Package ‘robcp’

January 24, 2020

Title Robust Change-Point Tests

Version 0.2.5

Description Provides robust methods to detect change-points in uni- or multivariate time series. They can cope with corrupted data and heavy tails. One can detect changes in location, scale and dependence structure of a possibly multivariate time series. Procedures are based on Huberized versions of CUSUM tests proposed in Duerre and Fried (2019) <arXiv:1905.06201>.

Depends R (>= 3.3.1)

License GPL-3

Encoding UTF-8

LazyData true

NeedsCompilation yes

Author Sheila Goerz [aut, cre],
Alexander Duerre [ctb]

Maintainer Sheila Goerz <sheila.goerz@tu-dortmund.de>

Imports methods

Repository CRAN

Date/Publication 2020-01-24 15:00:02 UTC

R topics documented:

huber_cusum *Huberized CUSUM test*

Description

Performs a CUSUM test on data transformed by psi. Depending in the choosen psi-function different types of changes can be detected.

Usage

```
huber_cusum(x, fun = "HLm", tol = 1e-8, b_n, k, constant)
```

Arguments

x	numeric vector containing a single time series or a numeric matrix containing multiple time series (column-wise).
fun	character string specifiyng the transformation function ψ, see details.
tol	tolerance of the distribution function (numeric), which is used do compute p-values.
b_n	bandwidth, which is used to estimate the long run variance, see the help page of sigma2 for details.
k	numeric bound used in psi.
constant	scale factor of the MAD. Default is 1.4826.

Details

The function performs a Huberized CUSUM test. First the data is transformed by a suitable psi-function. To detect changes in location one can apply fun = "HLm", "HLg", "VLm" or "VLg", for changes in scale fun = "HCm" is avaliable and for changes in the dependence respectively covariance structure fun = "HCm", "HCg", "VCm" and "VCg" are possible. The actual definitions of the psi-functions can be found in the help page of psi. Denote $Y_1, \ldots, Y_n$ the transformed time series. If Y_1 is one dimensional, then the teststatistik

$$V = \max_{k=1,\ldots,n} \frac{1}{\sqrt{n}\sigma} |\sum_{i=1}^{k} Y_i - \frac{k}{n} \sum_{i=1}^{n} Y_i|$$

is calculated, where σ^2 is an estimator for the long run variance, see the help function of sigma2 for details. V is asymptoticaly Kolmogorov-Smirnov distributed. We use a finite sample correction $V + 0.58/\sqrt{n}$ to improve finite sample performance.
If $Y[1]$ is multivariate, then the test statistic

$$W = \max_{k=1,\ldots,n} \frac{1}{n} (\sum_{i=1}^{k} Y_i - \frac{k}{n} \sum_{i=1}^{n} Y_i)^T \Sigma^{-1} (\sum_{i=1}^{k} Y_i - \frac{k}{n} \sum_{i=1}^{n} Y_i)$$

is computed, where Σ is the long run covariance, see also sigma2 for details. W is asymptotically distributed like the maximum of a squared Bessel bridge. We use the identity derived in Kiefer to derive p-values. Like in the one dimensional case we use a finite sample correction $(\sqrt{W} + 0.58/\sqrt{n})^2$.

Value

A list of the class "htest" containing the following compontents:

`statistic`	value of the test statistic (numeric).
`p.value`	p-value (numeric).
`alternative`	alternative hypothesis (character string).
`method`	name of the performed test(character string).
`data.name`	name of the data (character string).

Author(s)

Sheila Görz

References

Dürre, A. and Fried, R. (2019). "Robust change point tests by bounded transformations", https://arxiv.org/abs/1905.06201

Kiefer, J. (1959). "K-sample analogues of the Kolmogorov-Smirnov and Cramer-V. Mises tests", *The Annals of Mathematical Statistics*, 420–447.

See Also

`sigma2`, `psi`, `h_cumsum`, `teststat`, `pKSdist`

Examples

```
set.seed(1895)

#time series with a structural break at t = 20
Z <- c(rnorm(20, 0), rnorm(20, 2))
huber_cusum(Z)

# two time series with a structural break at t = 20
timeSeries <- matrix(c(rnorm(20, 0), rnorm(20, 2), rnorm(20, 1), rnorm(20, 3),
                     ncol = 2))

huber_cusum(timeSeries)
```

`h_cumsum` *Cumulative sum of transformed vectors*

Description

Computes the cumulative sum of a transformed numeric vector or matrix. Default transformation is `psi`.

