r-statistics.co (/) by Selva Prabhakaran

Search..

•

Tutorial

R Tutorial (R-Tutorial.html)

ggplot2

ggplot2 Short Tutorial (ggplot2-Tutorial-With-R.html)

ggplot2 Tutorial 1 - Intro (Complete-Ggplot2-Tutorial-Part1-With-R-Code.html)

ggplot2 Tutorial 2 - Theme (Complete-Ggplot2-Tutorial-Part2-Customizing-Theme-With-R-Code.html)

ggplot2 Tutorial 3 - Masterlist (Top50-Ggplot2-Visualizations-MasterList-R-Code.html)

ggplot2 Quickref (ggplot2-cheatsheet.html)

Foundations

Linear Regression (Linear-Regression.html)

Statistical Tests (Statistical-Tests-in-R.html)

Missing Value Treatment (Missing-Value-Treatment-With-R.html)

Outlier Analysis (Outlier-Treatment-With-R.html)

Feature Selection (Variable-Selection-and-Importance-With-R.html)

Model Selection (Model-Selection-in-R.html)

Logistic Regression (Logistic-Regression-With-R.html)

Advanced Linear Regression (Environments.html)

Advanced Regression Models

Advanced Regression Models (adv-regression-models.html)

Time Series

Time Series Analysis (Time-Series-Analysis-With-R.html)

Time Series Forecasting (Time-Series-Forecasting-With-R.html)

Contents

Ridge Regression

> Ridge Regression is a commonly used technique to address the problem of multi-collinearity. The effectiveness of the application is however debatable.

Introduction

Let us see a use case of the application of Ridge regression on the `longley` dataset. We will try to predict the `GNP.deflator` using `lm()` with the rest of the variables as predictors. This model and results will be compared with the model created using ridge regression.

```
library (car) # for VIF
library (ridge)
data(longley, package="datasets")  # initialize data
head (longley, 4)  # show top 4 rows of data
#>      GNP.deflator     GNP Unemployed Armed.Forces Population Year Employed
#> 1947         83.0 234.289      235.6        159.0    107.608 1947   60.323
#> 1948         88.5 259.426      232.5        145.6    108.632 1948   61.122
#> 1949         88.2 258.054      368.2        161.6    109.773 1949   60.171
#> 1950         89.5 284.599      335.1        165.0    110.929 1950   61.187
```

```
inputData <- data.frame (longley) # plug in your data here
colnames(inputData)[1] <- "response"  # rename response var
```

Calculate Correlations

```
XVars <- inputData[, -1] # X variables
round(cor(XVars), 2) # Correlation Test
#>               GNP Unemployed Armed.Forces Population Year Employed
#> GNP          1.00       0.60         0.45       0.99 1.00     0.98
#> Unemployed   0.60       1.00        -0.18       0.69 0.67     0.50
#> Armed.Forces 0.45      -0.18         1.00       0.36 0.42     0.46
#> Population   0.99       0.69         0.36       1.00 0.99     0.96
#> Year         1.00       0.67         0.42       0.99 1.00     0.97
#> Employed     0.98       0.50         0.46       0.96 0.97     1.00
```

Prepare Training And Test Data

```
set.seed(100) # set seed to replicate results
trainingIndex <- sample(1:nrow(inputData), 0.8*nrow(inputData)) # indices for 80% traini
ng data
trainingData <- inputData[trainingIndex, ] # training data
testData <- inputData[-trainingIndex, ] # test data
```

Predict Using Linear Regression

```
lmMod <- lm(response ~ ., trainingData)  # the linear reg model
summary (lmMod) # get summary
vif(lmMod) # get VIF

#>  VIF
#>        GNP   Unemployed Armed.Forces   Population         Year     Employed
#> 1523.74714     93.07635     10.74587    350.58472   2175.29221    182.93609

#>  Coefficients:
#>    (Intercept)           GNP    Unemployed  Armed.Forces    Population          Year
Employed
#>     7652.25192       0.39214       0.06462       0.01573      -2.33550      -3.83113
0.53060
```

There is significant multi-collinearity between GNP & Year and Population & Employed, with negative coefficients in 'population' and 'Employed'. These variables may not contribute much to explain the dependent variable, nevertheless, lets see what this model predicts.

```
predicted <- predict (lmMod, testData)  # predict on test data
compare <- cbind (actual=testData$response, predicted)  # combine actual and predicted
#>      actual predicted
#> 1949   88.2  88.45501
#> 1953   99.0  96.67492
#> 1957  108.4 106.59672
#> 1959  112.6 113.31106
mean (apply(compare, 1, min)/apply(compare, 1, max)) # calculate accuracy
#>  98.76%
```

Apply Ridge Regression On Same Data

```
linRidgeMod <- linearRidge(response ~ ., data = trainingData)  # the ridge regression mo
del
#>  No more Negative Coefficients!
#>   (Intercept)           GNP    Unemployed  Armed.Forces    Population          Year
Employed
#> -1.015385e+03  3.715498e-02  1.328002e-02  1.707769e-02  1.294903e-01  5.318930e-01
 5.976266e-01

predicted <- predict(linRidgeMod, testData)  # predict on test data
compare <- cbind (actual=testData$response, predicted)  # combine
#>      actual predicted
#> 1949   88.2  88.68584
#> 1953   99.0  99.26104
#> 1957  108.4 106.99370
#> 1959  112.6 110.95450
mean (apply(compare, 1, min)/apply(compare, 1, max)) # calculate accuracy
#>  99.10%
```

Clearly, in this case, ridge regression is successful in improving the accuracy by a minor but significant fraction.

Predicting With A Re-calibrated Linear Model

```
newlmMod <- lm(response ~ ., trainingData[, -c(2, 5, 6)]) # without "GNP", "Population"
 & "Year"
summary (newlmMod) # get summary
vif(newlmMod) # get VIF
#> Coefficients:
#>  (Intercept)    Unemployed  Armed.Forces      Employed
#>    -62.19771       0.03248       0.02714       2.24039
#>  VIF
#>   Unemployed Armed.Forces     Employed
#>     2.124153     1.452648     2.592474
predicted <- predict(newlmMod, testData) # predict on test data
compare <- cbind (actual=testData$response, predicted) # for comparison
mean (apply(compare, 1, min)/apply(compare, 1, max)) # calculate accuracy
#>  99.21%
```

The re-calibrated linear model yields better accuracy when the multicollinearity is taken care of. This analysis may not be sufficient to draw conclusions about the effectiveness of ridge regression. The intention, however, is to open up considerations for new modeling options for problem solving.