Machine Learning - Housekeeping

Jan-Philipp Kolb

10 Januar, 2020

Include all possible interaction effects

- This creates all combinations of two-way interactions

```
data(mtcars)
lm(mpg~(cyl+disp+hp)^2,data=mtcars)$coefficients
```

```
##   (Intercept)           cyl          disp            hp
##  5.600542e+01 -4.426716e+00 -1.183803e-01 -1.141578e-01  1.43
##        cyl:hp       disp:hp
##  1.556470e-02 -8.566769e-05
```

```
lm(mpg~(cyl+disp+hp)^3,data=mtcars)$coefficients
```

```
##   (Intercept)           cyl          disp            hp
##  9.289857e+01 -1.059493e+01 -3.864887e-01 -4.703167e-01  5.27
##        cyl:hp       disp:hp   cyl:disp:hp
##  6.733770e-02  2.808156e-03 -3.841232e-04
```

Colinearity diagnostics

- Collinearity implies two variables are near perfect linear combinations of one another.
- Multicollinearity involves more than two variables.
- In the presence of multicollinearity, regression estimates are unstable and have high standard errors.

variane inflation factor (VIF)

- VIF is the ratio of variance in a model with multiple terms, divided by the variance of a model with one term alone.
- It quantifies the severity of multicollinearity in an regression analysis.
- It provides an index that measures how much the variance (the square of the estimate's standard deviation) of an estimated regression coefficient is increased because of collinearity.

Interpretation

- The square root of the variance inflation factor indicates how much larger the standard error is, compared with what it would be if that variable were uncorrelated with the other predictor variables in the model.
- If the variance inflation factor of a predictor variable were 5.27 ($\sqrt{5.27} = 2.3$), this means that the standard error for the coefficient of that predictor variable is 2.3 times as large as it would be if that predictor variable were uncorrelated with the other predictor variables.

An ols model

```r
data(mtcars)
olsmod <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
```

```r
modmatx<-model.matrix(mpg ~disp+hp+wt+qsec,mtcars)[,-1]
mody <- mtcars$mpg
```

```r
library(genridge)
lambda <- c(0, 0.005, 0.01, 0.02, 0.04, 0.08)
lridge <- ridge(mody, modmatx, lambda=lambda)
```

Compute the vif

```
library(olsrr)
ols_vif_tol(olsmod)
```

```
## # A tibble: 4 x 3
##   Variables Tolerance   VIF
##   <chr>         <dbl> <dbl>
## 1 disp          0.125  7.99
## 2 hp            0.194  5.17
## 3 wt            0.145  6.92
## 4 qsec          0.319  3.13
```

Compare vif for ols and ridge

```
# https://rdrr.io/cran/genridge/man/vif.ridge.html
vif(olsmod)
```

```
##     disp       hp       wt     qsec
## 7.985439 5.166758 6.916942 3.133119
```

```
(vridge <- vif(lridge))
```

```
##           disp       hp       wt     qsec
## 0.000 7.985439 5.166758 6.916942 3.133119
## 0.005 7.955896 5.156178 6.890382 3.125994
## 0.010 7.926540 5.145636 6.863995 3.118906
## 0.020 7.868378 5.124669 6.811733 3.104842
## 0.040 7.754217 5.083192 6.709222 3.077150
## 0.080 7.534212 5.002024 6.511924 3.023453
```

An R^2 for ridge regression

```r
rsquare <- function(true, predicted) {
  sse <- sum((predicted - true)^2)
  sst <- sum((true - mean(true))^2)
  rsq <- 1 - sse / sst
  if (rsq < 0) rsq <- 0
  return (rsq)
}
```

Prediction and R^2

Make the prediction for the ridge model

```
library(glmnet)
cv_ridge <- cv.glmnet(modmatx,mody, alpha = 0)
pred <- predict(cv_ridge,s = cv_ridge$lambda.min, modmatx)
```

The R^2 for the ridge model

```
rsquare(mtcars$mpg,pred)
## [1] 0.827315
rsquare(mtcars$mpg,predict(olsmod))
## [1] 0.8351443
```