Ridge, Lasso and ElasticNet

Hakan Gogtas

11/22/2020

Ridge regression

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:

```
data("Seatbelts")
Seatbelts <- data.frame(Seatbelts)
str(Seatbelts)
```

```
## 'data.frame':    192 obs. of  8 variables:
##  $ DriversKilled: num  107 97 102 87 119 106 110 106 107 134 ...
##  $ drivers      : num  1687 1508 1507 1385 1632 ...
##  $ front        : num  867 825 806 814 991 ...
##  $ rear         : num  269 265 319 407 454 427 522 536 405 437 ...
##  $ kms          : num  9059 7685 9963 10955 11823 ...
##  $ PetrolPrice  : num  0.103 0.102 0.102 0.101 0.101 ...
##  $ VanKilled    : num  12 6 12 8 10 13 11 6 10 16 ...
##  $ law          : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
head(Seatbelts,n=10)
```

```
##    DriversKilled drivers front rear   kms PetrolPrice VanKilled law
## 1            107    1687   867  269  9059   0.1029718        12   0
## 2             97    1508   825  265  7685   0.1023630         6   0
## 3            102    1507   806  319  9963   0.1020625        12   0
## 4             87    1385   814  407 10955   0.1008733         8   0
## 5            119    1632   991  454 11823   0.1010197        10   0
```

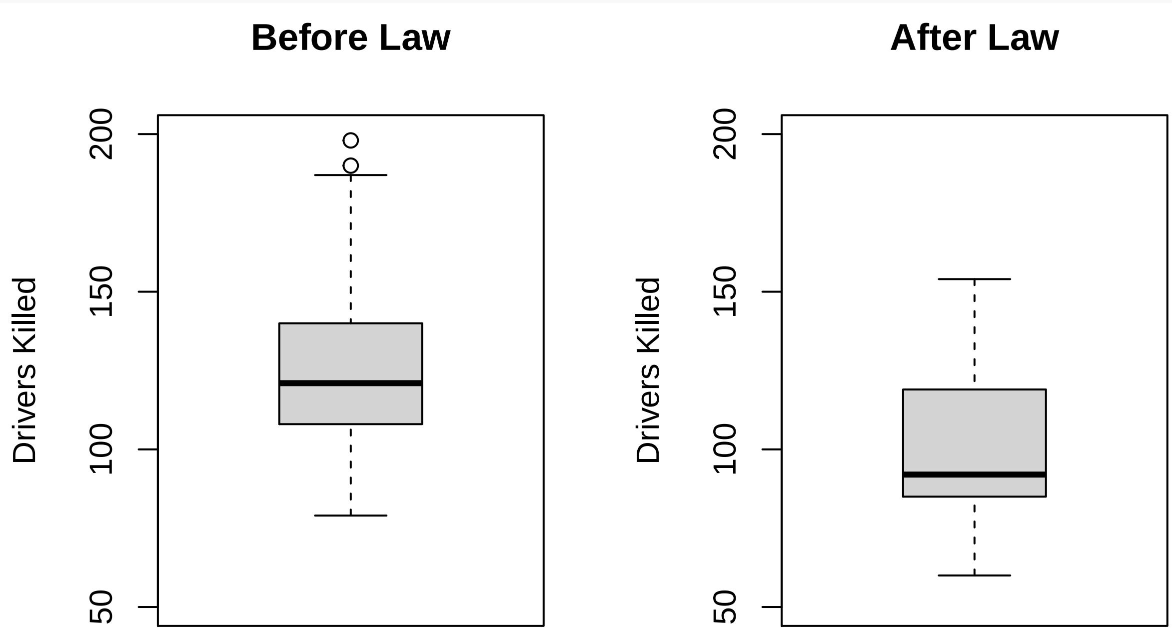
```
## 6             106    1511    945  427 12391   0.1005812        13   0
## 7             110    1559   1004  522 13460   0.1037740        11   0
## 8             106    1630   1091  536 14055   0.1040764         6   0
## 9             107    1579    958  405 12106   0.1037740        10   0
## 10            134    1653    850  437 11372   0.1030264        16   0
```

```r
summary(Seatbelts)
```

```
##  DriversKilled      drivers         front             rear      
##  Min.   : 60.0   Min.   :1057   Min.   : 426.0   Min.   :224.0  
##  1st Qu.:104.8   1st Qu.:1462   1st Qu.: 715.5   1st Qu.:344.8  
##  Median :118.5   Median :1631   Median : 828.5   Median :401.5  
##  Mean   :122.8   Mean   :1670   Mean   : 837.2   Mean   :401.2  
##  3rd Qu.:138.0   3rd Qu.:1851   3rd Qu.: 950.8   3rd Qu.:456.2  
##  Max.   :198.0   Max.   :2654   Max.   :1299.0   Max.   :646.0  
##       kms          PetrolPrice        VanKilled           law        
##  Min.   : 7685   Min.   :0.08118   Min.   : 2.000   Min.   :0.0000  
##  1st Qu.:12685   1st Qu.:0.09258   1st Qu.: 6.000   1st Qu.:0.0000  
##  Median :14987   Median :0.10448   Median : 8.000   Median :0.0000  
##  Mean   :14994   Mean   :0.10362   Mean   : 9.057   Mean   :0.1198  
##  3rd Qu.:17202   3rd Qu.:0.11406   3rd Qu.:12.000   3rd Qu.:0.0000  
##  Max.   :21626   Max.   :0.13303   Max.   :17.000   Max.   :1.0000
```

