Machine Learning Homework 4

Rob McCulloch

2/13/2018

Contents

1 Out of Sample Predictive Performance of Variable Subsets

In class we looked at an example where we did a simple train/test loop to investigate the out of sample predictive performance of different variable subsets.

The pdf is on the webpage at:

http://www.rob-mcculloch.org/2018_ml/webpage/notes/properties.pdf

and the R markdown is at:

http://www.rob-mcculloch.org/2018_ml/webpage/notes/properties.Rmd

We compared the predictive performance of the subset (x1,x2) with that of the subset (x3).

The subset (x1,x2) are the starred variables in the usual R regression summary.

First, check the code!

Is (x3) really better than (x1,x2) in terms of out of sample prediction error?

Second, modify the code to compare the subsets (x1), (x2), (x3), (x4), (x5), and (x1,x2).
That is, try each single x variable and the subset (x1,x2).

How do they compare??

2 Properties of Least Squares

2.1 $\hat{\beta}$ by Matrix Operations

Let's compute

$$\hat{\beta} = (X'X)^{-1}X'y$$

and check that we get the same thing as the R function lm. Let's also check the FOCs (first order conditions).

We'll use the same data as in the previous problem.

I'll quickly show you the matrix operations in R and then you can try them on the data. First, I'll make a toy X and y use as in my examples.

Toy data:

```
n=4
X = cbind(rep(1,4),1:4) #cbind concatenates columns, rep repeats 1 four times.
y = c(1,2,4,3)
print(X); print(y)
##      [,1] [,2]
## [1,]    1    1
## [2,]    1    2
## [3,]    1    3
## [4,]    1    4
## [1] 1 2 4 3
is.matrix(X) #does cbind give me a matrix?
## [1] TRUE
```

Matrix operations:

```
t(X) #transpose of X
##      [,1] [,2] [,3] [,4]
## [1,]    1    1    1    1
## [2,]    1    2    3    4
xtx = t(X) %*% X # X'X, note the R form for matrix multiplication
xtxi = solve(xtx) # matrix inversion
xtx %*% xtxi # check we get the identity
##      [,1] [,2]
## [1,]    1    0
## [2,]    0    1
```

Compute $\hat{\beta}$:

```
bhat = solve(t(X)%*%X) %*% t(X) %*% y
bhat
##      [,1]
## [1,]  0.5
## [2,]  0.8
lmf = lm(y~.,data.frame(X[,2],y))
lmf$coef
## (Intercept)      X...2.
##         0.5         0.8
```

FOC:

```r
t(X) %*% (y- X %*% matrix(bhat,ncol=1))
##                [,1]
## [1,] -4.662937e-15
## [2,] -1.421085e-14
```

Ok, now compute $\hat{\beta}$ and check the first order conditions for the data from problem 1.
Just repeat what I just did on the toy data with the real data.

2.2 $\hat{\sigma}$ and Standard errors

Get $\hat{\sigma}$ and $se(\hat{\beta}_i)$ for the data for the problem 1 data directly from the formulas using R matrix operations and vector calculations.

That is, use:

$$\hat{\sigma}^2 = \frac{1}{n-p}||y - X\hat{\beta}||^2 = \frac{1}{n-p}\sum(y_i - x_i'\hat{\beta})^2$$

and

$$Var(\hat{\beta}) = \sigma^2(X'X)^{-1}.$$

Compare your numbers to the regression output.

Note, for the toy data:

```r
shat = summary(lmf)$sigma
cat("sigma hat is: ",shat,"\n")
## sigma hat is:  0.9486833
sqrt(diag(solve(t(X) %*% X))) * shat
## [1] 1.1618950 0.4242641
summary(lmf)
##
## Call:
## lm(formula = y ~ ., data = data.frame(X[, 2], y))
##
## Residuals:
##    1    2    3    4
## -0.3 -0.1  1.1 -0.7
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.5000     1.1619   0.430    0.709
## X...2.        0.8000     0.4243   1.886    0.200
##
## Residual standard error: 0.9487 on 2 degrees of freedom
## Multiple R-squared:   0.64,  Adjusted R-squared:   0.46
## F-statistic: 3.556 on 1 and 2 DF,  p-value: 0.2
```

2.3 Correlations

Here is the regression of y on x1-x5 and the regression of y on x1-x5 with the x's demeaned.

How do the regression outputs compare??

