Package ‘npcp’
July 24, 2015

**Type** Package

**Title** Some Nonparametric Tests for Change-Point Detection in Possibly Multivariate Observations

**Version** 0.1-6

**Date** 2015-07-23

**Author** Ivan Kojadinovic

**Maintainer** Ivan Kojadinovic <ivan.kojadinovic@univ-pau.fr>

**Imports** stats

**Suggests** copula

**Description** Provides nonparametric tests for assessing whether possibly serially dependent univariate or multivariate observations have the same c.d.f. or not. In addition to tests focusing directly on the c.d.f., the package contains tests designed to be particularly sensitive to changes in the underlying copula, Spearman's rho or certain quantities that can be estimated using one-sample U-statistics of order two such as the variance, Gini's mean difference or Kendall's tau. The latest addition is a nonparametric test for detecting changes in the distribution of independent block maxima.

**License** GPL (>= 3) | file LICENCE

**LazyLoad** yes

**Encoding** UTF-8

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2015-07-24 08:41:09

**R topics documented:**

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>bOptEmpProc</td>
<td>2</td>
</tr>
<tr>
<td>cpTestBM</td>
<td>3</td>
</tr>
<tr>
<td>cpTestCn</td>
<td>5</td>
</tr>
<tr>
<td>cpTestFn</td>
<td>7</td>
</tr>
<tr>
<td>cpTestRho</td>
<td>10</td>
</tr>
<tr>
<td>cpTestU</td>
<td>12</td>
</tr>
</tbody>
</table>

**Index** 15
Description

In a particular empirical process setting, estimates the bandwidth parameter controlling the serial dependence when generating dependent multiplier sequences using the 'moving average approach'; see Section 5 of the third reference. This function is called in the functions `cpTestFn()` and `cpTestCn()` if b is set to NULL.

Usage

```r
bOptEmpProc(x, m=5, weights = c("parzen", "bartlett"),
             L.method=c("max","median","mean","min"))
```

Arguments

- `x`: a data matrix whose rows are continuous observations.
- `weights`: a string specifying the kernel for creating the weights used in the generation of dependent multiplier sequences within the 'moving average approach'; see Section 5 of the third reference.
- `m`: a strictly positive integer specifying the number of points of the uniform grid on \((0,1)^d\) (where \(d = ncol(x)\)) involved in the estimation of the bandwidth parameter; see Section 5 of the third reference. The number of points of the grid is given by \(m^{ncol(x)}\) so that \(m\) needs to be decreased as \(d\) increases.
- `L.method`: a string specifying how the parameter \(L\) involved in the estimation of the bandwidth parameter is computed; see Section 5 of the third reference.

Details

The implemented approach results from an adaptation of the procedure described in the first two references (see also the references therein). The use of this function in a context different from that considered in the third reference may not be meaningful.

Acknowledgment: Part of the code of the function results from an adaptation of R code of C. Parmeter and J. Racine, itself an adaptation of Matlab code by A. Patton.

Value

A strictly positive integer.

References


See Also

`cpTestFn()`, `cpTestCn()`.

cpTestBM

**Nonparametric tests for change-point detection in the distribution of independent block maxima**

**Description**

Nonparametric tests for change-point detection in the distribution of independent block maxima based either on the probability weighted moment method (see the second reference) or on the generalized probability weighted moment method (see the first reference) for estimating the parameters of the generalized extreme value (GEV) distribution. It is assumed that the block maxima are independent and that their unknown c.d.f.s are continuous, but not necessarily that they are GEV distributed. Three statistics are computed. Under the assumption that the block maxima are GEV distributed, these are statistics particularly sensitive to changes in the location, scale and shape parameters of the GEV. Details can be found in third reference.

