

Markdown에 사용된 R code와 code에 대한 설명은 CodeExample/Example\_sourcedetectionsimul.R 파일에서 확인하실 수 있습니다. 이 예제에서는 Simulation model in Section 3.1 with n0=50, p=30, q=20, r=5, h=5, nk=50 을고려합니다.

### 패키지 설치

```
In [8]: install.packages('RSpectra')
install.packages('corpcor')

install.packages('foreach')
install.packages('doParallel')

install.packages('splitTools')

Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

### 함수 불러오기

```
In [12]: source("/content/Functions_nr-basedmethods_simul.R")
```

### 데이터 생성

- Target data의 100개의 replicates 만들기

```
In [3]: targestset <- generator_target_simulsourcedetection(rep=100,n=50,p=30,q=20,r=5, corx=0.5,cory=0)
```

- Source datasets의 100개의 replicates 만들기

```
In [10]: Sources_numsources2_30level <- generator_source_simulsourcedetection(rep=100, n=50,p=30,q=20,
                                     rank_source=5,B=targestset$B,
                                     corx=0.5,cory=0,
                                     h=30, numsource=2)

Sources_numsources3_20level <- generator_source_simulsourcedetection(rep=100, n=50,p=30,q=20,
                                     rank_source=5,B=targestset$B,
                                     corx=0.5,cory=0,
                                     h=20, numsource=3)

Informativesources_numsources3_h5 <-generator_source_simulsourcedetection(rep=100, n=50,p=30,q=20,
                                     rank_source=5,B=targestset$B,
                                     corx=0.5,cory=0,
                                     h=5, numsource=3)

Sourceset_X <- Sourceset_Y <- list()
for(ss in 1:100){
  Sourceset_X_ss <- list()
  Sourceset_Y_ss <- list()
  ##### informative sources #####
  for(aa in 1:3){
    Sourceset_X_ss[[aa]] <- Informativesources_numsources3_h5$auxXlist_list[[ss]][[aa]]
    Sourceset_Y_ss[[aa]] <- Informativesources_numsources3_h5$auxYlist_list[[ss]][[aa]]
  }
  ##### h=20 #####
  for(aa in 4:6){
    Sourceset_X_ss[[aa]] <- Sources_numsources3_20level$auxXlist_list[[ss]][[aa-3]]
    Sourceset_Y_ss[[aa]] <- Sources_numsources3_20level$auxYlist_list[[ss]][[aa-3]]
  }
  ##### h=30 #####
  Sourceset_X_ss[[7]] <- Sources_numsources2_30level$auxXlist_list[[ss]][[1]]
  Sourceset_Y_ss[[7]] <- Sources_numsources2_30level$auxYlist_list[[ss]][[1]]

  Sourceset_X_ss[[8]] <- Sources_numsources2_30level$auxXlist_list[[ss]][[2]]
  Sourceset_Y_ss[[8]] <- Sources_numsources2_30level$auxYlist_list[[ss]][[2]]

  Sourceset_X[[ss]] <- Sourceset_X_ss
  Sourceset_Y[[ss]] <- Sourceset_Y_ss
}
```

### 처음 세개의 replicates에 대한 결과 얻기

```
In [11]: testrep1to3 <- rep_forsimul(Ylist=targestset$Ylist,
Xlist=targestset$Xlist,
auxYlist_list=Sourceset_Y,
auxXlist_list=Sourceset_X, repstart=1, repend=3, Btrue=targestset$B)
```

### 결과 요약

- Detection results

```
In [15]: Summary_ft_detection(testrep1to3, numinf=3)
```

PCI:FSD: '100%' Card:FSD: '3(0)' PCI:MSD: '0%' Card:MSD: '4.33333333333333(0.577)'

- Estimation results

```
In [14]: Summary_ft_est(testrep1to3)
```

NR: '0.198(0.03)' Pooled-NR: '0.193(0.008)' [K]-Trans: '0.139(0.009)' MSD: '0.119(0.009)' FSD: '0.082(0.004)'