Package ‘ccid’

July 22, 2025

Type Package

Title Cross-Covariance Isolate Detect: a New Change-Point Method for Estimating Dynamic Functional Connectivity

Version 1.2.0

Depends R (>= 3.6.0)

Imports IDetect, hdbinseg, GeneNet, gdata

Description Provides efficient implementation of the Cross-Covariance Isolate Detect (CCID) methodology for the estimation of the number and location of multiple change-points in the second-order (cross-covariance or network) structure of multivariate, possibly high-dimensional time series. The method is motivated by the detection of change points in functional connectivity networks for functional magnetic resonance imaging (fMRI), electroencephalography (EEG), magentoencephalography (MEG) and electrocorticography (ECoG) data. The main routines in the package have been extensively tested on fMRI data. For details on the CCID methodology, please see Anastasiou et al (2022), Cross-covariance isolate detect: A new change-point method for estimating dynamic functional connectivity. Medical Image Analysis, Volume 75.

License GPL-3

Encoding UTF-8

RoxygenNote 7.1.1

URL `https://github.com/Anastasiou-Andreas/ccid`

BugReports `https://github.com/Anastasiou-Andreas/ccid/issues`

Suggests testthat (>= 2.0.0)

Config/testthat/edition 2

NeedsCompilation no

Author Andreas Anastasiou [aut, cre],
Ivor Cribben [aut],
Piotr Fryzlewicz [aut]

Maintainer Andreas Anastasiou `<anastasiou.andreas@ucy.ac.cy>`

Repository CRAN

Date/Publication 2022-02-01 10:50:02 UTC

Contents

`ccid` *ccid: a change-point detection method for estimating dynamic functional connectivity*

Description

The `ccid` package implements the Cross-Covariance Isolate Detect (CCID) methodology for the estimation of the number and location of multiple change-points in the second-order (cross-covariance or network) structure of multivariate, possibly high-dimensional time series. The method is motivated by the detection of change points in functional connectivity networks for functional magnetic resonance imaging (fMRI), electroencephalography (EEG), magentoencephalography (MEG) and electrocorticography (ECoG) data. The stopping rules used for the change-point detection rely either on thresholding or on the optimization of a model selection criterion. The main routines of the package are `detect.th` and `detect.ic`. The functions have been extensively tested on fMRI data, therefore, their parameters have been tuned to work well on this data and the functions might not work well in other structures, such as time series that are negatively serially correlated.

Author(s)

Andreas Anastasiou, `<anastasiou.andreas@ucy.ac.cy>`, Piotr Fryzlewicz, `<p.fryzlewicz@lse.ac.uk>`, Ivor Cribben, `<cribben@ualberta.ca>`

References

“Cross-covariance isolate detect: a new change-point method for estimating dynamic functional connectivity”, Anastasiou et al (2020), preprint.

See Also

`detect.th` and `detect.ic`.

Examples

```
# See Examples for the function ``detect.th''.
```

`detect.ic` *Multiple change-point detection in the cross-covariance structure of multivariate high-dimensional time series using a model selection criterion optimisation*

Description

This function detects multiple change-points in the cross-covariance structure of a multivariate time series using a model selection criterion optimisation.

Usage

```
detect.ic(
  X,
  approach = c("euclidean", "infinity"),
  th_max = 2.1,
  th_sum = 0.5,
  pointsgen = 10,
  scales = -1,
  alpha_gen = 0.1,
  preaverage_gen = FALSE,
  scal_gen = 3,
  min_dist = 1
)
```

Arguments

`X`	A numerical matrix representing the multivariate time series, with the columns representing its components.
`approach`	A character string, which defines the metric to be used in order to detect the change-points. If approach = “euclidean”, which is also the default value, then the L_2 metric will be followed for the detection. If approach = “infinity”, then the L_∞ metric will be used for the detection.
`th_max`	A positive real number with default value equal to 2.1. It is used to define the threshold for the change-point overestimation step if the L_∞ metric is chosen in `approach` .
`th_sum`	A positive real number with default value equal to 0.5. It is used to define the threshold for the change-point overestimation step if the L_2 metric is chosen in `approach`.
`pointsgen`	A positive integer with default value equal to 10. It defines the distance between two consecutive end- or start-points of the right- or left-expanding intervals, respectively; see Details for more information.
`scales`	Negative integers for wavelet scales, with a small negative integer representing a fine scale. The default value is equal to -1.
`alpha_gen`	A positive real number with default value equal to 0.1. It is used to define how strict the user wants to be with the penalty used.