**Usage**

```r
cpTestBM(x, method = c("pwm", "gpwm"), r=10)
```

**Arguments**

- `x` a numeric vector representing independent block maxima whose unknown c.d.f.s are assumed continuous.
- `method` a string specifying how statistics will be defined; can be either "pwm" (the probability weighted moment method) or "gpwm" (the generalized probability weighted moment method). The method "pwm" is suggested for climate block maxima that are typically not too heavy tailed, more precisely, whose distributions are in the maximum domains of attraction of GEV distributions with shape parameters smaller than a half. The method "gpwm" should be preferred otherwise.
- `r` strictly positive integer specifying the set of breakpoints that will be tested; more precisely, starting from the initial sample of block maxima, the tests compare subsamples formed by the k first maxima and n-k last maxima for k in the set \{r, \ldots, n-r\}, where n is the sample size.

**Details**

Approximate p-values are computed from the estimated asymptotic null distributions, which involve the Kolmogorov distribution. The latter is dealt with reusing code from the `ks.test()` function; credit to RCore.
Value

An object of class htest which is a list, some of the components of which are

- statistic: value of the three test statistics.
- pvalues: corresponding approximate p-values.
- stats.loc: the values of the \( n - (2 \times r - 1) \) intermediate change-point statistics sensitive to changes in the location; the first test statistic is defined as the maximum of those.
- stats.scale: the values of the \( n - (2 \times r - 1) \) intermediate change-point statistics sensitive to changes in the scale; the second test statistic is defined as the maximum of those.
- stats.shape: the values of the \( n - (2 \times r - 1) \) intermediate change-point statistics sensitive to changes in the shape; the third test statistic is defined as the maximum of those.

Note

The tests were derived under the assumption of block maxima whose unknown c.d.f.s are continuous, which implies that ties occur with probability zero. A way to deal with ties based on randomization is proposed in the third reference.

References


See Also

cpTestFn() for a related test based on the empirical c.d.f.

Examples

```r
e <- 100
k <- 50 ## the true change-point

## change in the shape parameter of a GEV
x <- rgev(k, loc=0, scale=1, shape=-0.8)
y <- rgev(k, loc=0, scale=1, shape=0.4)
cp <- cpTestBM(c(x,y))
cp
## estimated change-point
```

which(cp$stats.shape == max(cp$stats.shape))

## change in the scale parameter of a GEV
x <- rgev(k,loc=0,scale=0.5,shape=0)
y <- rgev(k,loc=0,scale=1,shape=0)
cp <- cpTestBM(c(x,y))
cp
## estimated change-point
which(cp$stats.scale == max(cp$stats.scale))

## change in the location parameter of a GEV
x <- rgev(k,loc=0,scale=1,shape=0)
y <- rgev(k,loc=0.5,scale=1,shape=0)
cp <- cpTestBM(c(x,y))
cp
## estimated change-point
which(cp$stats.loc == max(cp$stats.loc))

## End(Not run)

cpTestCn

Test for change-point detection based on the empirical copula

Description

Nonparametric test for change-point detection particularly sensitive to changes in the copula of multivariate continuous observations. The observations can be serially independent or dependent (strongly mixing). Approximate p-values for the test statistic are obtained by means of a multiplier approach. Details can be found in first reference.

Usage

cpTestCn(x, method = c("seq", "nonseq"), b = 1,
weights = c("parzen", "bartlett"), m = 5,
L.method=c("max","median","mean","min"),
N = 1000, init.seq = NULL)

Arguments

x a data matrix whose rows are multivariate continuous observations.
method a string specifying the simulation method for generating multiplier replicates of the test statistic; can be either "seq" (the 'check' approach in the first reference) or "nonseq" (the 'hat' approach in the first reference). The 'check' approach appears to lead to better behaved tests in the case of samples of moderate size. The 'hat' approach is substantially faster.
b strictly positive integer specifying the value of the bandwidth parameter determining the serial dependence when generating dependent multiplier sequences using the 'moving average approach'; see Section 5 of the second reference.
The default value is 1, which will create i.i.d. multiplier sequences suitable for serially independent observations. If set to NULL, b will be estimated from x using the function bOptEmpProc(); see the procedure described in Section 5 of the second reference.

