Comparison With F-Test

Zheng Yuan

2019/9/16

```
library(simstudy)
library(dplyr)
library(caret)
library(lmtest)
library(knitr)
```

Evaluate new test statistic

```
#### leave one out cross validation prediction error function using trick

loocv=function(fit){

  h=lm.influence(fit)$h
  mean((residuals(fit)/(1-h))^2)

}

newts <- function(model1,model2) {
  (length(model1$model$y) * loocv(model1)
   -2*model1$rank)- (length(model2$model$y) * loocv(model2)-2*model2$rank)
}
```

Evaluate the F-statistic

Assuming $M_\alpha \subset M_\lambda$ and $dim(M_\lambda) - dim(M_\alpha) = d,\ dim(M_\lambda) = p,$

$$F_{d,n-p-1} = \frac{n-p-1}{\hat{\sigma}^2_\lambda}(\hat{\sigma}^2_\alpha - \hat{\sigma}^2_\lambda)\frac{1}{d}$$

```
mse <- function(object) {
  mean(residuals(object)^2)
}

fts <- function(model1,model2) {
  (length(model1$model$y)-model2$rank-1)*
    (mse(model1)/mse(model2)-1)/(model2$rank-model1$rank)
}
```

Fit each model with sample size n=1000 and scale the true model with $\sqrt{(n)}$

```
n = 1000

def <- defData(varname = "x1", dist="uniform",formula = "10;20")  ## x1 is from unifrom distribution
```

```
def <- defData(def,varname = "x2", dist="uniform",formula = "0;3")

def <- defData(def,varname = "x3", dist="uniform",formula = "0;5")

def <- defData(def,varname = "x4", dist="uniform",formula = "5;10")

def <- defData(def, varname = "y", formula = "3/sqrt(1000)+2/sqrt(1000)*x1+5/sqrt(1000)*x2+3/sqrt(1000)*

## The true linear model is y = 3/sqrt(1000)+2/sqrt(1000)*x1+5/sqrt(1000)*x2+3/sqrt(1000)*x3+1.5/sqrt(1

dt <- genData(n, def) ##generate dataset n=1000

dt <- dt%>%select(y,x1,x2,x3,x4)

fit1 <- lm(y~ x1, data = dt)

fit2 <- lm(y ~ x1+x2, data = dt)

fit3 <- lm(y ~ x1+x2+x3, data = dt)

fit4 <- lm(y ~ x1+x2+x3+x4, data = dt)
```

Value of each test statistic

```
ts1 = newts(fit1,fit4)

ts2 = newts(fit2,fit4)

ts3 = newts(fit3,fit4)


fts1 = fts(fit1,fit4)

fts2 = fts(fit2,fit4)

fts3 = fts(fit3,fit4)

ts1;fts1;ts2;fts2;ts3;fts3
```

```
## [1] 43.59044
## [1] 14.80135
## [1] 23.53616
## [1] 11.95637
## [1] 11.16538
## [1] 11.22342
```

Corresponding P-values

```
1-pchisq(ts1,3)
```

```
## [1] 1.843877e-09
```

```
1-pf(fts1,3,995)
```

```
## [1] 1.961954e-09
```

```
1-pchisq(ts2,2)
```

```
## [1] 7.747954e-06
```

```
1-pf(fts2,2,995)
```

```
## [1] 7.393149e-06
```

```
1-pchisq(ts3,1)
```

```
## [1] 0.0008333783
```

```
1-pf(fts3,1,995)
```

```
## [1] 0.0008380604
```

The p-value for new test statistic is slightly samller than that of f-test in some cases.

