Anly590_HW0_Archer

Alex Archer

9/18/2018

1

1.1

Lasso

```r
# the Hitters dataset is in the ISLR package
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 3.4.2
```

```r
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 3.4.2
## Loading required package: Matrix
## Loading required package: foreach
## Warning: package 'foreach' was built under R version 3.4.3
## Loaded glmnet 2.0-13
```

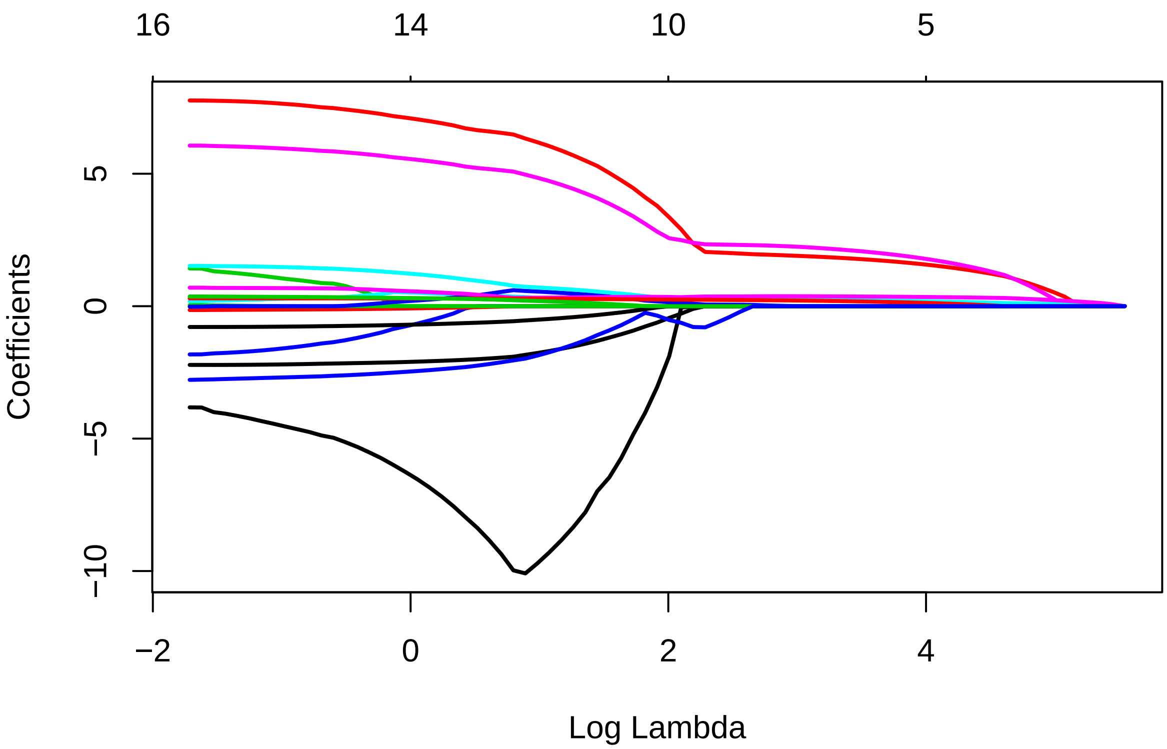
```r
# take a look at the variables
head(Hitters)
```

```
##                   AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits
## -Andy Allanson      293   66     1   30  29    14     1    293    66
## -Alan Ashby         315   81     7   24  38    39    14   3449   835
## -Alvin Davis        479  130    18   66  72    76     3   1624   457
## -Andre Dawson       496  141    20   65  78    37    11   5628  1575
## -Andres Galarraga   321   87    10   39  42    30     2    396   101
## -Alfredo Griffin    594  169     4   74  51    35    11   4408  1133
##                   CHmRun CRuns CRBI CWalks League Division PutOuts Assists
## -Andy Allanson         1    30   29     14      A        E     446      33
## -Alan Ashby           69   321  414    375      N        W     632      43
## -Alvin Davis          63   224  266    263      A        W     880      82
## -Andre Dawson        225   828  838    354      N        E     200      11
## -Andres Galarraga     12    48   46     33      N        E     805      40
## -Alfredo Griffin      19   501  336    194      A        W     282     421
##                   Errors Salary NewLeague
## -Andy Allanson        20     NA         A
## -Alan Ashby           10  475.0         N
## -Alvin Davis          14  480.0         A
## -Andre Dawson          3  500.0         N
## -Andres Galarraga      4   91.5         N
## -Alfredo Griffin      25  750.0         A
```

```r
# remove categorical predictors
Hitters <- Hitters[,-c(14, 15, 20)]
Hitters <- na.omit(Hitters)

# set x and y to be passed into model
y <- Hitters$Salary
x <- model.matrix(Salary ~., Hitters)[,-1]

# creat and plot lasso
lasso.mod <- glmnet(x, y, alpha = 1)
plot(lasso.mod, xvar = "lambda", lwd = 2)
```



From the trajectories of the coefficients in the mode, we can obtain an idea of how the lasso is doing variable selection. The penalty parameter is forcing the coefficients towards 0.

```r
# find last 3 predictors in model
coef(lasso.mod)[,5]
```

```
##  (Intercept)        AtBat         Hits        HmRun         Runs
## 444.06785122   0.00000000   0.08064999   0.00000000   0.00000000
##          RBI        Walks        Years       CAtBat        CHits
##   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000
##       CHmRun        CRuns         CRBI       CWalks      PutOuts
##   0.00000000   0.06719193   0.17823025   0.00000000   0.00000000
##      Assists       Errors
##   0.00000000   0.00000000
```

The last 3 variables in the model are Hits, CRuns, and CRBI.

