

# Lab

Peijun Xu

9/22/2023

```
setwd('/Users/ivanxu/Desktop/PhD/Robust MR-SPI CI')
library(devtools)
```

```
## Loading required package: usethis
```

```
devtools::install_github("MinhaoYaooo/MR-SPI", force = TRUE)
```

```
## Downloading GitHub repo MinhaoYaooo/MR-SPI@HEAD
```

```
##
```

```
## -- R CMD build -----
```

```
##   checking for file '/private/var/folders/cf/cl9mxhwx3m75m11pmlyf49zc0000gn/T/RtmpU4gdhZ/remotes4
```

```
## - preparing 'MR.SPI':
```

```
##   checking DESCRIPTION meta-information ... v checking DESCRIPTION meta-information
```

```
## - checking for LF line-endings in source and make files and shell scripts
```

```
## - checking for empty or unneeded directories
```

```
##   Omitted 'LazyData' from DESCRIPTION
```

```
## - building 'MR.SPI_0.1.0.tar.gz'
```

```
##
```

```
##
```

```
library(CVXR)
```

```
## Warning in .recacheSubclasses(def@className, def, env): undefined subclass
```

```
## "pcorMatrix" of class "ConstVal"; definition not updated
```

```
## Warning in .recacheSubclasses(def@className, def, env): undefined subclass
```

```
## "pcorMatrix" of class "ConstValListORExpr"; definition not updated
```

```
## Warning in .recacheSubclasses(def@className, def, env): undefined subclass
```

```
## "pcorMatrix" of class "ConstValORExpr"; definition not updated
```

```
## Warning in .recacheSubclasses(def@className, def, env): undefined subclass
```

```
## "pcorMatrix" of class "ConstValORNULL"; definition not updated
```

```
##
```

```
## Attaching package: 'CVXR'
```

```
## The following object is masked from 'package:stats':
```

```
##
```

```
##   power
```

```
library(MR.SPI)
```

```
library(bigstatsr)
```

```
library(igraph)
```

```
##
```

```
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##   decompose, spectrum
## The following object is masked from 'package:base':
##
##   union
library(intervals)

##
## Attaching package: 'intervals'
## The following objects are masked from 'package:igraph':
##
##   clusters, contract
## The following object is masked from 'package:CVXR':
##
##   size
library(matrixStats)
library(svMisc)

##
## Attaching package: 'svMisc'
## The following object is masked from 'package:utils':
##
##   ?
options(warn=0)
```

Step1: Find the smallest lower bound and largest upper bound by only using half the selected IV.

Short-term: iterate selected IV value Long-term: optimization algorithm

```
robust_ci_temp <- function(gamma, Gamma, se_gamma, se_Gamma, alpha=0.05,n1=10000,n2=20000) {
  pz = length(Gamma)

  V_gamma <- matrix(nrow = pz, ncol = pz)
  V_Gamma <- matrix(nrow = pz, ncol = pz)

  for (i in 1:pz) {
    for (j in i:pz) {
      if(i==j){
        V_gamma[i,i] <- se_gamma[i]^2
        V_Gamma[i,i] <- se_Gamma[i]^2
      }else{
        V_gamma[i,j] <- V_gamma[j,i] <- gamma[i]*gamma[j] / n1
        V_Gamma[i,j] <- V_Gamma[j,i] <- Gamma[i]*Gamma[j] / n2
      }
    }
  }
  } # Construct V_gamma_hat and V_Gamma_hat
L1 <- 100
U1 <- -100
L2 <- 100
U2 <- -100
comb <- combn(c(1:pz),pz%/%2+1)
```



```

u = robust_u = robust_strict_u0 = robust_loose_u0 = robust_u1=0
pi = 0.25*pi
cnt = robust_cnt = robust_cnt0_loose = robust_cnt0_strict = robust_cnt1=0
time.default = time.robust = time.robust0 = 0
max_iter <- 1000
pb <- txtProgressBar(min = 0, max = max_iter, style = 3)
for (i in 1:max_iter){
  setTxtProgressBar(pb, i)
  result <- generate_example(beta=0.5, pi=pi)
  t1 <- Sys.time()
  mr.spi.default <- MR.SPI(result$gammaHat, result$GammaHat, result$se_gamma, result$se_Gamma, result$se_gamma0, result$se_Gamma0)
  # print(c(mr.spi.robust$ci[[1]][1],mr.spi.robust$ci[[2]]))
  t2 <- Sys.time()
  suppressWarnings(mr.spi.robust <-MR.SPI(result$gammaHat, result$GammaHat, result$se_gamma, result$se_Gamma, result$se_gamma0, result$se_Gamma0))
  t3 <- Sys.time()
  robust_ci_0 <- robust_ci_temp(gamma = result$gammaHat[mr.spi.default$VHat[[1]]],
                                Gamma = result$GammaHat[mr.spi.default$VHat[[1]]],
                                se_gamma = result$se_gamma,
                                se_Gamma = result$se_Gamma,
                                )

  t4<-Sys.time()
  if (mr.spi.default$ci[[1]][1]>0.5 | mr.spi.default$ci[[1]][2]<0.5){
    cnt <- cnt + 1
  }
  if (mr.spi.robust$ci[[1]][1]>0.5 | mr.spi.robust$ci[[2]]<0.5){
    robust_cnt <- robust_cnt + 1
  }
  if (robust_ci_0$loose_ci[1]>0.5 | robust_ci_0$loose_ci[2]<0.5){
    robust_cnt0_loose <- robust_cnt0_loose + 1
  }
  if (robust_ci_0$strict_ci[1]>0.5 | robust_ci_0$strict_ci[2]<0.5){
    robust_cnt0_strict <- robust_cnt0_strict + 1
  }
  robust_l1_t <- max(mr.spi.robust$ci[[1]][1],robust_ci_0$strict_ci[1])
  robust_u1_t <- min(mr.spi.robust$ci[[2]],robust_ci_0$strict_ci[2])
  if (robust_l1_t>0.5 | robust_u1_t<0.5){
    robust_cnt1 <- robust_cnt1 + 1
  }
  time.default <- time.default + (t2 - t1)
  time.robust <- time.robust + (t3 - t2)
  time.robust0 <- time.robust0 + (t4 - t3)

  l <- l+mr.spi.default$ci[[1]][1]
  u <- u+mr.spi.default$ci[[1]][2]

  robust_l <- robust_l+mr.spi.robust$ci[[1]][1]
  robust_u <- robust_u+mr.spi.robust$ci[[2]]

  robust_loose_l0 <- robust_loose_l0+robust_ci_0$loose_ci[1]
  robust_loose_u0 <- robust_loose_u0+robust_ci_0$loose_ci[2]

  robust_strict_l0 <- robust_strict_l0+robust_ci_0$strict_ci[1]
  robust_strict_u0 <- robust_strict_u0+robust_ci_0$strict_ci[2]