weights a string specifying the kernel for creating the weights used in the generation of dependent multiplier sequences within the 'moving average approach'; see Section 5 of the second reference.

m a strictly positive integer specifying the number of points of the uniform grid on \((0,1)^d\) (where \(d = \text{ncol}(x)\)) involved in the estimation of the bandwidth parameter; see Section 5 of the third reference. The number of points of the grid is given by \(m^{\text{ncol}(x)}\) so that \(m\) needs to be decreased as \(d\) increases.

L.method a string specifying how the parameter \(L\) involved in the estimation of the bandwidth parameter is computed; see Section 5 of the second reference.

N number of multiplier replications.

init.seq a sequence of independent standard normal variates of length \(N \times (\text{nrow}(x) + R \times (b - 1))\) used to generate dependent multiplier sequences.

Details

The approximate p-value is computed as

\[
(0.5 + \sum_{i=1}^{N} 1_{\{S_i \geq S\}})/(N + 1),
\]

where \(S\) and \(S_i\) denote the test statistic and a multiplier replication, respectively. This ensures that the approximate p-value is a number strictly between 0 and 1, which is sometimes necessary for further treatments.

Value

An object of class htest which is a list, some of the components of which are

- statistic value of the test statistic.
- p.value corresponding approximate p-value.
- cvm the values of the \(\text{nrow}(x)-1\) intermediate Cramér-von Mises change-point statistics; the test statistic is defined as the maximum of those.
- b the value of parameter b.

Note

These tests were derived under the assumption of continuous margins.

References


See Also

\texttt{cpTestRho()} for a related test based on Spearman’s rho, \texttt{cpTestU()} for related tests based on U-statistics, \texttt{cpTestFn()} for a related test based on the multivariate empirical c.d.f., \texttt{bOptEmpProc()} for the function used to estimate b from x if b = NULL.

Examples

\begin{verbatim}
## Not run:
require/copula)
  n <- 100
  k <- 50 ## the true change-point
  u <- rCopula(k, gumbelCopula(1.5))
  v <- rCopula(n-k, gumbelCopula(3))
  x <- rbind(u, v)
  cp <- cpTestCn(x)
  cp
  ## estimated change-point
  which(cp$cvm == max(cp$cvm))

## End(Not run)
\end{verbatim}

\textbf{cpTestFn} \hspace{2cm} Test for change-point detection based on the multivariate empirical distribution function

\textbf{Description}

Nonparametric test for change-point detection based on the (multivariate) empirical distribution function. The observations can be continuous univariate or multivariate, and serially independent or dependent (strongly mixing). Approximate p-values for the test statistics are obtained by means of a multiplier approach. Details can be found in the first reference which treats the serially independent case.

\textbf{Usage}

\texttt{cpTestFn(x, statistic = c("cvmmax", "cvmmean", "ksmax", "ksmean"),
  method = c("nonseq", "seq"), b = 1,
  weights = c("parzen", "bartlett"),
  m = 5, L.method=c("max","median","mean","min"),
  N = 1000, init.seq = NULL)}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{x} \hspace{1cm} a data matrix whose rows are continuous observations.
  \item \texttt{statistic} \hspace{1cm} a string specifying the statistic whose value and p-value will be displayed; can
    be either "cvmmax" or "cvmmean" (the maximum or average of the nrow(x)-1
intermediate Cramér-von Mises statistics), or "ksmax" or "ksmean" (the maximum or average of the \(nrow(x)-1\) intermediate Kolmogorov-Smirnov statistics); see Section 3 in the first reference. The four statistics and the corresponding p-values are computed at each execution.

