Econoemtrics-ps3

Wei Ye

2/28/2022

a:

The qfunction we modify as below:

```
qfunction <- function(y,x,theta){
  N=length(y) #Find sample size
  #note that x should be N*K and theta should N*1
  m=x %*% theta#Define the function m(), %*% matrix mulplication, transpose of theta
  Q=(1/N)*sum((y-m)^2)# This is our objective function
  out=Q
}
```

b:

Since this question requires us to derive Hessian matrix: it's just the qderivfun we coded in class, and the analytic form as below:

First,set up the q:

$$q = \frac{(y - m(x, \theta_0))^2}{2} = \frac{(y - \theta_{01} - \theta_{02} priGPA - \theta_{03} ACT)^2}{2}$$

Thus, $\frac{\partial q}{\partial \theta_{01}} = (y - m(x, \theta_0)) \cdot (-1)$, $\frac{\partial q}{\partial \theta_{02}} = (\cdot)(-priGPA)$, and $\frac{\partial q}{\partial \theta_{03}} = (\cdot)(-ACT)$. Therefore, we can derive the Hessian Matrix as:

$$H_i = \begin{bmatrix} 1 & priGPA_i & ACT_i \\ priGPA_i & priGPA_i^2 & priGPA_i \cdot ACT_i \\ ACT_i & ACT \cdot priGPA_i & ACT_i^2 \end{bmatrix}$$

```
qderivfun<-function(y,x,thetahat){
N=length(y) #sample size
k1=dim(as.matrix(x)) #find number of x-variables
k=k1[2]+1 #add to x column to adjust for vector of ones
derivstore=matrix(0,N,k)
h=.Machine$double.eps^(1/3) #set approximation error
x=as.matrix(x) #declare as matrix
for (j in 1:k){
z=rep(0,k)
z[j]=h
for (i in 1:N){
Qph=qfunction(y[i],cbind(1,t(x[i,])),thetahat+z)
Qmh=qfunction(y[i],cbind(1,t(x[i,])),thetahat-z)
qderiv=(Qph-Qmh)/(2*h)
derivstore[i,j]=qderiv
}
```

```
}
B0hat=(1/N)*(t(derivstore)%*%derivstore) #B0hat from AVAR
ssum=colSums(derivstore)
return(list(derivstore=derivstore,ssum=ssum,B0hat=B0hat))
}
```

##c: For computing sum of hessian:it's the file of **qderivfun2.R**, and the analytical part as below:

$$\sum_{i=1}^{N} H_i = \begin{bmatrix} N & \sum_{i=1}^{N} priGPA_i & \sum_{i=1}^{N} ACT_i \\ \sum_{i=1}^{N} priGPA_i & \sum_{i=1}^{N} priGPA_i^2 & \sum_{i=1}^{N} priGPA_i \cdot ACT_i \\ \sum_{i=1}^{N} ACT_i & \sum_{i=1}^{N} ACT_i \cdot priGPA_i & \sum_{i=1}^{N} ACT_i^2 \end{bmatrix}$$

```
qderivfun2 <- function(y,x,thetahat){
  k=dim(as.matrix(x))#dimension of x-variable
  p=k[2]+1 #set dimension of theta space
  N=length(y)
  derivs=vector(,N)#pre-allocate space
  H=matrix(0,p,p)#Pre-allocate space of hessian matrix
  h=.Machine$double.eps^(1/3)
  for(i in 1:p){
    for (j in 1:p) {
      z=rep(0,p)#create scalar to approximate derivative at each p
      z[j]=h
      theta1=thetahat+z
      theta2=thetahat-z
      qout1=qderivfun(y,x,theta1)
      q1=qout1$derivstor[,i]
      qout2=qderivfun(y,x,theta2)
      q2=qout2$derivstor[,i]
      d=(q1-q2)/(2*h)# Central difference
      H[i,j]=sum(d)#store the hessian matrix
    }
  }
  A0hat=(1/N)*H
  return(list(H=H,A0hat=A0hat))
}
```

##d: Compute the analytical expression for A_0: $A_0 = E[H(w,\hat{\theta})] = \frac{1}{N}\sum_{i=1}^{N} H(w_i,\hat{\theta})$.

##e: Estimate the populaton model under consideration:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
attend<-read_csv("attend.csv")
```

```
## Rows: 680 Columns: 11
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (11): attend, termgpa, priGPA, ACT, final, atndrte, hwrte, frosh, soph, ...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
attach(attend)
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     attend
```

```r
nlsout=nls(termgpa~(b0+b1*priGPA+b2*ACT),start=list(b0=1,b1=0.04321,b2=.9))
summary(nlsout)
```

```
##
## Formula: termgpa ~ (b0 + b1 * priGPA + b2 * ACT)
##
## Parameters:
##    Estimate Std. Error t value Pr(>|t|)
## b0 0.257753   0.150849   1.709    0.088 .
## b1 0.875274   0.042065  20.808   <2e-16 ***
## b2 0.003514   0.006564   0.535    0.593
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5585 on 677 degrees of freedom
##
## Number of iterations to convergence: 1
## Achieved convergence tolerance: 1.036e-07
```

f: Verify the result with the code qderivfun2.R:

```r
source('~/Dropbox/My Mac (Wei's MacBook Air)/Downloads/PhD-Coursework/22Spring/Econometrics/My_solution
source('~/Dropbox/My Mac (Wei's MacBook Air)/Downloads/PhD-Coursework/22Spring/Econometrics/My_solution
 source('~/Dropbox/My Mac (Wei's MacBook Air)/Downloads/PhD-Coursework/22Spring/Econometrics/My_solutio

thetahat=cbind(0.257753,0.875274,0.003514)
X=cbind(priGPA,ACT)
out=qderivfun(termgpa,X,t(thetahat))
out1=qderivfun2(termgpa,X,t(thetahat))
out1
```

```
## $H
##           [,1]      [,2]      [,3]
## [1,]  1360.000  3518.014  30614.00
## [2,]  3518.014  9503.247  80105.03
## [3,] 30614.000 80105.032 705678.00
##
## $A0hat
##          [,1]      [,2]       [,3]
## [1,]  2.00000   5.17355   45.02059
## [2,]  5.17355  13.97536  117.80152
## [3,] 45.02059 117.80152 1037.76176
```

##g: Calculate the estimated variance-covariance matrix $A\hat{var}(\hat{\theta})$:

```
avartheta=solve(out1$A0hat)%*% out$B0hat%*% solve(out1$A0hat)/length(termgpa)
avartheta
```

```
##                [,1]          [,2]          [,3]
## [1,]  0.0214344252 -0.0027355384 -5.955213e-04
## [2,] -0.0027355384  0.0022330729 -1.567609e-04
## [3,] -0.0005955213 -0.0001567609  4.598071e-05
```

##h: Test the null hypothesis:

```
vartheta=diag(avartheta)
se=vartheta^(1/2)
se
```

```
## [1] 0.146405004 0.047255400 0.006780908
```

```
t_stat=thetahat/se
t_stat
```

```
##          [,1]    [,2]      [,3]
## [1,] 1.760548 18.5222 0.5182197
```

Thus, from the results we derived above, we can't accept the alternative condition for $\theta_{01}, \theta_{03} \neq 0$, but another θ, we accept the alternative conditions that they are not equal to 0.