seatbelt legislation was introduced on January 31, 1983, so it is convenient to split the dataset into two (before/after the legislation). This subdivision will be executed with the subset() function that returns subsets of vectors, matrices, or data frames which meet conditions. The condition will be $law == 0$ (before the legislation), and $law! = 1$ (after the legislation):

```r
BeforeLaw <-subset(Seatbelts,law==0)
AfterLaw <- subset(Seatbelts,law!=0)
par(mfrow=c(1,2))
boxplot(BeforeLaw$DriversKilled,ylim=c(50,200),main="Before Law",ylab="Drivers Killed")
boxplot(AfterLaw$DriversKilled,ylim=c(50,200),main="After Law",ylab="Drivers Killed")
```



It can be seen that there were less drivers killed after the legislation was passed. After carrying out an exploratory analysis, we return to our goal: we are interested in predicting drivers killed using a multi-linear model. To perform Ridge regression, we will use the glmnet package that provides methods to algorithm regularization.

This package provides extremely efficient procedures for fitting the Ridge, Lasso, and ElasticNet regularization paths for linear regression, logistic and multinomial regression models, Poisson regression, and the Cox model. The algorithm is extremely fast and exploits sparsity in the input matrix where it exists. A variety of predictions can be made from the fitted models. The main function in the package is glmnet(). This function fits a generalized linear model (GLM) via penalized maximum likelihood. The regularization path is computed for the Ridge, Lasso, or ElasticNet penalty at a grid of values for the regularization parameter lambda. Can deal with all shapes of data, including very large sparse data matrices.

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.0-2
```

```
#need to define X and Y matrix for glmnet
#take out Y and law indicator from data for X (all independent variable)
x <- model.matrix(DriversKilled~., BeforeLaw)[,-c(1,8)]
y <- BeforeLaw$DriversKilled
RidgeMod <- glmnet(x, y, alpha=0, nlambda=100,lambda.min.ratio=0.0001)
```

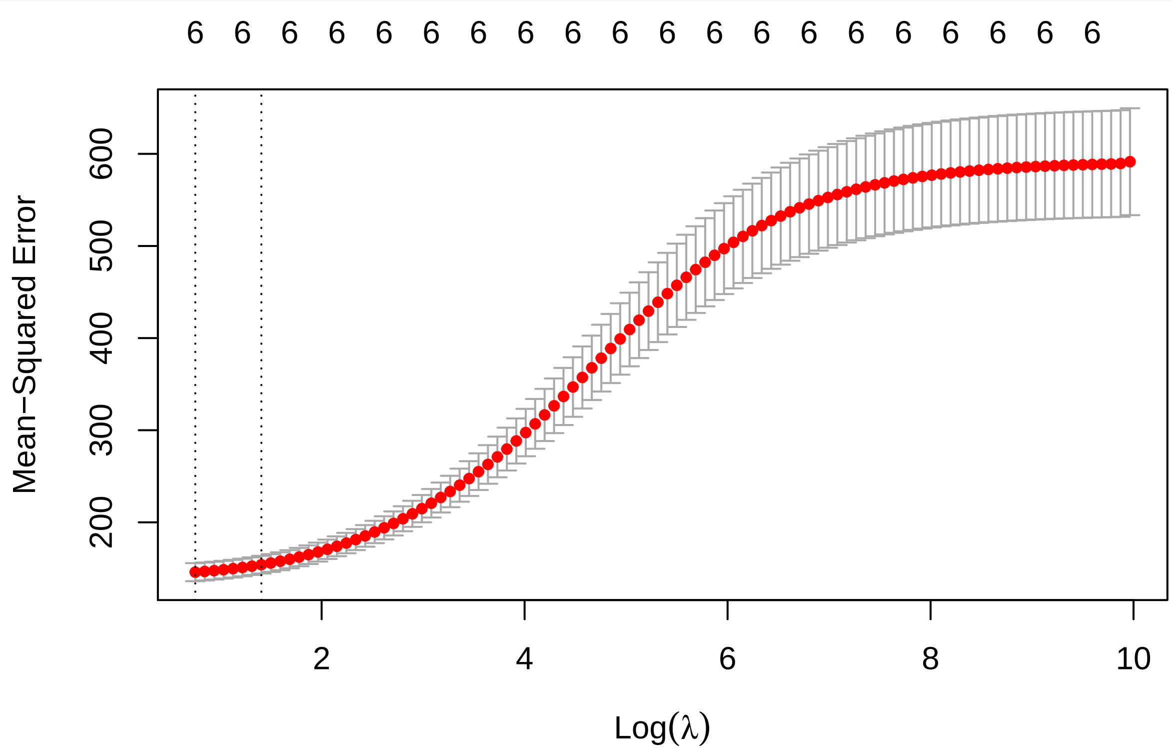
In the glmnet function are used the following arguments:

-nlambda=100: Set the number of lambda values (the default is 100)

-lambda.min.ratio=0.0001: Set the smallest value for lambda, as a fraction of lambda.max, the (data derived) entry value (that is, the smallest value for which all coefficients are zero)

we will use cross valdidation to select the best lamda

