Comparison of Two Tests with Different Sample Sizes

Zheng Yuan

2019/9/30

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

This simulation report continues the comparison between the powers of the new test and the traditional F-test. Different sample sizes are set, specifically, $n = 50, 100, 300, 500$. For each samples size, model with dimension $d = 1, 2, 3, 4$ are fitted and the full model whose coeffients have been scaled by $\sqrt{n}$ is always set as true model. Then for each null hypothesis, during 1000 times simulation, record the times for which two hypothesis-testing procedures rejects the null hypothesis respectively. Then we get estimations of the powers for two tests.

From the result, under each sample size, $n = 50, 100, 300, 500, 1000$, the new test procedure always enjoys equal or a little bit higher power than the traditional F-test, no matter what the null hypothesis is, i.e., no matter how small the difference in dimensions between null hypothesis and alternative hypothesis is.

New test statistic

$$NTS = (\frac{n}{\hat{\sigma^2_\lambda}}\hat{\Gamma}_{\alpha,n} - 2d_\alpha) - (\frac{n}{\hat{\sigma^2_\lambda}}\hat{\Gamma}_{\lambda,n} - 2d_\lambda)$$

, where $\hat{\sigma}^2$ is the mle for σ^2 under M_λ.

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

loocv=function(fit){

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

}
#### New Test Statistic
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 $dim(M_\lambda) - dim(M_\alpha) = d,\ dim(M_\lambda) = p$,

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

```r
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=300 and scale the true model with $\sqrt{(n)}$

```r
n = 300

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(300)+2/sqrt(300)*x1+5/sqrt(300)*x2+3/sqrt(300)*x3+

dt <- genData(n, def) ##generate dataset n=300

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)
```

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

```r
md1<-c()

md2<-c()

for (i in 1:200){

n = 300
```

```
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(300)+2/sqrt(300)*x1+5/sqrt(300)*x2+3/sqrt(300)*x3+

dt <- genData(n, def) ##generate dataset n=300

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)


md1<-c(md1,ifelse((1-pchisq(newts(fit1,fit4),3))<0.05,1,0))

md2<-c(md2,ifelse((1-pf(fts(fit1,fit4),3,295))<0.05,1,0)) ##n-p-1=295


}

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 = 300

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(300)+2/sqrt(300)*x1+5/sqrt(300)*x2+3/sqrt(300)*x3+

dt <- genData(n, def) ##generate dataset n=300

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)


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

md2<-c(md2,ifelse((1-pf(fts(fit2,fit4),2,295))<0.05,1,0))


}

sum(md1)/1000
```

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

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

```
## [1] 0.993
```

model 3 vs model 4

```
md1<-c()

md2<-c()

for (i in 1:1000){

n = 300

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(300)+2/sqrt(300)*x1+5/sqrt(300)*x2+3/sqrt(300)*x3+

dt <- genData(n, def) ##generate dataset n=300
```

```
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)

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

md2<-c(md2,ifelse((1-pf(fts(fit3,fit4),1,295))<0.05,1,0))


}

sum(md1)/1000
```

```
## [1] 0.572
```

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

```
## [1] 0.556
```

Sample Size = 50

model 1 vs model 4

```
md1<-c()

md2<-c()

for (i in 1:200){

n = 50

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(50)+2/sqrt(50)*x1+5/sqrt(50)*x2+3/sqrt(50)*x3+1.5/

dt <- genData(n, def) ##generate dataset n=50

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)


md1<-c(md1,ifelse((1-pchisq(newts(fit1,fit4),3))<0.05,1,0))

md2<-c(md2,ifelse((1-pf(fts(fit1,fit4),3,45))<0.05,1,0)) ##n-p-1=45


}

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 = 50

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(50)+2/sqrt(50)*x1+5/sqrt(50)*x2+3/sqrt(50)*x3+1.5/

dt <- genData(n, def) ##generate dataset n=50

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)
```

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

md2<-c(md2,ifelse((1-pf(fts(fit2,fit4),2,45))<0.05,1,0))


}

sum(md1)/1000
```

```
## [1] 0.987
```

