Package 'BrainCon'

September 30, 2021

Type Package

Title Inference the Partial Correlations Based on Time Series Data

Version 0.1.0

Author Yunhaonan Yang [aut, cre],
Peng Wu [aut],
Xin Gai [aut],
Yumou Qiu [aut],
Xiaohua Zhou [aut]

Maintainer Yunhaonan Yang <haonan_yy@pku.edu.cn>

Description A statistical tool to inference the multi-level partial correlations based on multi-subject time series data. It combines both individual and population level inference by using the methods of Qiu and Zhou. (2021)<DOI:10.1080/01621459.2021.1917417> and Genovese and Wasserman. (2006)<DOI:10.1198/016214506000000339>. It can be used to identify nonzero individual- or population-level partial correlations, and find out unequal partial correlation coefficients between two populations.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports glmnet, MASS

Depends R (>= 2.10)

NeedsCompilation no

Repository CRAN

Date/Publication 2021-09-30 09:10:05 UTC

R topics documented:

`individual.est` *Estimate individual-level partial correlation coefficients*

Description

Estimate individual-level partial correlation coefficients in time series data with $1-\alpha$ confidence interval. It's not a joint confidence interval for multiple tests.

Usage

```
individual.est(X, alpha = 0.05, lambda = NULL, ci = TRUE)
```

Arguments

`X`	time series data of an individual which is a $n*p$ numeric matrix.
`alpha`	significance level, default value is `0.05`.
`lambda`	a penalty parameter used in lasso of order `sqrt(log(p)/n)`, if `NULL`, `2*sqrt(log(p)/n)` will be used.
`ci`	a logical indicating whether to compute $1-\alpha$ confidence interval, default value is `TRUE`.

Value

An `indEst` class object containing two or four components.

`coef` a $p*p$ partial correlation coefficients matrix.

`ci.lower` a $p*p$ numeric matrix containing the lower bound of $1-\alpha$ confidence interval, returned if `ci` is `TRUE`.

`ci.upper` a $p*p$ numeric matrix containing the upper bound of $1-\alpha$ confidence interval, returned if `ci` is `TRUE`.

`asym.ex` a matrix measuring the asymptotical expansion of estimates, which will be used for multiple tests.

References

Qiu Y. and Zhou X. (2021). Inference on multi-level partial correlations based on multi-subject time series data, *Journal of the American Statistical Association*, 00, 1-15

Examples

```
## Quick example for the individual-level estimates
data(indsim)
pc = individual.est(indsim)       # estimating partial correlation coefficients
```

`individual.test` *Identify nonzero individual-level partial correlations*

Description

Identify nonzero individual-level partial correlations in time series data by controlling the exceedance rate of the false discovery proportion (FDP) at $\alpha = 0.05$, considering time dependence. Input data X contains values of p interested variables over n periods.

Usage

```
individual.test(indEst, alpha = 0.05, c0 = 0.1, MBT = 3000)
```

Arguments

`indEst`	An `indEst` class object.
`alpha`	significance level, default value is `0.05`.
`c0`	threshold of the exceedance rate of the false discovery proportion (FDP), default value is `0.1`. The choice of `c0` depends on the empirical problem. A smaller value of `c0` will reduce false positives, but it may also cost more false negatives.
`MBT`	times of multiplier bootstrap, default value is `3000`.

Value

A $p * p$ matrix with values 0 or 1. If the j-th row and k-th column of the matrix is 1, then the partial correlation coefficient between the j-th variable and the k-th variable is identified to be nonzero.

References

Qiu Y. and Zhou X. (2021). Inference on multi-level partial correlations based on multi-subject time series data, *Journal of the American Statistical Association*, 00, 1-15

See Also

`population.test` for making inferences on one individual in the population.

Examples

```
## Quick example for the individual-level inference
data(indsim)
pc = individual.est(indsim)       # estimating partial correlation coefficients
Res = individual.test(pc)         # conducting hypothesis test
```

---

indsim *Simulation time series data for individual*

Description

A dataset containing values of 10 interested variables over 50 periods.

Usage

```
indsim
```

Format

An object of class `matrix` (inherits from `array`) with 50 rows and 10 columns.

Examples

```
## Generated by the following R codes
set.seed(1000)
n = 50
p = 10
precision = diag(rep(2, p))        # generate precision matrix
for (i in 1 : (p - 1)){
  if (i <= 2 * p / 3) temp = 1
  if (i > 2 * p / 3) temp = 0.4
  precision[i, i + 1] = temp
  precision[i + 1, i] = temp
}
Sigma = solve(precision)           # generate covariance matrix
rho = 0.5
X = matrix(0, n, p)                # observed time series data
Epsilon = MASS::mvrnorm(n, rep(0, p), Sigma)
X[1, ] = Epsilon[1, ]
for (i in 2 : n){
  X[i, ] = rho * X[i - 1, ] + sqrt(1 - rho^2) * Epsilon[i, ]
}
indsim = X
```

popsimA *Simulation time series data for population A*

Description

A dataset containing values of 10 interested variables of 20 subjects over 50 periods.