Usage

```
h_cumsum(y, fun = "HLm", k, constant)
```

Arguments

`y`	numeric vector containing a single time series or a numeric matrix containing multiple time series (column-wise).
`fun`	character string specifiyng the transformation function ψ.
`k`	numeric bound used in `psi`.
`constant`	scale factor of the MAD. Default is 1.4826.

Details

Prior to computing the sums, y is being transformed by the function `fun`.

Value

Numeric vector or matrix containing the cumulative sums of the transformed values. In case of a matrix, cumulative sums are being computed for each time series (column) independentely.

Author(s)

Sheila Görz

See Also

`psi`.

Examples

```
h_cumsum(rnorm(100))
```

---

kthPair *K-th largest element in a sum of sets.*

Description

Selects the k-th largest element of X + Y, a sum of sets. X + Y denotes the set $\{x+y|x \in X, y \in Y\}$.

Usage

```
kthPair(x, y, k)
```

Arguments

`x`	Numeric vector.
`y`	Numeric vector.
`k`	Index of element to be selected. Must be an integer and between 1 and the product of the lengths of x and y.

Value

K-th largest value (numeric).

Author(s)

Sheila Görz

References

Johnson, D. B., & Mizoguchi, T. (1978). Selecting the K-th Element in X+Y and X_1+X_2+ ... +X_m. SIAM Journal on Computing, 7(2), 147-153.

Examples

```
x <- rnorm(100)
y <- runif(100)

kthPair(x, y, 5000)
```

---

modifChol — *Revised Modified Cholesky Factorization*

Description

Computes the revised modified Cholesky factorization described in Schnabel and Eskow (1999).

Usage

```
modifChol(x, tau = .Machine$double.eps^(1 / 3),
          tau_bar = .Machine$double.eps^(2 / 3), mu = 0.1)
```

Arguments

`x`	a symmetric matrix.
`tau`	(machine epsilon)^(1/3).
`tau_bar`	(machine epsilon^(2/3)).
`mu`	numeric, $0 < \mu \leq 1$.

Details

`modif.chol` computes the revised modified Cholesky Factorization of a symmetric, not neccessarily positive definite matrix x + E such that $LL' = x + E$ for $E \geq 0$.

Value

Lower triangular matrix L of the form $LL' = x + E$. The attribute `swaps` is a vector of the lenght of dimension of x. It cointains the indices of the rows and columns that were swapped in x in order to compute the modified Cholesky factorization. For example if the `i`-th element of swaps is the number `j`, then the `i`-th and the `j`-th row and column were swapped. To reconstruct the original matrix swaps has to be read backwards.

Author(s)

Sheila Görz

References

Schnabel, R. B., & Eskow, E. (1999). "A revised modified Cholesky factorization algorithm" SIAM Journal on optimization, 9(4), 1135-1148.

Examples

```
y <- matrix(runif(9), ncol = 3)
x <- psi(y)
modifChol(sigma2(x))
```

---

pKSdist — *Asymptotic cumulative distribution for the Huberized CUSUM Test statistic*

Description

Computes the asymptotic cumulative distribution of the statistic of `teststat`.

Usage

```
pKSdist(tn, tol = 1e-8)
pbessel3(tn, h)
```

Arguments

`tn`	vector of test statistics (numeric). For `pbessel3` length of `tn` has to be 1.
`h`	dimension of time series (integer). If `h` is equal to 1 `pbessel3` uses `pKS2` to compute the corresponding probability.
`tol`	tolerance (numeric).

