Package ‘trendsegmentR’

October 14, 2022

Title Linear Trend Segmentation

Version 1.1.0

Maintainer Hyeyoung Maeng <h.maeng4@lancaster.ac.uk>

Description Performs the detection of linear trend changes for univariate time series by implementing the bottom-up unbalanced wavelet transformation proposed by H. Maeng and P. Fryzlewicz (2021). The estimated number and locations of the change-points are returned with the piecewise-linear estimator for signal.

License GPL

Encoding UTF-8

RoxygenNote 7.1.2

NeedsCompilation no

Repository CRAN

Author Hyeyoung Maeng [aut, cre],
Piotr Fryzlewicz [aut]

Date/Publication 2021-11-19 07:30:02 UTC

R topics documented:

invTGUW *Inverse Tail-Greedy Unbalanced Wavelet (TGUW) transformation*

Description

This function is used inside trendsegment and performs the inverse TGUW transformation by undoing the orthonormal transformation of TGUW in reverse order. Details of the inverse TGUW transformation can be found in H. Maeng and P. Fryzlewicz (2021), Detecting linear trend changes in data sequences, preprint.

Usage

```
invTGUW(ts.obj)
```

Arguments

ts.obj	A list returned by thresholding.

Value

ts.obj	The modified ts.obj is the result of the inverse TGUW transformation and ts.coeffs now presents the estimated piecewise-linear signal of the data.

Author(s)

Hyeyoung Maeng <h.maeng4@lancaster.ac.uk>, Piotr Fryzlewicz <p.fryzlewicz@lse.ac.uk>

See Also

trendsegment, TGUW, thresholding

Examples

```
x <- c(1:10, rep(5,9))
n <- length(x)
x <- x + rnorm(n)
tguwfit <- TGUW(x)
th.const <- 1.3
lambda <- (stats::mad(diff(diff(x)))/sqrt(6)) * sqrt(2 * log(n)) * th.const
thrfit <- thresholding(ts.obj = tguwfit, lambda = lambda, minsegL = 5, bal = 0, connected = TRUE)
invfit <- invTGUW(ts.obj = thrfit)
invfit
```

TGUW *Tail-Greedy Unbalanced Wavelet (TGUW) transformation of a vector*

Description

Performs the bottom-up unbalanced wavelet decomposition. This function is used inside `trendsegment`. Details of the TGUW transformation can be found in H. Maeng and P. Fryzlewicz (2021), Detecting linear trend changes in data sequences, preprint.

Usage

```
TGUW(x, p = 0.01)
```

Arguments

`x`	An input vector to be decomposed.
`p`	Proportion of all possible remaining merges which specifies the number of merges allowed in a single pass over the data. The default is 0.01.

Value

A list with the followings:

`x`	The original input vector `x`.
`n`	The length of `x`.
`twotogether`	A vector indicating locations of the detail coefficients returned by Type 3 merges (merging two sets of paired smooth coefficients). This is used in `thresholding` to apply the "two together" rule which makes both detail coefficients (paired by a Type 3 merge) survived if at least one of their size is over threshold.
`merging.hist`	An array of dimension 4 by 3 by `n`-2 which has the full record of the `n`-2 merges in the TGUW transformation. Each matrix contains the information of each merge. The first row shows the indices of merged smooth coefficients in increasing order and the second row gives the value of detail filter coefficients which is the weight vector for computing the corresponding detail coefficient. The third row shows the (detail coefficient, first smooth coefficient, second smooth coefficient) obtained by an orthonormal transform. The fourth row gives the balancedness of merging. If it is Type 1 merging (three initial smooth coefficients) then the fourth row is always (1/3, 1/3, 1/3). In Type 2 and Type 3 mergings, the values depend on the ratio of the length of the left and right wings to the entire merged region and only first two components of the fourth row are filled with the corresponding ratios (sum to 1) but the third one is left as NA.
`ts.coeffs`	The transformed `x` by the TGUW transformation.

Author(s)

Hyeyoung Maeng <h.maeng4@lancaster.ac.uk>, Piotr Fryzlewicz <p.fryzlewicz@lse.ac.uk>

See Also

trendsegment, thresholding, invTGUW

Examples

```
x <- c(1:10, rep(5,9))
n <- length(x)
x <- x + rnorm(n)
tguwfit <- TGUW(x)
tguwfit
```

---

thresholding — *Noise reduction from the sequence of detail coefficients returned by the Tail-Greedy Unbalanced Wavelet (TGUW) transformation*

Description

This function is used inside trendsegment and performs the thresholding of the detail coefficients returned by the Tail-Greedy Unbalanced Wavelet (TGUW) transformation. The denoising is achieved by a prespecified threshold in a "connected" way in that it prunes the branches if and only if the detail coefficient itself and all of its children coefficients are below some thresholds in its size. Also, the "two together" rule is applied to any paired detail coefficients returned by Type 3 merging (merging two sets of paired smooth coefficients) in the sense that both detail coefficients should be survived if at least one of their size is over threshold. For details, see H. Maeng and P. Fryzlewicz (2021), Detecting linear trend changes in data sequences, preprint.