```
CvRidgeMod <- cv.glmnet(x, y, alpha=0, nlambda=100,lambda.min.ratio=0.0001)
par(mfrow=c(1,1))
plot(CvRidgeMod)
```



```r
best.lambda <- CvRidgeMod$lambda.min
best.lambda
```

```
## [1] 2.127603
```

The figure includes the cross-validation curve (red dotted line) and upper and lower standard deviation curves along the λ sequence (error bars). In the beginning of the procedure (to the right of the figure), the MSE is very high, and the coefficients are restricted to be too small; and then at some point, it kind of levels off. This seems to indicate that the full model is doing a good job.

There are two vertical lines: one is at the minimum, and the other vertical line is within one standard error of the minimum. The second line is a slightly more restricted model that does almost as well as the minimum, and sometimes, we'll go for that. These lines then lie at two lambda values:

-lambda.min is the value of λ that gives the minimum mean cross-validated error

-lambda.1se, gives the most regularized model such that the error is within one standard error of the minimum

At the top of the plot, you actually see how many nonzero variables' coefficients are in the model. There are all six variables in the model (five variables, plus the intercept), and no coefficient is zero.

Once we have the best lambda, we can use predict to obtain coefficients:

```r
predict(RidgeMod, s=best.lambda, type="coefficients")[1:6, ]
```

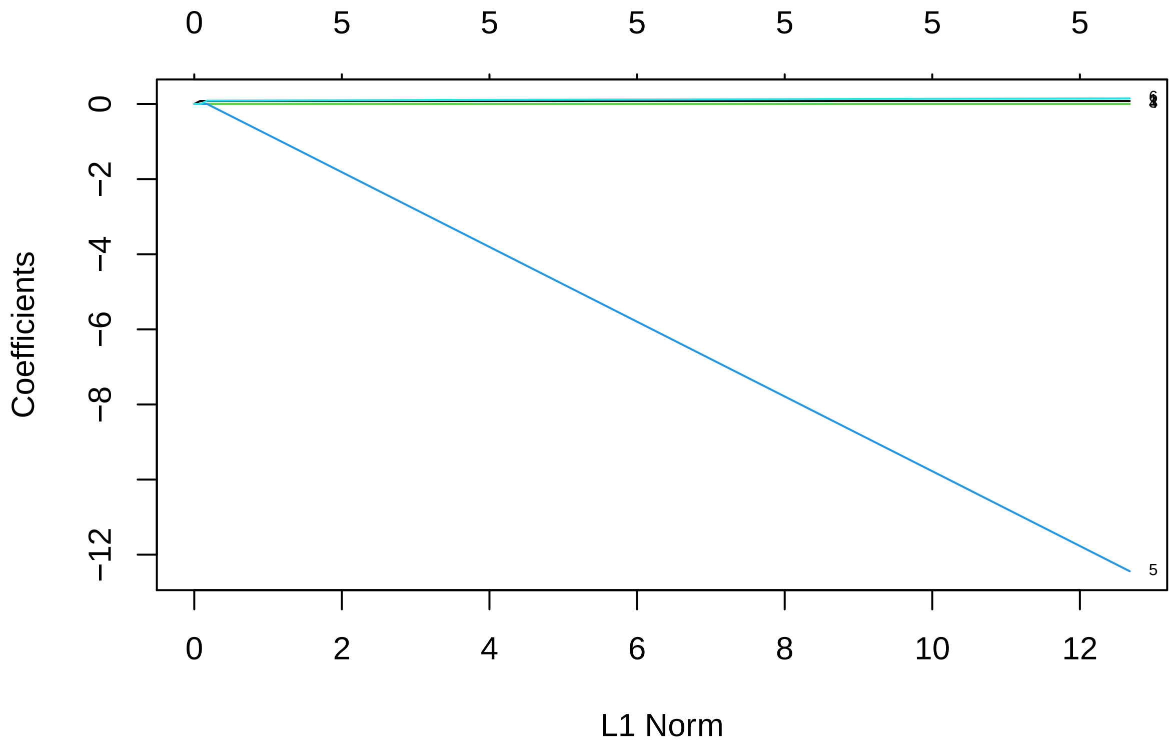
```
##   (Intercept)       drivers         front          rear           kms
## -1.219728e+01  6.655047e-02  1.851418e-02 -2.842390e-03  5.644939e-04
##   PetrolPrice
## -1.921260e+01
```

This is the best regression model for our data.

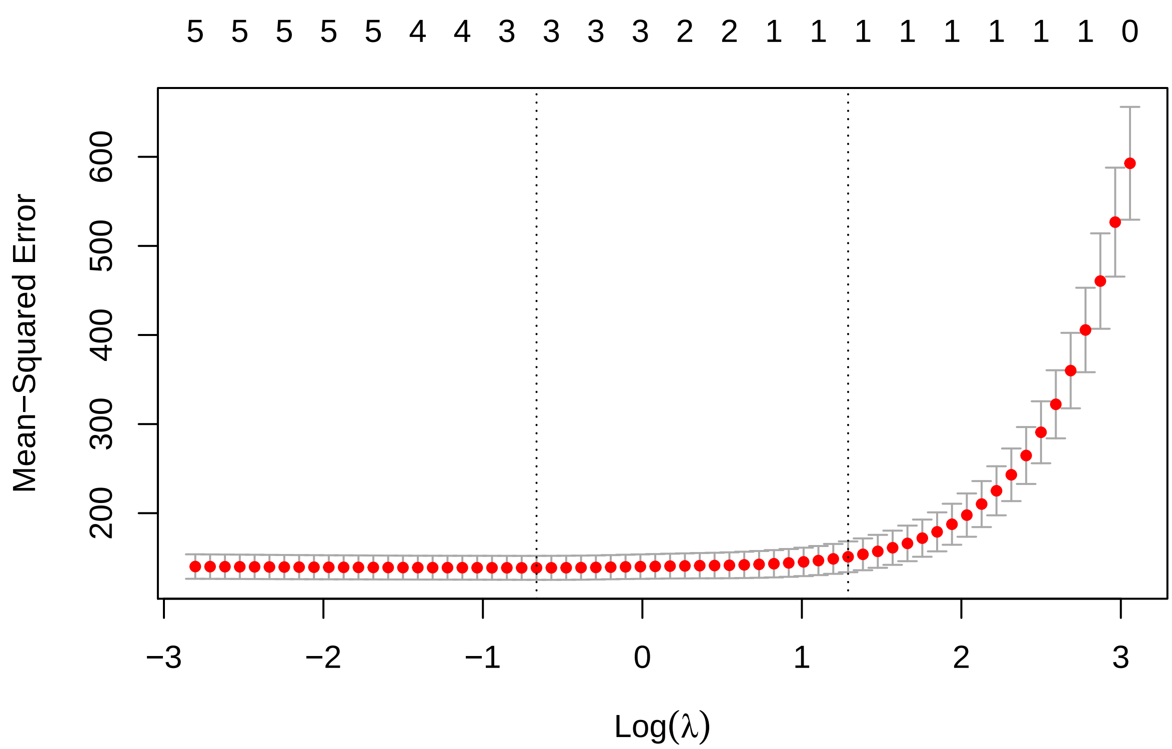
Lasso Regression

To perform Lasso regression, we will use the same dataset. On the R function below $\alpha = 1$ for Lasso regression. Everyting is very similar.

