Properties of Least Squares

Rob

2/5/2018

The Data

Train/Test Loop

The Data

Let's read in some data that we will use to illustrate properties of least squares regression.

```
yx = read.csv("sim-reg-data.csv")
print(summary(yx))
```

```
##        y                  x1                 x2
##  Min.   :-2.2770   Min.   :0.00348   Min.   : 0.004425
##  1st Qu.: 0.4562   1st Qu.:2.33474   1st Qu.: 2.398081
##  Median : 1.2172   Median :4.83693   Median : 4.775174
##  Mean   : 1.2110   Mean   :4.90519   Mean   : 4.920732
##  3rd Qu.: 2.0133   3rd Qu.:7.48680   3rd Qu.: 7.440223
##  Max.   : 5.2494   Max.   :9.99582   Max.   : 9.999662
##        x3                 x4                 x5
##  Min.   :0.0001646   Min.   :0.001629   Min.   :0.001979
##  1st Qu.:0.2444854   1st Qu.:0.247012   1st Qu.:0.248805
##  Median :0.5096922   Median :0.514615   Median :0.515297
##  Mean   :0.5017638   Mean   :0.506709   Mean   :0.506733
##  3rd Qu.:0.7567860   3rd Qu.:0.761047   3rd Qu.:0.761768
##  Max.   :0.9999581   Max.   :1.006039   Max.   :1.005441
```

So we have a y and 5 x variables.

Let's go ahead and run the regression:

```
lmf= lm(y~.,yx)
print(summary(lmf))
```

```
##
## Call:
## lm(formula = y ~ ., data = yx)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4575 -0.6490  0.0287  0.6639  4.0229
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.024640   0.089157  -0.276    0.782
## x1           0.025323   0.007686   3.295    0.001 **
## x2           0.035590   0.007742   4.597 4.55e-06 ***
## x3           4.248433  10.888697   0.390    0.696
## x4          -5.126662   7.773133  -0.660    0.510
## x5           2.767445   7.835586   0.353    0.724
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.007 on 1994 degrees of freedom
## Multiple R-squared:  0.2434, Adjusted R-squared:  0.2415
## F-statistic: 128.3 on 5 and 1994 DF,  p-value: < 2.2e-16
```

What do *most people think the stars mean* ???

Train/Test Loop

Let's see if x1 and x2 are the *best* x's.

We will compare that subset to the subset which is just x3.

```
n=nrow(yx)
nd = 100
set.seed(99)
rmse =function(y,yhat) {sqrt(mean((y-yhat)^2))}
ntrain = floor(n*.75)
resM = matrix(0.0,nd,2)
for(i in 1:nd) {
  print(i)
  ii = sample(1:n,ntrain)
  dftrain= yx[ii,]; dftest = yx[-ii,]

  lm12 = lm(y~x1+x2,dftrain)
  resM[i,1] = rmse(dftest$y,predict(lm12,dftest))

  lm3 = lm(y~x3,dftrain)
  resM[i,2] = rmse(dftest$y,predict(lm3,dftest))
}
```

Ok, now let's use boxplots to look at the columns of rMat.

```
colnames(resM)= c("x12","x3")
boxplot(resM)
```