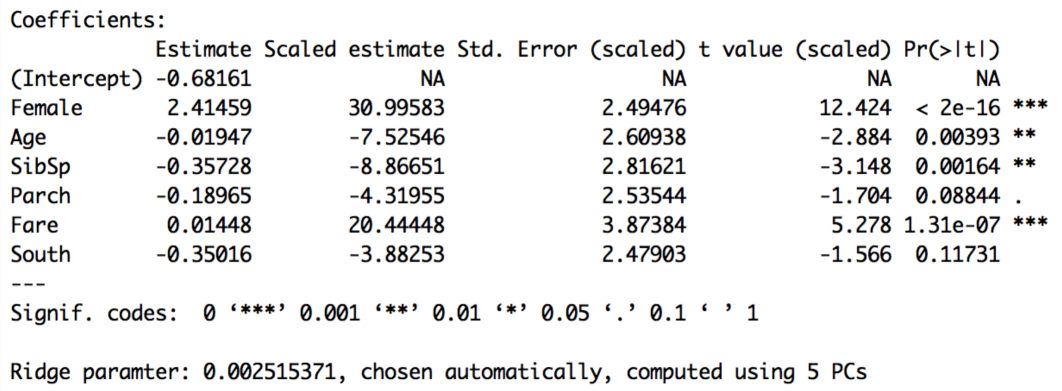
plot(Titanic.Train.Ridge.model)



#Shows a summary of the model including the coefficients applied and whether

#they are statistically significant (note that statistical significance may be misleading for ridge regression)

summary(Titanic.Train.Ridge.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])

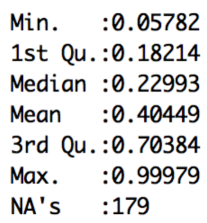
#In this case response gives us the % chance that they survive according to our model

Titanic.Train.Ridge.Preds <-predict(Titanic.Train.Ridge.model,Titanic.Train,type="response")

#Look at a summary of our predictions. The % chance of survival varies from 5% to 100%

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

summary(Titanic.Train.Ridge.Preds)



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

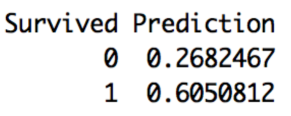
Titanic.Train.With.Predictions <- data.frame(Titanic.Train, Prediction = Titanic.Train.Ridge.Preds)

#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 17% 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.csv")