Compute the power of two tests

Here, for each null hypothesis, we repeat two testing procedures 200 or 1000 times and in each time we record whether they reject the null hypothesis or not. Then $power = \frac{number\ of\ rejection\ times}{number\ of\ simulation\ times}$

model 1 vs model 4

```
md1<-c()

md2<-c()

for (i in 1:200){

n = 1000

def <- defData(varname = "x1", dist="uniform",formula = "10;20")  ## x1 is from unifrom distribution

def <- defData(def,varname = "x2", dist="uniform",formula = "0;3")

def <- defData(def,varname = "x3", dist="uniform",formula = "0;5")

def <- defData(def,varname = "x4", dist="uniform",formula = "5;10")

def <- defData(def, varname = "y", formula = "3/sqrt(1000)+2/sqrt(1000)*x1+5/sqrt(1000)*x2+3/sqrt(1000)*

## The true linear model is y = 3/sqrt(1000)+2/sqrt(1000)*x1+5/sqrt(1000)*x2+3/sqrt(1000)*x3+1.5/sqrt(1

dt <- genData(n, def) ##generate dataset n=1000
```

```
dt <- dt%>%select(y,x1,x2,x3,x4)

fit1 <- lm(y~ x1, data = dt)

fit4 <- lm(y ~ x1+x2+x3+x4, data = dt)


md1<-c(md1,ifelse((1-pchisq(newts(fit1,fit4),2))<0.05,1,0))

md2<-c(md2,ifelse((1-pf(fts(fit1,fit4),3,995))<0.05,1,0))


}

sum(md1)/200
```

```
## [1] 1
```

```
sum(md2)/200
```

```
## [1] 1
```

model 2 vs model 4

```
md1<-c()

md2<-c()

for (i in 1:1000){

n = 1000

def <- defData(varname = "x1", dist="uniform",formula = "10;20")  ## x1 is from unifrom distribution

def <- defData(def,varname = "x2", dist="uniform",formula = "0;3")

def <- defData(def,varname = "x3", dist="uniform",formula = "0;5")

def <- defData(def,varname = "x4", dist="uniform",formula = "5;10")

def <- defData(def, varname = "y", formula = "3/sqrt(1000)+2/sqrt(1000)*x1+5/sqrt(1000)*x2+3/sqrt(1000)*

## The true linear model is y = 3/sqrt(1000)+2/sqrt(1000)*x1+5/sqrt(1000)*x2+3/sqrt(1000)*x3+1.5/sqrt(1

dt <- genData(n, def) ##generate dataset n=1000

dt <- dt%>%select(y,x1,x2,x3,x4)

fit2 <- lm(y~ x1+x2, data = dt)

fit4 <- lm(y ~ x1+x2+x3+x4, data = dt)
```

```
md1<-c(md1,ifelse((1-pchisq(newts(fit2,fit4),2))<0.05,1,0))

md2<-c(md2,ifelse((1-pf(fts(fit2,fit4),2,995))<0.05,1,0))


}

sum(md1)/1000
```

```
## [1] 0.995
```

```
sum(md2)/1000
```

```
## [1] 0.995
```

model 3 vs model 4

```
md1<-c()

md2<-c()

for (i in 1:1000){

n = 1000

def <- defData(varname = "x1", dist="uniform",formula = "10;20")  ## x1 is from unifrom distribution

def <- defData(def,varname = "x2", dist="uniform",formula = "0;3")

def <- defData(def,varname = "x3", dist="uniform",formula = "0;5")

def <- defData(def,varname = "x4", dist="uniform",formula = "5;10")

def <- defData(def, varname = "y", formula = "3/sqrt(1000)+2/sqrt(1000)*x1+5/sqrt(1000)*x2+3/sqrt(1000)

## The true linear model is y = 3/sqrt(1000)+2/sqrt(1000)*x1+5/sqrt(1000)*x2+3/sqrt(1000)*x3+1.5/sqrt(1

dt <- genData(n, def) ##generate dataset n=1000

dt <- dt%>%select(y,x1,x2,x3,x4)

fit3 <- lm(y~ x1+x2+x3, data = dt)

fit4 <- lm(y ~ x1+x2+x3+x4, data = dt)


md1<-c(md1,ifelse((1-pchisq(newts(fit3,fit4),1))<0.05,1,0))

md2<-c(md2,ifelse((1-pf(fts(fit3,fit4),1,995))<0.05,1,0))


}
```

```r
sum(md1)/1000
```

```
## [1] 0.595
```

```r
sum(md2)/1000
```

```
## [1] 0.593
```

From the result above, we can see that for each null hypothesis, our new test procedure could enjoy a little higher power (although not obviously higher) than traditional f-test.