`preaverage_gen`	A logical variable with default value equal to `FALSE`. If `FALSE`, then pre-averaging the data is not required. If `TRUE`, then we need to pre-average the data before proceeding with the detection of the change-points.
`scal_gen`	A positive integer number with default value equal to 3. It is used to define the way we pre-average the given data sequence only if `preaverage_gen = TRUE`. See the Details in `preaverage` for more information on how we pre-average.
`min_dist`	A positive integer number with default value equal to 1. It is used in order to provide the minimum distance acceptable between detected change-points if such restrictions apply.

Details

The time series X_t is of dimensionality p and we are looking for changes in the cross-covariance structure between the different time series components $X_t^{(1)}, X_t^{(2)}, ..., X_t^{(p)}$. We first use a wavelet-based approach for the various given scales in `scales` in order to transform the given time series X_t to a multiplicative model $Y_t^{(k)} = \sigma_t^{(k)}(Z_t^{(k)})^2; t = 1, 2, \ldots, T; k = 1, 2, \ldots, d$, where $Z_t^{(k)}$ is a sequence of standard normal random variables, $E(Y_t^{(k)}) = \sigma_t^{(k)}$, and d is the new dimensionality, which depends on the value given in `scales`. The function has been extensively tested on fMRI data, hence, its parameters have been tuned for this data type. The function might not work well in other structures, such as time series that are negatively serially correlated.

Value

A list with the following components:

`changepoints`	The locations of the detected change-points.
`no.of.cpts`	The number of the detected change-points.
`sol_path`	A vector containing the solution path.
`ic_curve`	A vector with values of the information criterion for different number of change-points.

If the minimum distance between the detected change-points is less than the value given in `min_dist`, then only the number and the locations of the "pruned" change-points are returned.

Author(s)

Andreas Anastasiou, <`anastasiou.andreas@ucy.ac.cy`>

References

"Cross-covariance isolate detect: a new change-point method for estimating dynamic functional connectivity", Anastasiou et al (2020), preprint <doi:10.1101/2020.12.20.423696>.

See Also

`detect.th`.

Examples

```
set.seed(11)
A <- matrix(rnorm(10*200), nrow = 200) ## No change-point
M1 <- detect.ic(A, approach = 'euclidean', scales = -1)
M2 <- detect.ic(A, approach = 'infinity', scales = -1)
M1$changepoints
M2$changepoints

set.seed(1)
num.nodes <- 30 # number of nodes
etaA.1    <- 0.95
etaA.2    <- 0.05
pcor1     <- GeneNet::ggm.simulate.pcor(num.nodes, etaA = etaA.1)
pcor2     <- GeneNet::ggm.simulate.pcor(num.nodes, etaA = etaA.2)

n <- 50
data1 <- GeneNet::ggm.simulate.data(n, pcor1)
data2 <- GeneNet::ggm.simulate.data(n, pcor2)
X1 <- rbind(data1, data2, data1, data2) ## change-points at 50, 100, 150
N1 <- detect.ic(X1, approach = 'euclidean', scales = -1)
N2 <- detect.ic(X1, approach = 'infinity', scales = -1)
N1$changepoints
N2$changepoints
N1$no.of.cpts
N2$no.of.cpts
N1$sol_path
N2$sol_path
```

`detect.th` *Multiple change-point detection in the cross-covariance structure of multivariate high-dimensional time series using a thresholding based procedure and, wherever possible, extraction of the component time series where the changes occurred*

Description

This function detects multiple change-points in the cross-covariance structure of a multivariate time series using a thresholding based procedure. It also, wherever possible, returns the relevant, transformed time series where each change-point was detected. See Details for a brief explanation.