Next, we want to find the optimal value of the regularization penalty using cross validation.

```r
# use cv to find optimal value of penalty
cv.lasso <- cv.glmnet(x, y, alpha = 1)
bestlam <- cv.lasso$lambda.min
print(bestlam)
```

```
## [1] 2.935124
```

```r
print(coef(cv.lasso))
```

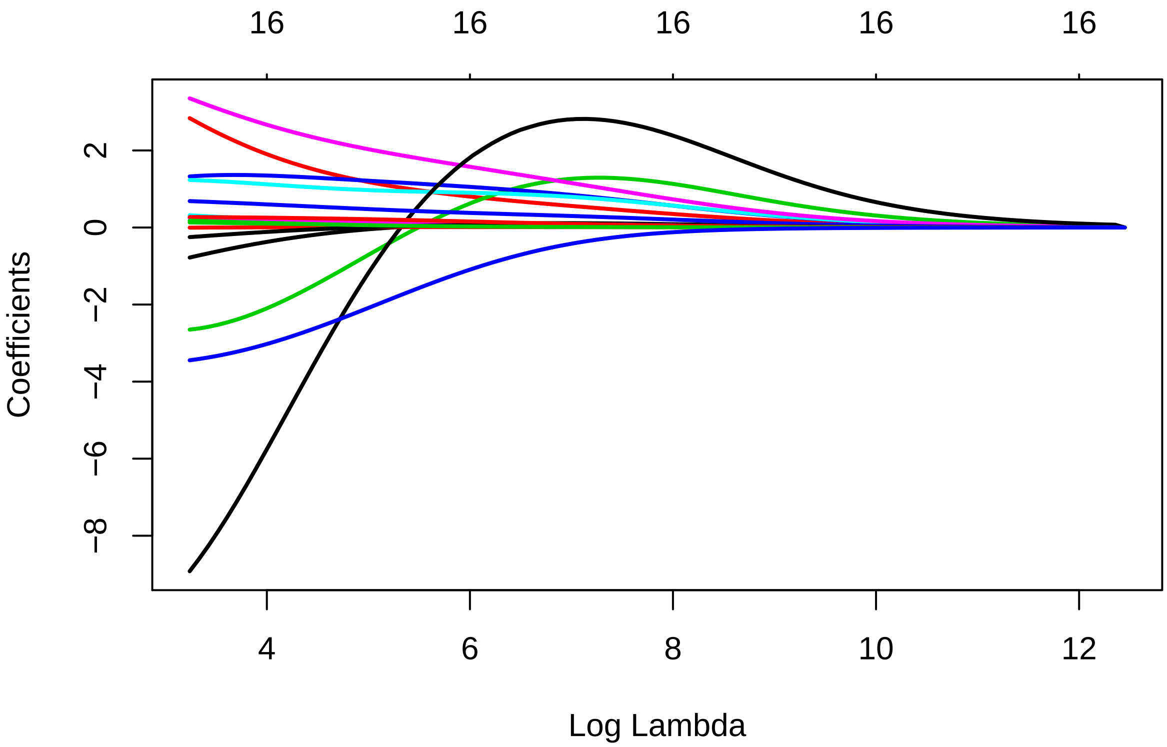
```
## 17 x 1 sparse Matrix of class "dgCMatrix"
##                        1
## (Intercept) 193.74263858
## AtBat         .
## Hits          1.21471320
## HmRun         .
## Runs          .
## RBI           .
## Walks         1.28957902
## Years         .
## CAtBat        .
## CHits         .
## CHmRun        .
## CRuns         0.12923755
## CRBI          0.31515925
## CWalks        .
## PutOuts       0.02533305
## Assists       .
## Errors        .
```

Using cross validation, we find the optimal value for λ. We also check the coefficients for this model and find that there are 5 predictors left.

1.2

Ridge

```r
# visualize ridge coef trajectories
ridge.mod <- glmnet(x, y, alpha = 0)
plot(ridge.mod, xvar = "lambda", lwd = 2)
```




```r
# find optimal penalty
cv.ridge <- cv.glmnet(x, y, alpha = 0)
print(cv.ridge$lambda.min)
```

```
## [1] 28.01718
```

Here we found the optimal value for the penalty parameter, but note that the coefficients will not go to 0 for the ridge.

The bias variance trade-off means that the realtionship between bias and variance in models is inverse. So if a model has low bias, it is likely sacrificing from high variance. As models become more complex and flexible they tend to have lower bias, but higher variance. The model can achieve low bias and can get so complex that it fits nearly every point in the data. But at this point, the model is overfit and has high variance. If new data were to be generated, the model would need to change drastically. There is also an issue if the model has extremely low variance but high bias. In this case, the model would likely not change much with new data, but it does not fit the data very well.

Thus regularization is a nice way to reduce the risk of overfitting, thus decreasing the variance in models. The Lasso can also be used for model selection. As in Number 1, some variable coefficients were reduced to 0 with the Lasso model. Additionally, in both the lasso and ridge models in Number 1, the variance is being controlled by shrinking the coefficient estimates to 0, when otherwise they may grow large. Finally, if we had a train and test set, we would find that prediction error is also lower when using regularization as above.