**method**
a string specifying the simulation method for generating multiplier replicates of the test statistic; can be either "nonseq" (the 'check' approach in the first reference) or "seq" (the 'hat' approach in the first reference). The 'check' approach appears to lead to better behaved tests and is recommended.

**b**
strictly positive integer specifying the value of the bandwidth parameter determining the serial dependence when generating dependent multiplier sequences using the 'moving average approach'; see Section 5 of the second reference. The default value is 1, which will create i.i.d. multiplier sequences suitable for serially independent observations. If set to NULL, b will be estimated from x using the function \(\text{bOptEmpProc}\); see the procedure described in Section 5 of the second reference.

**weights**
a string specifying the kernel for creating the weights used in the generation of dependent multiplier sequences within the 'moving average approach'; see Section 5 of the second reference.

**m**
a strictly positive integer specifying the number of points of the uniform grid on \((0,1)^d\) (where \(d = ncol(x)\)) involved in the estimation of the bandwidth parameter; see Section 5 of the third reference. The number of points of the grid is given by \(m \times ncol(x)\) so that \(m\) needs to be decreased as \(d\) increases.

**L.method**
a string specifying how the parameter \(L\) involved in the estimation of the bandwidth parameter is computed; see Section 5 of the second reference.

**N**
number of multiplier replications.

**init.seq**
a sequence of independent standard normal variates of length \(N \times (nrow(x) + 2 \times (b - 1))\) used to generate dependent multiplier sequences.

### Details

The approximate p-value is computed as

\[(0.5 + \sum_{i=1}^{N} 1\{S_i \geq S_I\})/(N + 1),\]

where \(S\) and \(S_i\) denote the test statistic and a multiplier replication, respectively. This ensures that the approximate p-value is a number strictly between 0 and 1, which is sometimes necessary for further treatments.

### Value

An object of class \texttt{h.test} which is a list, some of the components of which are

- **statistic**
  value of the test statistic.

- **p.value**
  corresponding approximate p-value.

- **cvm**
  the values of the \(nrow(x)-1\) intermediate Cramér-von Mises change-point statistics.
ks  the values of the \( nrow(x) - 1 \) intermediate Kolmogorov-Smirnov change-point statistics.

all.statistics  the values of all four test statistics.

all.p.values  the corresponding p-values.

b  the value of parameter b.

Note

Note that when the observations are continuous univariate and serially independent, independent realizations of the tests statistics under the null hypothesis of no change in the distribution can be obtained by simulation; see Section 4 in the first reference.

References


See Also

`cpTestCn()` for a related test based on the empirical copula, `cpTestRho()` for a related test based on Spearman’s rho, `cpTestU()` for related tests based on U-statistics, `bOptEmpProc()` for the function used to estimate \( b \) from \( x \) if \( b = \text{NULL} \).

Examples

```r
## a univariate example
n <- 100
k <- 50  ## the true change-point
y <- rnorm(k)
z <- rexp(n-k)
x <- matrix(c(y,z))
cp <- cpTestFn(x)
cp

## all statistics
cp$all.statistics

## corresponding p-values
cp$all.p.values

## estimated change-point
which(cp$cvm == max(cp$cvm))
which(cp$ks == max(cp$ks))

## a very artificial trivariate example
## with a break in the first margin
n <- 100
k <- 50  ## the true change-point
y <- rnorm(k)
```
### cpTestRho

Test for change-point detection based on Spearman's rho

**Description**

Nonparametric test for change-point detection particularly sensitive to changes in Spearman's rho in multivariate time series. The observations can be serially independent or dependent (strongly mixing). Approximate p-values for the test statistic are obtained by means of a multiplier approach or by estimating the asymptotic null distribution. Details can be found in first reference.