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

```
## [1] 0.979
```

model 3 vs model 4

```
md1<-c()

md2<-c()

for (i in 1:1000){

n = 50

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(50)+2/sqrt(50)*x1+5/sqrt(50)*x2+3/sqrt(50)*x3+1.5/

dt <- genData(n, def) ##generate dataset n=50

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)

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

md2<-c(md2,ifelse((1-pf(fts(fit3,fit4),1,45))<0.05,1,0))
```

```
}

sum(md1)/1000
```

```
## [1] 0.547
```

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

```
## [1] 0.501
```

Sample Size = 500

model 1 vs model 4

```
md1<-c()

md2<-c()

for (i in 1:200){

n = 500

def <- defData(varname = "x1", dist="uniform",formula = "10;20")

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(500)+2/sqrt(500)*x1+5/sqrt(500)*x2+3/sqrt(500)*x3+

dt <- genData(n, def) ##generate dataset n=500

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)


md1<-c(md1,ifelse((1-pchisq(newts(fit1,fit4),3))<0.05,1,0))

md2<-c(md2,ifelse((1-pf(fts(fit1,fit4),3,495))<0.05,1,0)) ## n-p-1=495


}

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 = 500

def <- defData(varname = "x1", dist="uniform",formula = "10;20")

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(500)+2/sqrt(500)*x1+5/sqrt(500)*x2+3/sqrt(500)*x3+

dt <- genData(n, def) ##generate dataset n=50

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)


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

md2<-c(md2,ifelse((1-pf(fts(fit2,fit4),2,495))<0.05,1,0))


}
```

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

```
## [1] 0.994
```

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

```
## [1] 0.994
```

model 3 vs model 4

```
md1<-c()

md2<-c()

for (i in 1:1000){

n = 500

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(500)+2/sqrt(500)*x1+5/sqrt(500)*x2+3/sqrt(500)*x3+

dt <- genData(n, def) ##generate dataset n=500

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)

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

md2<-c(md2,ifelse((1-pf(fts(fit3,fit4),1,495))<0.05,1,0))


}

sum(md1)/1000
```

```
## [1] 0.584
```

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

```
## [1] 0.579
```

Sample Size = 100

model 1 vs model 4

```
md1<-c()
```

```
md2<-c()

for (i in 1:200){

n = 100

def <- defData(varname = "x1", dist="uniform",formula = "10;20")

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(100)+2/sqrt(100)*x1+5/sqrt(100)*x2+3/sqrt(100)*x3+

dt <- genData(n, def) ##generate dataset n=100

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)


md1<-c(md1,ifelse((1-pchisq(newts(fit1,fit4),3))<0.05,1,0))

md2<-c(md2,ifelse((1-pf(fts(fit1,fit4),3,95))<0.05,1,0)) ## n-p-1=95


}

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 = 100
```

```
def <- defData(varname = "x1", dist="uniform",formula = "10;20")

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(100)+2/sqrt(100)*x1+5/sqrt(100)*x2+3/sqrt(100)*x3+

dt <- genData(n, def) ##generate dataset n=100

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)


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

md2<-c(md2,ifelse((1-pf(fts(fit2,fit4),2,95))<0.05,1,0)) ## n-p-1=95

}

sum(md1)/1000
```

```
## [1] 0.994
```

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

```
## [1] 0.991
```

model 3 vs model 4

```
md1<-c()

md2<-c()

for (i in 1:1000){

n = 100

def <- defData(varname = "x1", dist="uniform",formula = "10;20")

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

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

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

def <- defData(def, varname = "y", formula = "3/sqrt(100)+2/sqrt(100)*x1+5/sqrt(100)*x2+3/sqrt(100)*x3+

dt <- genData(n, def) ##generate dataset n=100

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)


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

md2<-c(md2,ifelse((1-pf(fts(fit3,fit4),1,95))<0.05,1,0)) ## n-p-1=95


}

sum(md1)/1000
```

```
## [1] 0.578
```

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

```
## [1] 0.542
```