Regression (again) of y on x1-5.

```r
xyd = read.csv("sim-reg-data.csv")
lmd = lm(y~.,xyd)
summary(lmd)
##
## Call:
## lm(formula = y ~ ., data = xyd)
##
## 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
```

Now demean the x's.

```r
xydd = xyd
for(i in 2:6) xydd[[i]] = xydd[[i]] - mean(xydd[[i]])
lmdd = lm(y~.,xydd)
summary(lmdd)
##
## Call:
## lm(formula = y ~ ., data = xydd)
##
## 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)  1.211044   0.022515  53.788  < 2e-16 ***
## 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
```

```r
mean(xyd$y)
## [1] 1.211044
```

Now let's look at all the correlations between y, x1-5, yhat, and e.

```r
fmat = cbind(xyd,lmd$fitted,lmd$residuals)
names(fmat)[c(7,8)] = c("yhat","e")
cor(fmat)
##               y           x1            x2            x3            x4
## y    1.00000000 8.462773e-02  8.654911e-02  4.802826e-01  4.801103e-01
## x1   0.08462773 1.000000e+00  2.637943e-02  3.653650e-02  3.620588e-02
## x2   0.08654911 2.637943e-02  1.000000e+00 -9.683881e-03 -9.707390e-03
## x3   0.48028258 3.653650e-02 -9.683881e-03  1.000000e+00  9.999510e-01
## x4   0.48011029 3.620588e-02 -9.707390e-03  9.999510e-01  1.000000e+00
## x5   0.48031024 3.661684e-02 -9.893077e-03  9.999518e-01  9.999054e-01
## yhat 0.49334026 1.715403e-01  1.754349e-01  9.735321e-01  9.731829e-01
## e    0.86983641 3.656747e-17 -2.264149e-17 -3.861346e-17 -4.599954e-17
##                 x5          yhat             e
## y     4.803102e-01  4.933403e-01  8.698364e-01
## x1    3.661684e-02  1.715403e-01  3.656747e-17
## x2   -9.893077e-03  1.754349e-01 -2.264149e-17
## x3    9.999518e-01  9.735321e-01 -3.861346e-17
## x4    9.999054e-01  9.731829e-01 -4.599954e-17
## x5    1.000000e+00  9.735882e-01 -8.156156e-17
## yhat  9.735882e-01  1.000000e+00 -2.176997e-17
## e    -8.156156e-17 -2.176997e-17  1.000000e+00
```

Note that the residuals (e) are uncorrelated with each x.

Why are the residuals uncorrelated with the fitted values (yhat)?

Square the correlation between y and yhat.
How does it compare with R^2 (see "Multiple R-squared") in the R regression summary.

3 Orthogonalized Regression

Now let's compare the regression of y on x1-x5 with the regression of y on x1-x4 and the xt5, where xt5 is the residual from regression x5 on x1-x4.

```r
lmfy = lm(y~x1+x2+x3+x4,xyd)
summary(lmfy)
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4, data = xyd)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4637 -0.6513  0.0286  0.6629  4.0287
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.010466   0.079596  -0.131 0.895406
## x1           0.025350   0.007684   3.299 0.000987 ***
## x2           0.035531   0.007738   4.592 4.67e-06 ***
## x3           6.942769   7.768001   0.894 0.371555
```

```
## x4          -5.054789   7.768765  -0.651 0.515344
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.007 on 1995 degrees of freedom
## Multiple R-squared:  0.2433, Adjusted R-squared:  0.2418
## F-statistic: 160.4 on 4 and 1995 DF,  p-value: < 2.2e-16
```

Regress x5 on x1-4 and then replace x5 with the residuals from this regression.

```
lmf5 = lm(x5~x1+x2+x3+x4,xyd)
e5 = lmf5$residuals
xyde = cbind(xyd[,1:5],e5)
lmfe = lm(y~.,xyde)
summary(lmfe)
##
## Call:
## lm(formula = y ~ ., data = xyde)
##
## 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) -0.010466   0.079613  -0.131 0.895429
## x1           0.025350   0.007685   3.298 0.000989 ***
## x2           0.035531   0.007740   4.591  4.7e-06 ***
## x3           6.942769   7.769706   0.894 0.371660
## x4          -5.054789   7.770469  -0.651 0.515436
## e5           2.767445   7.835586   0.353 0.723984
## ---
## 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
```

How do the coefficents from this last regression compare to the coefficients of x1-4 in the regression of y on x1-4?

How does the coefficient of e5 in this last regression compare to the coefficient of x5 in the regression of y on x1-5?

What is this number?

```
lmf = lm(y~.,xyd); shat = summary(lmf)$sigma
shat/sqrt(sum(e5^2))
## [1] 7.835586
```

Find this number on the regression of y on x1-5, that is, check that this number is indeed the standard error for the coefficient of x5.

What is the R^2 from the regression of x5 on x1-4?

Run the regression of y on just x5. How does the standard error for the coefficient of the coefficient for x5 compare to the standard error of the coefficient of x5 in the regression of y on x1-5?

Explain (in English) why they are so different.

4 ** Predictive Variance

Suppose we have training data (X, y) and x_f at which we wish to predict the future Y_f.

Then our usual prediction is

$$\hat{Y}_f = x_f \, \hat{\beta}$$

where $\hat{\beta}$ is the OLS (ordinary least squares) estimate of β obtained from the training data.

Obtain a nice matrix formula for

$$Var(E_f) = Var(Y_f - \hat{Y}_f)$$

Note in particular that $Var(E_f)$ incoporated both variation in (X, y) and Y_f.

5 ** R-squared

Show that in linear regression

$$cor(\hat{y}, y)^2 = \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2}$$

That is show that the square of the correlation between y and the fitted values is indeed the same as the usual formula for R-squared.