Details

For a single time series, the distribution is the same distribution as in the two sample Kolmogorov Smirnov Test, namely the distribution of the maximal value of the absolute values of a Brownian bridge. It is computated as follows (van Mulbregt, 2018):

For $t_n(x) < 1$:

$$P(t_n(X) \leq t_n(x)) = \sqrt{2 * \pi}/t_n(x) * t(1 + t^8(1 + t^{16}(1 + t^{24}(1 + ...))))$$

up to $t^{8*k_{max}}, k_{max} = \lfloor \sqrt{2 - \log(tol)} \rfloor$ where $t = \exp(-\pi^2/(8 * x^2))$

else:

$$P(t_n(X) \leq t_n(x)) = 2 * \sum_{k=1}^{\infty} (-1)^{k-1} * \exp(-2 * k^2 * x^2)$$

until $|2 * (-1)^{k-1} * \exp(-2 * k^2 * x^2) - 2 * (-1)^{(k-1)-1} * \exp(-2 * (k-1)^2 * x^2)| \leq tol$.

In case of multiple time series, the distribution equals that of the maximum of an h dimensional squared Bessel bridge. It can be computed by (Kiefer, 1959):

$$P(t_n(X) \leq t_n(x)) = (4/(\Gamma(h/2)2^{h/2}t_n^h)) \sum_{i=1}^{\infty} (((\gamma_{(h-2)/2,n})^{h-2} \exp(-(\gamma_{(h-2)/2,n})^2/(2t_n^2)))/(J_{h/2}(\gamma_{(h-2)/2,n}))^2)$$

where J_h is the Bessel function of first kind and h-th order, Γ is the gamma function and $\gamma_{h,n}$ denotes the n-th zero of J_h.

Value

vector of $P(t_n(X) \leq tn[i])$.

Author(s)

Sheila Görz, Alexander Dürre

References

van Mulbregt, P. (2018) "Computing the Cumulative Distribution Function and Quantiles of the limit of the Two-sided Kolmogorov-Smirnov Statistic." arXiv preprint arXiv:1803.00426.

/src/library/stats/src/ks.c rev60573

Kiefer, J. (1959). "K-sample analogues of the Kolmogorov-Smirnov and Cramer-V. Mises tests", *The Annals of Mathematical Statistics*, 420–447.

See Also

psi, teststat, h_cumsum, huber_cusum

Examples

```
# single time series
timeSeries <- c(rnorm(20, 0), rnorm(20, 2))
tn <- teststat(timeSeries)

pKSdist(tn)

# two time series
timeSeries <- matrix(c(rnorm(20, 0), rnorm(20, 2), rnorm(20, 1), rnorm(20, 3),
                     ncol = 2))
tn <- teststat(timeSeries)

pbessel3(tn, 2)
```

---

psi *Transformation of time series*

Description

Computation of values tranformed by their median, MAD and a ψ function.

Usage

```
psi(y, fun = "HLm", k, constant = 1.4826)
```

Arguments

`y`	vector or matrix with each column representing a time series (numeric).
`fun`	character string specifiyng the transformation function ψ.
`k`	numeric bound used for Huber type psi-functions which determines robustness and efficiency of the test. Default for `psi = "HLg"` or `"HCg"` is `sqrt(qchisq(0.8,df = m)` where `m` are the number of time series, and otherwise it is 1.5.
`constant`	scale factor of the MAD.

Details

Let $x = (y - Median(y))/MAD(y)$ be the standardized values of a single time series.

Available ψ functions are:

marginal Huber for location:
`fun = "HLm"`.
$\psi_{HLm}(x) = k * 1_{x>k} + z * 1_{-k \leq x \leq k} - k * 1_{x<-k}$.

global Huber for location:
`fun = "HLg"`.

$\psi_{HLg}(x) = x * 1_{0<|x|\leq k} + k * x/|x| * 1_{|x|>k}$.

marginal sign for location:
`fun = "VLm"`.
$\psi_{VLm}(x_i) = sign(x_i)$.

global sign for location:
`fun = "VLg"`.
$\psi_{VLg}(x) = x/|x| * 1_{|x|>0}$.

marginal Huber for covariance:
`fun = "HCm"`.
$\psi_{HCm}(x) = \psi_{HLm}(x)\psi_{HLm}(x)^T$.

global Huber for covariance:
`fun = "HCg"`.
$\psi_{HCg}(x) = \psi_{HLg}(x)\psi_{HLg}(x)^T$.

marginal sign covariance:
`fun = "VCm"`.
$\psi_{VCm}(x) = \psi_{VLm}(x)\psi_{VLm}(x)^T$.

gloabl sign covariance:
`fun = "VCg"`.
$\psi_{VCg}(x) = \psi_{VCg}(x)\psi_{VCg}(x)^T$.