```r
LassoMod <- glmnet(x, y, alpha=1, nlambda=100,lambda.min.ratio=0.0001)
plot(LassoMod,xvar="norm",label=TRUE)
```




```r
CvLassoMod <- cv.glmnet(x, y, alpha=1, nlambda=100,lambda.min.ratio=0.0001)
plot(CvLassoMod)
```




```r
best.lambda <- CvLassoMod$lambda.min
best.lambda
```

```
## [1] 0.5149073
```

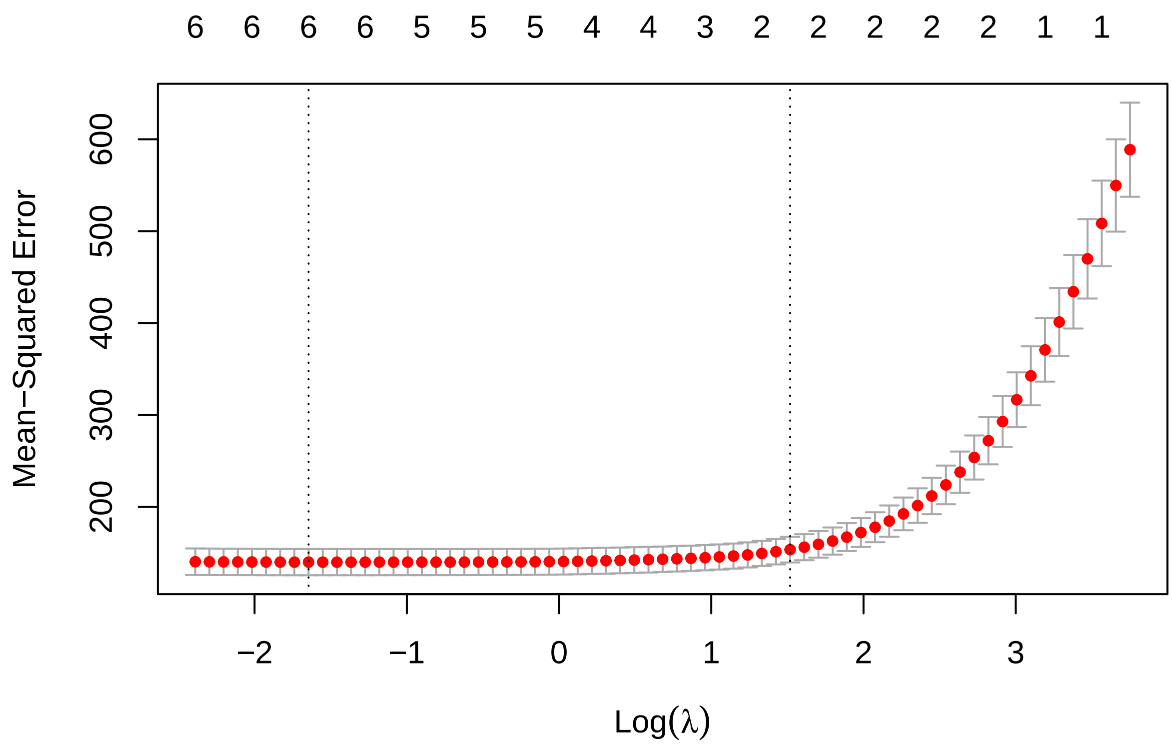
```r
coef(CvLassoMod, s = "lambda.min")
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                         1
## (Intercept) -1.661050e+01
## drivers      7.853578e-02
## front        .
## rear         5.030482e-03
## kms          3.845301e-04
## PetrolPrice  .
## VanKilled    .
```

We have confirmed that the Lasso method is able to make a selection of variables. Ultimately, we can say that both Lasso and Ridge balance the trade-off bias-variance with the choice of λ. Lasso implicitly assumes that part of the coefficients are zero, or at least not significant. Lasso tends to have a higher performance than Ridge in cases where many predictors are not actually tied to the response variables. In opposite cases, the Ridge tends to have better performance. Both approaches can be compared by cross-validation.

ElasticNet

```r
CvElasticnetMod <- cv.glmnet(x, y,alpha=0.5,nlambda=100,lambda.min.ratio=0.0001)
plot(CvElasticnetMod)
```




```r
best.lambda <- CvElasticnetMod$lambda.min
best.lambda
```

```
## [1] 0.1929684
```

```r
coef(CvElasticnetMod, s = "lambda.min")
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
```

```
##                          1
## (Intercept) -2.369388e+01
## drivers      7.956498e-02
## front        1.591839e-03
## rear         2.117853e-03
## kms          7.020322e-04
## PetrolPrice -8.452946e+00
## VanKilled    1.418822e-01
```