Exercises week 5

Michele De Vita

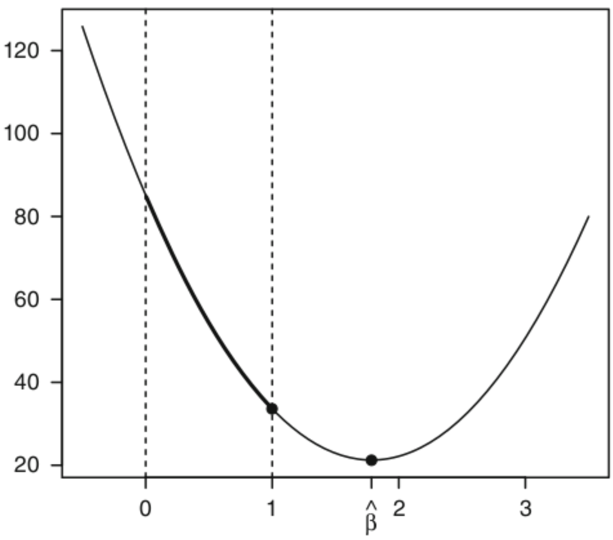
April 3, 2018

1.

$$i)\ \mathbf{C} = (0, 5, -7, 0), \qquad d = -2$$

$$ii)\ \mathbf{C} = \begin{pmatrix} 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \end{pmatrix}, \qquad d = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

2. This plot shows the SSE in function of β:



The F-test score is $\dfrac{SSE_{H_0} - SSE}{SSE}$ where $SSE = SSE(\hat{\beta})$ that is the y in the plot where $x = \hat{\beta}$ and $SSE_{H_0} = SSE(\beta = 1)$
The plot for the numerator of Wald test is the same of the previous because the numerator is the same except for a square difference instead of normal difference.
The main difference between F-test and Wald test is the denominator in the first is $\hat{\theta}$ while in the second is $var(\hat{\theta})$. There is also a relation between test statistics F and W: $W = rF$

3. The confidence interval for μ_0 is $\mathbf{x}_0'\hat{\beta} \pm t_{n-p}(1-\alpha/2)\hat{\sigma}(\mathbf{x}_0'(\mathbf{X}'\mathbf{X})^{-1}x_0)^{1/2}$ while for prediction is $\mathbf{x}_0'\hat{\beta} \pm t_{n-p}(1-\alpha/2)\hat{\sigma}(1+\mathbf{x}_0'(\mathbf{X}'\mathbf{X})^{-1}x_0)^{1/2}$. The

difference between the two formulas is the "1 +" near x_0 that become from the fact that the estimation try to give an confidence interval of a parameter , so with no variance, while the prediction try to estimate a aleatory variable with a expected value and a variance.

4. $f(x_0) = \mathbf{x}_0'\hat{\beta} \pm t_{n-2}(1-\alpha/2)\hat{\sigma}(1+\mathbf{x}_0'(\mathbf{X}'\mathbf{X})^{-1}x_0)^{1/2}$ where $X = \begin{pmatrix} 1 & x_{11} \\ \vdots & \vdots \\ 1 & x_{n1} \end{pmatrix}$ and $p = 2$.
   The length of interval is minimum when $\mathbf{x_0} = \mu$ because we can write the product $\mathbf{x_0'}(\mathbf{X'X})^{-\mathbf{1}}\mathbf{x_0}$ as $\dfrac{\sum(x_i - x_0)^2}{dev(x)}$ then the numerator $\sum(x_i - x_0)^2$ is minimized when $x_0 = \mu$

5. Increasing the number of parameters is not always a good idea: we can express the SPSE (expected squared prediction error as): $\underbrace{n\sigma^2}_{irreducible\ part} + \underbrace{|M|\sigma^2}_{Variance\ error} + \underbrace{\sum_{i=1}^{n}(u_{iM} - u_i)^2}_{BIAS^2}$

6. If we consider only $\mathbf{X_1}$ we have $\mathbf{X} = \mathbf{X_1}$ then

$$\begin{aligned} E\left[\tilde{\beta}_1\right] &= E\left[(\mathbf{X_1'X_1})^{-\mathbf{1}}\mathbf{X_1'y}\right] \\ &= (\mathbf{X_1'X_1})^{-\mathbf{1}}\mathbf{X_1'}E\left[\mathbf{y}\right] \\ &= (\mathbf{X_1'X_1})^{-\mathbf{1}}\mathbf{X_1'}(\mathbf{X_1}\beta_\mathbf{1} + \mathbf{X_2}\beta_\mathbf{2}) \\ &= \beta_1 + (\mathbf{X_1'X_1})^{-\mathbf{1}}\mathbf{X_1'X_2}\beta_\mathbf{2} \end{aligned}$$