Note that for all covariances only the upper diagonal is used and turned into a vector. In case of the marginal sign covariance, the main diagonal is also left out. At the global sign covariance matrix the last element of the main diagonal is left out.

Value

Transformed numeric vector or matrix with the same number of rows as `y`.

Author(s)

Sheila Görz

See Also

`h_cumsum`, `teststat`

Examples

```
psi(rnorm(100))
```

sigma2 *Autocorrelation*

Description

Estimates the long run variance respectively covariance matrix of the supplied time series.

Usage

```
sigma2(x, b_n)
```

Arguments

x	vector or matrix with each column representing a time series (numeric).
b_n	Must be greater than 0. default is $n^{1/3}$ with n being the number of observations.

Details

The long run variance equals n times the asymptotic variance of the arithmetic mean of a short range dependent time series, where n is the length of the time series. It is used to standardize CUSUM Tests.

The long run variance is estimated by a kernel estimator using the bandwidth $b_n = n^{1/3}$ and the flat top kernel

$$k(x) = x * 1_{|x|<0.5} + (2 - |x|) * 1_{0.5<|x|<1}$$

. In the one dimensional case this results in:

$$\hat{\sigma}^2 = (1/n) \sum_{i=1}^{n} (x[i] - mean(x))^2 + (2/n) \sum_{h=1}^{b_n} \sum_{i=1}^{n-h} (x[i] - mean(x)) * (x[i+h] - mean(x)) * k(h/b_n).$$

If x is a multivariate timeseries the k, l-element of Σ is estimated by

$$\hat{\Sigma}^{(k,l)} = (1/n) \sum_{i,j=1}^{n} (x[i]^{(k)} - mean(x)^{(k)}) * (x[j]^{(l)} - mean(x)^{(l)}) * k((i-j)/b_n).$$

Value

long run variance $sigma^2$ respectively Σ (numeric)

Author(s)

Sheila Görz

See Also

psi, h_cumsum, teststat, pKSdist, huber_cusum

Examples

```
Z <- c(rnorm(20), rnorm(20, 2))
sigma2(Z)
```

`teststat` *Test statistic for the Huberized CUSUM Test*

Description

Computes the test statistic for a structural break test called 'Huberized CUSUM Test'.

Usage

```
teststat(y, fun = "HLm", b_n, k, constant)
```

Arguments

`y`	vector or matrix with each column representing a time series (numeric).
`fun`	character string specifiyng the transformation function ψ.
`b_n`	for `sigma2`.
`k`	numeric bound used in `psi`.
`constant`	scale factor of the MAD. Default is 1.4826.

Details

y is transformed by `fun`. Let x be the resulting vector or matrix and n be the length of a time series.

In case of a vector the test statistic can be written as

$$max_{k=1,...,n} \frac{1}{\sqrt{n}\sigma} |\sum_{i=1}^{k} x_i - (k/n) \sum_{i=1}^{n} x_i|,$$

where σ is the square root of `sigma2`.

In case of a matrix the test statistic follows as

$$max_{k=1,...,n} \frac{1}{n} (\sum_{i=1}^{k} X_i - \frac{k}{n} \sum_{i=1}^{n} X_i)^T \Sigma^{-1} (\sum_{i=1}^{k} X_i - \frac{k}{n} \sum_{i=1}^{n} X_i),$$

where X_i denotes the ith row of x and Σ^{-1} is the inverse of `sigma2`.

Value

test statistic (numeric value).

Author(s)

Sheila Görz

See Also

h_cumsum, psi

Examples

```
# time series with structural break at t = 20
ts <- c(rnorm(20, 0), rnorm(20, 2))

teststat(ts)
```

---

zeros — *Zero of the Bessel function of first kind*

Description

Contains the zeros of the Bessel function of first kind.

Usage

```
data("zeros")
```

Format

A data frame where the ith column contains the first 50 zeros of the Bessel function of the first kind and ((i - 1) / 2)th order, i = 1, ..., 5001.

Source

The zeros are computed by the mathematical software octave.

References

Eaton, J., Bateman, D., Hauberg, S., Wehbring, R. (2015). "GNU Octave version 4.0.0 manual: a high-level interactive language for numerical computations".

Index