

Package ‘NCP’

January 21, 2015

Type Package

Title Network Change Point Detection Using Spectral Clustering

Version 0.1

Date 2015-01-21

Author Ivor Cribben and Yi Yu

Maintainer Yi Yu <y.yu@statslab.cam.ac.uk>

Depends R (>= 3.1.1), parallel, MASS, igraph

Description This package detects the change points in terms of network structures. Spectral clustering is used to extract network structure, principal angles are the criterion for change points, and stationary bootstrap is used to test the significance of the change points chosen by exhaustive search.

License GPL-3

LazyLoad yes

R topics documented:

NCP-package	1
group.corr	3
ncpplots	3
spectral.clustering	4
spectralfunction	5

Index	7
--------------	----------

NCP-package	<i>Network Change Point Detection Using Spectral Clustering</i>
-------------	---

Description

This package detects the change points in terms of network structures. Spectral clustering is used to extract network structure, principal angles are the criterion for change points, and stationary bootstrap is used to test the significance of the change points chosen by exhaustive search.

Details

Package: NCP
 Type: Package
 Version: 0.1
 Date: 2015-01-21
 License: GPL-3

Author(s)

Ivor Cribben and Yi Yu

Maintainer: Yi Yu <y.yu@statslab.cam.ac.uk>

References

Cribben, I. and Yu, Y. (2014+). Estimating whole brain dynamics using spectral clustering.

Examples

```
### generate data
t      = 100
s.p    = 200
gen.k  = 2
n.tot  = gen.k * s.p
rho    = 0.5
Sigma1 = group.corr(small.n = s.p, rho = rho, p = n.tot)
reorder = sample(c(1:n.tot), n.tot)
Sigma2 = Sigma1[reorder, reorder]
set.seed(1)
Y1     = mvrnorm(n = t, mu = rep(0, n.tot), Sigma = Sigma1)
Y2     = mvrnorm(n = t, mu = rep(0, n.tot), Sigma = Sigma2)
y      = rbind(Y1, Y2)

### set arguments
changeplot = 2
network    = 1
T          = 200
quantile_1 = 0.025
quantile_2 = 0.975
K          = 5
thres     = 0.3
core.limit = 2
# by R policy, the examples cannot use more than 2 cores
# in practice, just leave this argument missing
# all the cores will be used then

### run
spectral(y = y, K = K, core.limit = core.limit)
### four files are automatically saved in the working directory from the previous function
```

```
### three of them are required to proceed the following function
# ncplots(y = y, changepointsplot = changepointsplot, network = network, K = K, thres = thres)
```

group.corr	<i>Generate a correlation matrix with a group structure.</i>
------------	--

Description

Used in generating the example data. For a $p \times p$ correlation matrix with group structure, the off-diagonal entries within each group is ρ and 0 otherwise.

Usage

```
group.corr(small.n, rho, p)
```

Arguments

small.n	group size – a positive integer.
rho	the within group correlation – a real value in [0, 1].
p	the matrix dimension – a positive integer greater than or equal to small.n.

Value

a $p \times p$ correlation matrix.

Author(s)

Ivor Cribben and Yi Yu

ncplots	<i>Network change plots.</i>
---------	------------------------------

Description

[ncplots](#) is the main function to plot the change points against time and the undirected graphs, results thereof are provided by [spectral.clustering.plot](#) and [network.cluster.plot](#).

Usage

```
ncplots(y, changepointsplot, network = 0, quantile_1 = 0.025, quantile_2 = 0.975,
actualcps = 'ActualCPs', bootcps = 'BootCPs', signcps = 'SignCPs', K, thres = 0.3)
spectral.clustering.plot(ny, nK)
network.cluster.plot(datay, K, thres = 0.3)
```

Arguments

y	input matrix – n.timepoints x n.voxels, used in ncpplots .
datay	the input matrix – an n.timepoints x n.voxels matrix, used in network.cluster.plot .
ny	the input matrix – an n.timepoints x n.voxels matrix, used in spectral.clustering.plot .
nK	number of communities – a positive integer, used in spectral.clustering.plot .
changeplot	the number of significant change points – a positive integer.
network	number of networks to plot – a positive integer. If no network to plot, network = 0. The default is 0.
quantile_1	the lower quantile for change points. The default is 0.025.
quantile_2	the upper quantile for change points. The default is 0.975.
actualcps	the output file containing acutal change points from spectralfunction .
bootcps	the output file containing stationary bootstrap change points from spectralfunction .
signcps	the output file containing significant bootstrap change points from spectralfunction .
K	number of communities – a positive integer, used in ncpplots and network.cluster.plot .
thres	the threshold on the correlation used to construct an adjacency matrix – a real value in [0,1]. The default is 0.3.