Usage

```
popsimA
```

Format

An object of class `array` of dimension 50 x 10 x 20.

See Also

popsimB.

Examples

```
## Generated by the following R codes
set.seed(1000)
n = 50
p = 10
m1 = 20; m2 = 15
precision1 = diag(rep(1, p))                  # generate precision matrix
for (i in 1 : (p - 1)){
  if (i <= 2 * p / 3) temp = -0.05
  if (i > 2 * p / 3) temp = -0.2
  precision1[i, i + 1] = temp
  precision1[i + 1, i] = temp
}
res = matrix(0, p, p)
for (i in 1 : p){
  for (j in 1 : p){
    if (abs(i - j) == 1) res[i, j] = 0.15
  }
}
precision2 = precision1 + res
Index0 = matrix(0, p, p)                     # generate covariance matrix
for (i in 1 : p){
  for (j in 1 : p){
    if (i != j & abs(i - j) <= 5) Index0[i, j] = 1
  }
}
SigmaAll1 = array(dim = c(p, p, m1))
SigmaAll2 = array(dim = c(p, p, m2))
for (sub in 1 : m1){
  RE = matrix(rnorm(p^2, 0, sqrt(2) * 0.05), p, p) * Index0
```

```
  RE1 = (RE + t(RE)) / 2
  precisionMatrixInd1 = precision1 + RE1
  SigmaInd1 = solve(precisionMatrixInd1)
  SigmaAll1[, , sub] = SigmaInd1
}
for (sub in 1 : m2){
  RE = matrix(rnorm(p^2, 0, sqrt(2) * 0.05), p, p) * Index0
  RE1 = (RE + t(RE)) / 2
  precisionMatrixInd2 = precision2 + RE1
  SigmaInd2 = solve(precisionMatrixInd2)
  SigmaAll2[, , sub] = SigmaInd2
}
rho = 0.5
Z1 = array(dim = c(n, p, m1))                # observed time series data
Z2 = array(dim = c(n, p, m2))
for (sub in 1 : m1){
  SigmaInd1 = SigmaAll1[, , sub]
  Xtemp = matrix(0, n, p)
  Epsilon = MASS::mvrnorm(n, rep(0, p), SigmaInd1)
  Xtemp[1, ] = Epsilon[1, ]
  for (i in 2 : n){
    Xtemp[i, ] = rho * Xtemp[i - 1, ] + sqrt(1 - rho^2) * Epsilon[i, ]
  }
  Z1[, , sub] = Xtemp
}
for (sub in 1 : m2){
  SigmaInd2 = SigmaAll2[, , sub]
  Xtemp = matrix(0, n, p)
  Epsilon = MASS::mvrnorm(n, rep(0, p), SigmaInd2)
  Xtemp[1, ] = Epsilon[1, ]
  for (i in 2 : n){
    Xtemp[i, ] = rho * Xtemp[i - 1, ] + sqrt(1 - rho^2) * Epsilon[i, ]
  }
  Z2[, , sub] = Xtemp
}
popsimA = Z1
popsimB = Z2
```

`popsimB` *Simulation time series data for population B*

Description

A dataset containing values of 10 interested variables of 15 subjects over 50 periods.

Usage

```
popsimB
```

Format

An object of class `array` of dimension 50 x 10 x 15.

See Also

`popsimA`.

`population.est` *Estimate population-level partial correlation coefficients*

Description

Estimate population-level partial correlation coefficients in time series data. And also return each individual-level coefficients.

Usage

```
population.est(Z, alpha = 0.05, lambda = NULL, ind.ci = FALSE)
```

Arguments

`Z`	is a $n * p * m$ dimensional array, where m is number of subjects.
`alpha`	significance level, default value is `0.05`.
`lambda`	a penalty parameter used in lasso of order `sqrt(log(p)/n)`, if `NULL`, `2*sqrt(log(p)/n)` will be used.
`ind.ci`	a logical indicating whether to compute $1-\alpha$ confidence interval of each subject, default value is `FALSE`.

Value

A `popEst` class object containing two components.

`coef` a $p * p$ partial correlation coefficients matrix.

`ind.est` a m-length list, containing estimates for each individuals.

References

Qiu Y. and Zhou X. (2021). Inference on multi-level partial correlations based on multi-subject time series data, *Journal of the American Statistical Association*, 00, 1-15

Examples

```
## Quick example for the individual-level estimates
data(popsimA)
pc = population.est(popsimA)        # estimating partial correlation coefficients
```

population.test *The one-sample population inference*

Description

Identify the nonzero partial correlations in one-sample population, based on false discovery proportion controlling. at $\alpha = 0.05$, considering time dependence. Input data Z , contains values of p interested variables.

Usage

```
population.test(popEst, alpha = 0.05, c0 = 0.1, MBT = 3000)
```

Arguments

popEst	A popEst class object.
alpha	significance level, default value is 0.05.
c0	threshold of the exceedance rate of the false discovery proportion (FDP), default value is 0.1. The choice of c0 depends on the empirical problem. A smaller value of c0 will reduce false positives, but it may also cost more false negatives.
MBT	times of multiplier bootstrap, default value is 3000.