Usage

```
detect.th(
  X,
  approach = c("euclidean", "infinity"),
  th_max = 2.25,
  th_sum = 0.65,
  pointsgen = 10,
  scales = -1,
```

```
  preaverage_gen = FALSE,
  scal_gen = 3,
  min_dist = 1
)
```

Arguments

`X`	A numerical matrix representing the multivariate time series, with the columns representing its components.
`approach`	A character string, which defines the metric to be used in order to detect the change-points. If approach = "euclidean", which is also the default value, then the L_2 metric will be followed for the detection. If approach = "infinity", then the L_∞ metric will be used for the detection.
`th_max`	A positive real number with default value equal to 2.25. It is used to define the threshold if the L_∞ metric is chosen in `approach` .
`th_sum`	A positive real number with default value equal to 0.65. It is used to define the threshold if the L_2 metric is chosen in `approach`.
`pointsgen`	A positive integer with default value equal to 10. It defines the distance between two consecutive end- or start-points of the right- or left-expanding intervals, respectively; see Details for more information.
`scales`	Negative integers for wavelet scales, with a small negative integer representing a fine scale. The default value is equal to -1.
`preaverage_gen`	A logical variable with default value equal to `FALSE`. If `FALSE`, then pre-averaging the data is not required. If `TRUE`, then we need to pre-average the data before proceeding with the detection of the change-points.
`scal_gen`	A positive integer number with default value equal to 3. It is used to define the way we pre-average the given data sequence only if `preaverage_gen = TRUE`. See the Details in `preaverage` for more information on how we pre-average.
`min_dist`	A positive integer number with default value equal to 1. It is used in order to provide the minimum distance acceptable between detected change-points if such restrictions apply.

Details

The time series X_t is of dimensionality p and we are looking for changes in the cross-covariance structure between the different time series components $X_t^{(1)}, X_t^{(2)}, ..., X_t^{(p)}$. We first use a wavelet-based approach for the various given scales in `scales` in order to transform the given time series X_t to a multiplicative model $Y_t^{(k)} = \sigma_t^{(k)}(Z_t^{(k)})^2; t = 1, 2, \ldots, T; k = 1, 2, \ldots, d$, where $Z_t^{(k)}$ is a sequence of standard normal random variables, $E(Y_t^{(k)}) = \sigma_t^{(k)}$, and d is the new dimensionality, which depends on the value given in `scales`. The function has been extensively tested on fMRI data, hence, its parameters have been tuned for this data type. The function might not work well in other structures, such as time series that are negatively serially correlated.

Value

A list with the following components:

`changepoints`	The locations of the detected change-points.
`no.of.cpts`	The number of the detected change-points.
`time_series`	A list with two components that indicates which combinations of time series are responsible for each change-point detected. See the outcome values `time_series_indicator` and `most_important` of the function `match.cpt.ts` for more information.

If the minimum distance between the detected change-points is less than the value given in `min_dist`, then only the number and the locations of the "pruned" change-points are returned.

Author(s)

Andreas Anastasiou, `<anastasiou.andreas@ucy.ac.cy>`

References

"Cross-covariance isolate detect: a new change-point method for estimating dynamic functional connectivity", Anastasiou et al (2020), preprint <doi:10.1101/2020.12.20.423696>.

See Also

`detect.ic`.

Examples

```
set.seed(111)
A <- matrix(rnorm(20*400), nrow = 400) ## No change-point
M1 <- detect.th(A, approach = 'euclidean', scales = -1)
M2 <- detect.th(A, approach = 'infinity', scales = -1)
M1
M2

set.seed(111)
num.nodes <- 40 # number of nodes
etaA.1    <- 0.95
etaA.2    <- 0.05
pcor1     <- GeneNet::ggm.simulate.pcor(num.nodes, etaA = etaA.1)
pcor2     <- GeneNet::ggm.simulate.pcor(num.nodes, etaA = etaA.2)

n <- 100
data1 <- GeneNet::ggm.simulate.data(n, pcor1)
data2 <- GeneNet::ggm.simulate.data(n, pcor2)

X1 <- rbind(data1, data2) ## change-point at 100
N1 <- detect.th(X1, approach = 'euclidean', scales = -1)
N2 <- detect.th(X1, approach = 'infinity', scales = -1)
N1$changepoints
N1$time_series
N2$changepoints
N2$time_series
```

`match.cpt.ts` *Associating the change-points with the component time series*

Description

This function performs a contrast function based approach in order to match each change-point and time series. In simple terms, for a given change-point set this function associates each change-point with the respective data sequence (or sequences) from which it was detected.