   The bias vanish if $\mathbf{X_1'X_2} = \mathbf{0}$ that in statical words mean that $\mathbf{X_1}$ and $\mathbf{X_2}$ are uncorrelated or when $\beta_\mathbf{2}$ is irrelevant

7. If we omit a relevant covariate, we can have a decrease the variance of model such that the increase of $bias^2$ is overall convenient

8. $SPSE = \sum(y_{n+i} - \hat{y}_{iM}^2)$. See point 5

9.

$$\begin{aligned} E\left(SSE\right) &= E\left(\sum\left(y_i - \hat{y}_{iM}\right)^2\right) \\ &= E\left(\varepsilon'\varepsilon\right) \\ &= (n-p)\sigma^2 \\ &= E(SPSE) - 2|M|\sigma^2 \\ &= n\sigma^2 + |M|\sigma^2 - 2|M|\sigma^2 \\ &= n\sigma^2 - |M|\sigma^2 \\ &= (n-p)\sigma^2 \end{aligned}$$

   SSE underestimate the $SPSE$ because $\hat{SPSE} = \hat{SSE} + 2|M|\hat{\sigma}^2$.
   This bias is more severe for complex models because the bias is proportional to model complexity

10. $AIC = n\log(\hat{\sigma}^2) + 2(|M|+1)$
$BIC = n\log(\hat{\sigma}^2) + \log(n)(|M|+1)$.
Since $SPSE$ is not observable the two latter indicators are used to evaluate the fit of the model.
The difference between AIC and BIC is the term before $(|M|+1)$: in AIC is equal to 2 while in BIC is $\log(n)$. This mean that BIC have a major penalty when the model complexity increase in big dataset.
Next we can see the tables with the results:

n = 100	AIC	BIC
$l = -300, \|M\| = 5$	612	627.63
$l = -290, \|M\| = 9$	600	626.05

n = 200	AIC	BIC
$l = -300, \|M\| = 5$	612	631.79
$l = -290, \|M\| = 9$	600	632.98

Between $n = 100$ and $n = 200$ we can note the phenomenon described before with BIC, because with $n = 100$ there is a decrement of BIC between the two configurations while for $n = 200$ the BIC increase between the two configurations

11. $VIF_j = \dfrac{1}{1-R_j^2}$.
It measures the correlation between the covariate x_j with the other covariates. The greater is $R_j^2 \in [0,1]$ the greater is VIF_j. An empiric alert for variance is when $VIF_j > 10$

12. The ridge regression add a penalty term λ to the linear regression. The $\mathbf{PLS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta) + \lambda\beta'\beta$.
The ridge regression is biased respect to linear regression:
$E_{LS}(\hat{\beta}) = (\mathbf{X'X})^{-\mathbf{1}}\mathbf{X'X}\beta = \beta$
$E_{PLS}(\hat{\beta}) = (\mathbf{X'X} + \lambda\mathbf{X'X})^{-\mathbf{1}}\mathbf{X'X}\beta$ (when penalty $= 0$ $E_{PLS}(\hat{\beta}) = E_{LS}(\hat{\beta})$)
When λ increases the coefficients β tends to zero because $\lambda\mathbf{K} = (\mathbf{X'X} + \lambda\mathbf{K}) - \mathbf{X'X}$.
If we multiply two definite semipositive matrices $(\mathbf{X'X} + \lambda\mathbf{K})^{-\mathbf{1}}$ with $(\mathbf{X'X} + \lambda\mathbf{K}) - \mathbf{X'X}$ also the result is definite semipositive that is $\mathbf{I_p} - (\mathbf{X'X} + \lambda\mathbf{K})^{-\mathbf{1}}\mathbf{X'X}$.
This imply that $(\mathbf{X'X} + \lambda\mathbf{K})^{-\mathbf{1}}\mathbf{X'X}$ have the diagonal elements between $[0,1]$ and this term appears when we express $\hat{\beta}_{PLS}$ in function of $\hat{\beta}_{LS}$:
$\hat{\beta}_{PLS} = (\mathbf{X'X} + \lambda\mathbf{K})^{-\mathbf{1}}\mathbf{X'X}\beta_{\mathbf{LS}}$
The covariance matrix of $\hat{\beta}_{PLS}$ for the same reason is lower than variance of $\hat{\beta}_{LS}$ so in some cases can be convenient use a ridge regression when the bias error is less relevant than variance error

Stata code

```
clear
set seed 1032299
set obs 150
gen x1 = runiform()
gen x2 = x1 + runiform()
```

```
gen x3 = runiform()
gen y = rnormal(-1 + 0.3 * x1 + 0.2*x3, 0.2^2)
regress y x1 x2 x3
regress y x1 x3
```