**Usage**

```r
cpTestRho(x, method = c("mult", "asym.var"),
          statistic = c("pairwise", "global"),
          b = 1, weights = c("parzen", "bartlett"),
          N = 1000, init.seq = NULL)
```

**Arguments**

- **x**: a data matrix whose rows are multivariate continuous observations.
- **method**: a string specifying the method for computing the approximate p-value for the test statistic; can be either "mult" (the multiplier approach 'tilde' in the first reference) or "asym.var" (the approach based on the estimation of the asymptotic null distribution of the test statistic described in the first reference). The 'mult' approach appears to lead to better behaved tests.
- **statistic**: a string specifying the test statistic; can be either "pairwise" (the statistic $S_{n,3}$ in the first reference) or "global" (the statistic $S_{n,1}$ in the first reference).
- **b**: strictly positive integer specifying the value of the bandwidth parameter determining the serial dependence when generating dependent multiplier sequences using the 'moving average approach'; see Section 5 of the second reference. The default value is 1, which will create i.i.d. multiplier sequences suitable for serially independent observations. If set to NULL, b will be estimated from x using the procedure described in the first reference.
weights

weights is a string specifying the kernel for creating the weights used in the generation of dependent multiplier sequences within the 'moving average approach'; see Section 5 of the second reference.

N

N is the number of multiplier replications.

init.seq

init.seq is a sequence of independent standard normal variates of length \(N \times (\text{ncol}(x) + R \times (b - 1))\) used to generate dependent multiplier sequences.

Details

When \texttt{method} == "mult", the approximate p-value is computed as

\[
(0.5 + \sum_{i=1}^{N} 1\{s_i \geq S\})/(N + 1),
\]

where \(S\) and \(s_i\) denote the test statistic and a multiplier replication, respectively. This ensures that the approximate p-value is a number strictly between 0 and 1, which is sometimes necessary for further treatments.

When \texttt{method} == "asym.var", the approximate p-value is computed from the estimated asymptotic null distribution, which involves the Kolmogorov distribution. The latter is dealt with reusing code from the \texttt{ks.test()} function; credit to RCore.

Value

An object of class \texttt{htest} which is a list, some of the components of which are

\begin{itemize}
  \item \texttt{statistic} value of the test statistic.
  \item \texttt{p.value} corresponding approximate p-value.
  \item \texttt{rho} the values of the \(nrow(x)-1\) intermediate change-point statistics; the test statistic is defined as the maximum of those.
  \item \texttt{b} the value of parameter \(b\).
\end{itemize}

Note

These tests were derived under the assumption of continuous margins.

References


See Also

\texttt{cpTestU()} for related tests based on U-statistics, \texttt{cpTestFn()} for a related test based on the multivariate empirical c.d.f., \texttt{cpTestCn()} for a related test based on the empirical copula.
Examples

```r
## Not run:
require(copula)
n <- 100
k <- 50  ## the true change-point
u <- rCopula(k, gumbelCopula(1.5))
v <- rCopula(n-k, gumbelCopula(3))
x <- rbind(u, v)
cp <- cpTestRho(x)
cp
## estimated change-point
which(cp$rho == max(cp$rho))
## End(Not run)
```

---

**cpTestU**

Some tests for change-point detection based on U-statistics

Description

Nonparametric tests for change-point detection particularly sensitive to changes in certain quantities that can be estimated using one-sample U-statistics of order two. Thus far, the quantities under consideration are the variance, Gini’s mean difference and Kendall’s tau (a generic mechanism for defining the U-statistic will be implemented in future releases). The observations can be serially independent or dependent (strongly mixing). Approximate p-values for the test statistic are obtained by means of a *multiplier* approach or by estimating the asymptotic null distribution. Details can be found in first reference.