Usage

```
match.cpt.ts(
  X,
  cpt,
  thr_const = 1,
  thr_fin = thr_const * sqrt(2 * log(nrow(X))),
  scales = -1,
  count = 5
)
```

Arguments

`X`	A numerical matrix representing the multivariate periodograms. Each column contains a different periodogram which is the result of applying the wavelet transformation to the initial multivariate time series.
`cpt`	A positive integer vector with the locations of the change-points. If missing, then our approach with the L_2 aggregation is called internally to extract the change-points in `X`.
`thr_const`	A positive real number with default value equal to 1. It is used to define the threshold; see `thr_fin`.
`thr_fin`	With `T` the length of the data sequence, this is a positive real number with default value equal to `thr_const * log(T)`. It is the threshold, which is used in the detection process.
`scales`	Negative integers for the wavelet scales used to create the periodograms, with a small negative integer representing a fine scale. The default value is equal to -1.
`count`	Positive integer with default value equal to 5. It can be used so that the function will return only the `count` most important matches of each change-points with the time series.

Value

A list with the following components:

`time_series_indicator`	A list of matrices. There are as many matrices as

	the number of change-points. Each change-point has its own matrix, with each row of the matrix representing the associated combination of time series that are associated with the respective change-point.
`most_important`	A list of matrices. There are as many matrices as the number of change-points. Each change-point has its own matrix, with each row of the matrix representing the associated combination of time series that are associated with the respective change-point. It shows the `count` most important time series combinations for each change-point.

Author(s)

Andreas Anastasiou, `<anastasiou.andreas@ucy.ac.cy>`

References

"Cross-covariance isolate detect: a new change-point method for estimating dynamic functional connectivity", Anastasiou et al (2020), preprint <doi:10.1101/2020.12.20.423696>.

Examples

```
set.seed(1)
num.nodes <- 40 # number of nodes
etaA.1    <- 0.95
etaA.2    <- 0.05
pcor1     <- GeneNet::ggm.simulate.pcor(num.nodes, etaA = etaA.1)
pcor2     <- GeneNet::ggm.simulate.pcor(num.nodes, etaA = etaA.2)

n <- 100
data1 <- GeneNet::ggm.simulate.data(n, pcor1)
data2 <- GeneNet::ggm.simulate.data(n, pcor2)
X <- rbind(data1, data2, data1, data2) ## change-points at 100, 200, 300
sgn <- sign(stats::cor(X))
M1 <- match.cpt.ts(t(hdbinseg::gen.input(x = t(X),scales = -1, sq = TRUE,
diag = FALSE, sgn = sgn)))
M1
```

`preaverage` *Preaveraging the multivariate time series*

Description

This function pre-processes the given data in order to remove serial correlation that might exist in the given data.

Usage

```
preaverage(X, scal = 3)
```

Arguments

`X`	A numerical matrix representing the multivariate time series, with the columns representing its components.
`scal`	A positive integer number with default value equal to 3. It is used to define the way we pre-average the data sequences.

Details

For a given natural number `scal` and data matrix `X` of dimensionality $T \times d$, let us denote by $Q = \lceil T/scal \rceil$. Then, `preaverage` calculates, for all $j = 1, 2, ..., d$,

$$\tilde{X}_{q,j} = 1/scal \sum_{t=(q-1)*sc+1}^{q*sc} X_{t,j},$$

for $q = 1, 2, ..., Q - 1$, while

$$\tilde{x}_{Q,j} = (T - (Q-1)*sc)^{-1} \sum_{t=(Q-1)*sc+1}^{T} X_{t,j}.$$

Value

The “preaveraged” matrix $\tilde{X}$ of dimensionality $Q \times d$, as explained in Details.

Author(s)

Andreas Anastasiou, `<anastasiou.andreas@ucy.ac.cy>`

References

“Cross-covariance isolate detect: a new change-point method for estimating dynamic functional connectivity”, Anastasiou et al (2020), preprint <doi:10.1101/2020.12.20.423696>.

Examples

```
A <- matrix(1:32, 8, 4)
A
A1 <- preaverage(A, scal = 3)
A1
```

Index