Value

A plot is given with x-axis being the time points and y-axis being the cosine values of the angles at the change points.

Author(s)

Ivor Cribben and Yi Yu

References

Cribben, I. and Yu, Y. (2014+). Estimating whole brain dynamics using spectral clustering.

See Also

[spectralfunction](#).

spectral.clustering	<i>Using spectral clustering to detect the community structure in an undirected simple network.</i>
---------------------	---

Description

Different from the conventional spectral clustering, the output of this function is a centroid matrix. After a K-dimensional embedding, replacing each node by its corresponding centroid.

Usage

```
spectral.clustering(Y, K, t.min = 1, t.max)
```

Arguments

Y	input matrix – an n.timepoints x n.voxels matrix.
K	the pre-specified community number – a positive integer.
t.min	the first time point of interest. The default is 1.
t.max	the last time point of interest. If missing, the last point available is set.

Value

a n.voxels x K correlation matrix.

Author(s)

Ivor Cribben and Yi Yu

References

- Cribben, I. and Yu, Y. (2014+). Estimating whole brain dynamics using spectral clustering.
 Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17, 395–416.

Examples

```
### generate data
t      = 100
s.p    = 200
gen.k  = 2
n.tot  = gen.k * s.p
rho    = 0.5
Sigma1 = group.corr(small.n = s.p, rho = rho, p = n.tot)
y      = mvrnorm(n = t, mu = rep(0, n.tot), Sigma = Sigma1)

### set arguments
K      = 5

### run
fit = spectral.clustering(Y = y, K = K)
```

spectralfunction *Network change points detection using spectral clustering.*

Description

The main function in this package.

Usage

```
spectralfunction(y, K, lower = 0, upper, maxn = 50, quan = 0.9, n.boot = 100, block = 0.05,
  quantile_1 = 0.025, quantile_2 = 0.975, actualcps = 'ActualCPs',
  bootcps = 'BootCPs', signcps = 'SignCPs', core.limit)
```

Arguments

<code>y</code>	dataset to be analysed – a <code>n.timepoints x n.voxels</code> matrix.
<code>K</code>	the pre-specified number of communities – a positive integer.
<code>lower</code>	first time point for the algorithm. The default is 0.
<code>upper</code>	last time point for the algorithm. If missing, then the default is the last time point in <code>y</code> .
<code>maxn</code>	minimum number of time points to construct a network. The default choice is 50.
<code>quan</code>	quantile threshold – used in detect outliers in the criterion values. The default choice is 0.9.
<code>n.boot</code>	number of iterations used in the stationary bootstrap. The default choice is 100.
<code>block</code>	size of average block length for stationary bootstrap. The default is 0.05.
<code>quantile_1</code>	the lower quantile for testing, used in the stationary bootstrap. The default is 0.025.
<code>quantile_2</code>	the upper quantile for testing, used in the stationary bootstrap. The default is 0.975.
<code>actualcps</code>	the name of the file containing the actual change points. The default is 'Actual-CPs'.
<code>bootcps</code>	the name of the file containing the actual change points. The default is 'BootCPs'.
<code>signcps</code>	the name of the file containing the actual change points. The default is 'SignCPs'.
<code>core.limit</code>	the upper limit of the cores could be used. If missing, all the cores detected will be used.

Details

The smaller the criterion value is, the more significant the corresponding change point is.

Value

<code>ActualCPs</code>	a plain text file containing the change points and their corresponding criterion values by exhaustive search.
<code>BootCPs</code>	a plain text file containing the stationary bootstrap results on the change points detected by exhaustive search.
<code>SignCPs</code>	a plain text file containing the significant change points and their corresponding criterion values after stationary bootstrap inference.

Author(s)

Ivor Cribben and Yi Yu

References

Cribben, I. and Yu, Y. (2014+). Estimating whole brain dynamics using spectral clustering.

Index

***Topic fMRI data, brain connectivity,
community detection,
spectral clustering**

NCP-package, 1

group.corr, 3

NCP (NCP-package), 1

NCP-package, 1

ncplots, 3, 3, 4

network.cluster.plot, 3, 4

network.cluster.plot (ncplots), 3

spectral.clustering, 4

spectral.clustering.plot, 3, 4

spectral.clustering.plot (ncplots), 3

spectralfunction, 4, 5