Value

A $p * p$ matrix with values 0 or 1.

References

Qiu Y. and Zhou X. (2021). Inference on multi-level partial correlations based on multi-subject time series data, *Journal of the American Statistical Association*, 00, 1-15

See Also

individual.test.

Examples

```
## Quick example for the one-sample population inference
data(popsimA)
pc = population.est(popsimA)     # estimating partial correlation coefficients
Res = population.test(pc)        # conducting hypothesis test

## Inference on the first subject in population
Res1 = individual.test(pc$ind.est[[1]])
```

population.test.MinPv *The one-sample population inference using Genovese and Wasserman's method*

Description

Identify the nonzero partial correlations in one-sample population, based on controlling the exceedance rate of the false discovery proportion (FDP) at $\alpha = 0.05$. The method is based on the minimum of the p-values. Input data Z , contains values of p interested variables.

Usage

```
population.test.MinPv(popEst, alpha = 0.05, c0 = 0.1)
```

Arguments

`popEst`	A `popEst` class object.
`alpha`	significance level, default value is `0.05`.
`c0`	threshold of the exceedance rate of the false discovery proportion (FDP), default value is `0.1`.

Value

A $p * p$ matrix with values 0 or 1.

References

Genovese C., and Wasserman L. (2006). Exceedance Control of the False Discovery Proportion, *Journal of the American Statistical Association*, 101, 1408-1417

Qiu Y. and Zhou X. (2021). Inference on multi-level partial correlations based on multi-subject time series data, *Journal of the American Statistical Association*, 00, 1-15

Examples

```
## Quick example for the one-sample population inference
data(popsimA)
pc = population.est(popsimA)          # estimating partial correlation coefficients
Res  = population.test.MinPv(pc)       # conducting hypothesis test
```

`population2sample.test`
Identify differences of partial correlations between two populations

---

Description

Identify differences of partial correlations between two populations in two groups of time series data by controlling the exceedance rate of the false discovery proportion (FDP) at $\alpha = 0.05$, considering time dependence. Input two groups of data Z_1 and Z_2, each contains values of p interested variables of individuals (the number of individuals in two groups can be different) over n periods.

Usage

```
population2sample.test(popEst1, popEst2, alpha = 0.05, c0 = 0.1, MBT = 3000)
```

Arguments

`popEst1`	A `popEst` class object.
`popEst2`	A `popEst` class object.
`alpha`	significance level, default value is `0.05`.
`c0`	threshold of the exceedance rate of the false discovery proportion (FDP), default value is `0.1`. The choice of `c0` depends on the empirical problem. A smaller value of `c0` will reduce false positives, but it may also cost more false negatives.
`MBT`	times of multiplier bootstrap, default value is `3000`.

Value

A $p * p$ matrix with values 0 or 1. If the j-th row and k-th column of the matrix is 1, then the partial correlation coefficients between the j-th variable and the k-th variable in two populations are identified to be unequal.

References

Qiu Y. and Zhou X. (2021). Inference on multi-level partial correlations based on multi-subject time series data, *Journal of the American Statistical Association*, 00, 1-15

Examples

```
## Quick example for the two-sample case inference
data(popsimA)
data(popsimB)
pc1 = population.est(popsimA)
pc2 = population.est(popsimB)
Res = population2sample.test(pc1, pc2)          # conducting hypothesis test
```

`population2sample.test.MinPv`

Identify differences of partial correlations between two populations using Genovese and Wasserman's method

Description

Identify differences of partial correlations between two populations in two groups of time series data by controlling the exceedance rate of the false discovery proportion (FDP) at $\alpha = 0.05$. The method is based on the minimum of the p-values. Input two groups of data Z_1 and Z_2, each contains values of p interested variables of individuals (the number of individuals in two groups can be different) over n periods.

Usage

```
population2sample.test.MinPv(popEst1, popEst2, alpha = 0.05, c0 = 0.1)
```

Arguments

popEst1	A `popEst` class object.
popEst2	A `popEst` class object.
alpha	significance level, default value is `0.05`.
c0	threshold of the exceedance rate of the false discovery proportion (FDP), default value is `0.1`. The choice of `c0` depends on the empirical problem. A smaller value of `c0` will reduce false positives, but it may also cost more false negatives.

Value

A $p * p$ matrix with values 0 or 1. If the j-th row and k-th column of the matrix is 1, then the partial correlation coefficients between the j-th variable and the k-th variable in two populations are identified to be unequal.

References

Genovese C., and Wasserman L. (2006). Exceedance Control of the False Discovery Proportion, *Journal of the American Statistical Association*, 101, 1408-1417

Qiu Y. and Zhou X. (2021). Inference on multi-level partial correlations based on multi-subject time series data, *Journal of the American Statistical Association*, 00, 1-15

Examples

```
## Quick example for the two-sample case inference
data(popsimA)
data(popsimB)
pc1 = population.est(popsimA)
```

```
pc2 = population.est(popsimB)
Res = population2sample.test.MinPv(pc1, pc2)         # conducting hypothesis test
```

Index