Usage

```
thresholding(ts.obj, lambda, minsegL, bal = 0, connected = FALSE)
```

Arguments

`ts.obj`	A list returned by `TGUW`.
`lambda`	The magnitude of the threshold. It has a form of `sigma * th.const * sqrt(2 * log(n))` where `n` is the length of input data `x`, the default of `th.const` is 1.3 and the `sigma` can be estimated by Median Absolute Deviation (MAD) method under the Gaussian assumption for noise.
`minsegL`	The minimum segment length of estimated signal returned by `trendsegment`.
`bal`	The minimum ratio of the length of the shorter region to the length of the entire merging region especially when the merges of Type 2 (merging one initial and a paired smooth coefficient) or of Type 3 (merging two sets of (paired) smooth coefficients) are performed. Only triplets which satisfy this balancedness condition survives in denoising. Point anomalies can be detected only if `bal < 1/n` and `minsegL = 1`. The default is set to 0.
`connected`	If connected=TRUE, the thresholding puts the connected rule above the `minsegL`, otherwise it makes keeping the `minsegL` a priority.

Value

ts.obj The modified ts.obj containing zero detail coefficients in the `merging.hist` if not survived from thresholding.

Author(s)

Hyeyoung Maeng <h.maeng4@lancaster.ac.uk>, Piotr Fryzlewicz <p.fryzlewicz@lse.ac.uk>

See Also

`trendsegment`, `TGUW`, `invTGUW`

Examples

```
x <- c(1:10, rep(5,9))
n <- length(x)
x <- x + rnorm(n)
tguwfit <- TGUW(x)
th.const <- 1.3
lambda <- (stats::mad(diff(diff(x)))/sqrt(6)) * sqrt(2 * log(n)) * th.const
thrfit <- thresholding(ts.obj = tguwfit, lambda = lambda, minsegL = 5, bal = 0, connected = FALSE)
thrfit
```

`trendsegment` *Detecting linear trend changes for univariate time series*

Description

The main function of the package `trendsegmentR`. This function estimates the number and locations of change-points in linear trend of noisy data. The estimated change-points may contain point anomalies (segments including only one data point) if any. It also returns the estimated signal, the best linear fit for each segment between a pair of adjacent change-points. The algorithm includes three steps, Tail-Greedy Unbalanced Wavelet (TGUW) transform (`TGUW`), thresholding (`thresholding`) and inverse TGUW transform (`invTGUW`).

Usage

```
trendsegment(
  x,
  th.const = 1.3,
  p = 0.01,
  bal = 0,
  minsegL = 5,
  continuous = FALSE,
  connected = FALSE
)
```

Arguments

x	A data vector to be examined for change-point detection.
th.const	Thresholding parameter used in thresholding. The default is 1.3 and the exact magnitude of the threshold is sigma * th.const * sqrt(2 * log(n)) where n is the length of data sequence x and sigma is estimated by Median Absolute Deviation (MAD) method under the Gaussian assumption for noise.
p	Proportion of all possible remaining merges which specifies the number of merges allowed in a single pass over the data. This is used in TGUW and the default is 0.01.
bal	The minimum ratio of the length of the shorter region to the length of the entire merging region especially when the merges of Type 2 (merging one initial and a paired smooth coefficient) or of Type 3 (merging two sets of (paired) smooth coefficients) are performed. If bal < 1/n and minsegL = 1, point anomalies (segments including only one data point) can possibly be detected, otherwise, the detection of point anomalies is not guaranteed. The default is set to 0.
minsegL	The minimum segment length of estimated signal returned by trendsegment. The default is set to 5 for avoiding too frequent change-points.
continuous	If continuous=TRUE, the estimated signal returned by trendsegment is continuous at change-points, otherwise discontinuous at change-points. The default is set to FALSE.
connected	If connected=TRUE, the trendsegment algorithm puts the connected rule above the minsegL, otherwise it makes keeping the minsegL a priority. The default is set to FALSE.

Details

The algorithm is described in H. Maeng and P. Fryzlewicz (2021), Detecting linear trend changes in data sequences, preprint.

Value

A list with the following.

x	The original input vector x.
est	The estimated piecewise-linear signal of x.
no.of.cpt	The estimated number of change-points.
cpt	The estimated locations of change-points.

Author(s)

Hyeyoung Maeng <h.maeng4@lancaster.ac.uk>, Piotr Fryzlewicz <p.fryzlewicz@lse.ac.uk>

See Also

TGUW, thresholding, invTGUW

Examples

```
x <- c(rep(0,100), seq(0, 4, length.out = 100), rep(3, 100), seq(3, -1, length.out=99))
n <- length(x)
x <- x + rnorm(n)
tsfit <- trendsegment(x = x, bal = 0)
tsfit

plot(x, type = "b", ylim = range(x, tsfit$est))
lines(tsfit$est, col=2, lwd=2)
```

`trendsegmentR` *trendsegmentR: A device for detecting multiple change-points corresponding to linear trend changes in one dimensional data*

Description

Performs the detection of linear trend changes by transforming the data through an adaptively constructed unbalanced wavelet basis in a hierarchical way. To start with, see the function `trendsegment`.

Author(s)

Hyeyoung Maeng <h.maeng4@lancaster.ac.uk>, Piotr Fryzlewicz <p.fryzlewicz@lse.ac.uk>

References

H. Maeng and P. Fryzlewicz (2021), Detecting linear trend changes in data sequences, preprint.

See Also

`trendsegment`, `TGUW`, `thresholding`, `invTGUW`

Index