Usage

```r
cpTestU(x, statistic = c("kendall", "variance", "gini"),
       method = c("seq", "nonseq", "asym.var"),
       b = 1, weights = c("parzen", "bartlett"),
       N = 1000, init.seq = NULL)
```

Arguments

- **x** a data matrix whose rows are continuous observations.
- **statistic** a string specifying the statistic of interest; can be either "kendall" (Kendall’s tau, in which case `ncol(x)` must be greater than one), "variance" or "gini" (the variance or Gini’s mean difference, in which case `ncol(x)` must be equal to one).
- **method** a string specifying the method for computing the approximate p-value for the test statistic: can be either "seq" (the 'check' approach in the first reference), "nonseq" (the 'hat' approach in the first reference), or "asym.var" (the approach based on the estimation of the asymptotic null distribution of the test statistic described in the first reference). The 'seq' approach appears overall
to lead to better behaved tests when statistic == "kendall". More experiments are necessary for the other two statistics.

b

strictly positive integer specifying the value of the bandwidth parameter determining the serial dependence when generating dependent multiplier sequences using the 'moving average approach'; see Section 5 of the second reference. The default value is 1, which will create i.i.d. multiplier sequences suitable for serially independent observations. If set to NULL, b will be estimated from x using the procedure described in the first reference.

weights

a string specifying the kernel for creating the weights used in the generation of dependent multiplier sequences within the 'moving average approach'; see Section 5 of the second reference.

N

number of multiplier replications.

init.seq

a sequence of independent standard normal variates of length \( N \times (\text{nrow}(x) + 2 \times (b - 1)) \) used to generate dependent multiplier sequences.

Details

When method is either "seq" or "nonseq", the approximate p-value is computed as

\[
(0.5 + \sum_{i=1}^{N} 1_{\{S_i \geq S\}})/(N + 1),
\]

where \( S \) and \( S_i \) denote the test statistic and a multiplier replication, respectively. This ensures that the approximate p-value is a number strictly between 0 and 1, which is sometimes necessary for further treatments.

When method == "asym.var", the approximate p-value is computed from the estimated asymptotic null distribution, which involves the Kolmogorov distribution. The latter is dealt with reusing code from the ks.test() function; credit to RCore.

Value

An object of class htest which is a list, some of the components of which are

- statistic
  - value of the test statistic.
- p.value
  - corresponding approximate p-value.
- u
  - the values of the \( \text{nrow}(x) - 3 \) intermediate change-point statistics; the test statistic is defined as the maximum of those.
- b
  - the value of parameter b.

Note

A generic mechanism for defining the U-statistic will be implemented in future releases.

References


See Also

cpTestFn() for a related test based on the multivariate empirical c.d.f., cpTestCn() for a related test based on the empirical copula, cpTestRho() for a related test based on Spearman’s rho.

Examples

```r
## Not run:
require/copula
n <- 100
k <- 50 ## the true change-point
u <- rCopula(k,gumbelCopula(1.5))
v <- rCopula(n-k,gumbelCopula(3))
x <- rbind(u,v)
cp <- cpTestU(x)
cp
## estimated change-point
which(cp$u == max(cp$u))

## End(Not run)
```
Index

*Topic **htest**
  cpTestBM, 3
  cpTestCn, 5
  cpTestFn, 7
  cpTestRho, 10
  cpTestU, 12

*Topic **multivariate**
  bOptEmpProc, 2
  cpTestCn, 5
  cpTestFn, 7
  cpTestRho, 10
  cpTestU, 12

*Topic **nonparametric**
  bOptEmpProc, 2
  cpTestBM, 3
  cpTestCn, 5
  cpTestFn, 7
  cpTestRho, 10
  cpTestU, 12

*Topic **ts**
  bOptEmpProc, 2
  cpTestBM, 3
  cpTestCn, 5
  cpTestFn, 7
  cpTestRho, 10
  cpTestU, 12

bOptEmpProc, 2, 6–9

class, 4, 6, 8, 11, 13
  cpTestBM, 3
  cpTestCn, 2, 3, 5, 9, 11, 14
  cpTestFn, 2–4, 7, 7, 11, 14
  cpTestRho, 7, 9, 10, 14
  cpTestU, 7, 9, 11, 12

ks.test, 3, 11, 13