```

```

    robust_l1 <- robust_l1+robust_l1_t
    robust_u1 <- robust_u1+robust_u1_t
  }
  close(pb)

sim.result4 <- rbind(sim.result4, data.frame(
  #beta = beta,
  # tau = tau,
  #gamma = gamma,
  coverage = 1-cnt/max_iter,
  ci_length = u/max_iter-l/max_iter,
  robust_ci_length = robust_u/max_iter-robust_l/max_iter,
  robust_loose_ci0_length = robust_loose_u0/max_iter-robust_loose_l0/max_iter,
  robust_strict_ci0_length = robust_strict_u0/max_iter-robust_strict_l0/max_iter,
  robust_ci1_length = robust_u1/max_iter-robust_l1/max_iter,
  robust_coverage = 1-robust_cnt/max_iter,
  robust_coverage0_loose = 1-robust_cnt0_loose/max_iter,
  robust_coverage0_strict = 1-robust_cnt0_strict/max_iter,
  robust_coverage1 = 1-robust_cnt1/max_iter,
  time.default <- time.default/max_iter,
  time.robust <- time.robust/max_iter,
  time.robust0 <- time.robust0/max_iter

))
}

## [1] 0 0 0 0 0 0 1 1 1 1
## |

```

The new robust CI guarantees the CI length is smaller than the robust CI found using MR-SPI and simulation results show it has over 95% coverage.

```

sim.result <- data.frame(
  default_ci_length = sim.result4$ci_length,
  robust_ci_length = sim.result4$robust_ci_length,
  new_robust_ci_length = sim.result4$robust_ci1_length,
  default_coverage = sim.result4$coverage,
  robust_coverage = sim.result4$robust_coverage,
  new_robust_coverage = sim.result4$robust_coverage1
)
rownames(sim.result) <- c("(0,0,0,0,0,0,1,1,1,1)",
  "(0, 0,0,0,1,1,1,-1,-1,-1)",
  "(0,0,0,0,0,0,1,1,0.25,0.25)",
  "(0,0,0,0,1,1,0.25,-1,-1,-0.25)")
print(sim.result)

##                                default_ci_length robust_ci_length
## (0,0,0,0,0,0,1,1,1,1)                0.09826303      0.2025967
## (0, 0,0,0,1,1,1,-1,-1,-1)            0.12885330      1.5808239

```

## (0,0,0,0,0,0,1,1,0.25,0.25)	0.09733507	0.2000011
## (0,0,0,0,1,1,0.25,-1,-1,-0.25)	0.12477270	0.2544223
##	new_robust_ci_length	default_coverage
## (0,0,0,0,0,0,1,1,1,1)	0.1920005	0.950
## (0, 0,0,0,1,1,1,-1,-1,-1)	0.2263439	0.939
## (0,0,0,0,0,0,1,1,0.25,0.25)	0.1892054	0.924
## (0,0,0,0,1,1,0.25,-1,-1,-0.25)	0.2204170	0.748
##	robust_coverage	new_robust_coverage
## (0,0,0,0,0,0,1,1,1,1)	0.999	0.999
## (0, 0,0,0,1,1,1,-1,-1,-1)	0.992	0.979
## (0,0,0,0,0,0,1,1,0.25,0.25)	1.000	1.000
## (0,0,0,0,1,1,0.25,-1,-1,-0.25)	0.997	0.970