Package ‘tseriesEntropy’

August 9, 2023

Title Entropy Based Analysis and Tests for Time Series

Version 0.7-2

Depends R (>= 2.14.0)

Imports cubature, methods, parallel, stats, graphics, ks

Suggests knitr, rmarkdown

LazyLoad yes

Encoding UTF-8

Description Implements an Entropy measure of dependence based on the Bhattacharya-Hellinger-Matusita distance. Can be used as a (nonlinear) autocorrelation/crosscorrelation function for continuous and categorical time series. The package includes tests for serial and cross dependence and nonlinearity based on it. Some routines have a parallel version that can be used in a multicore/cluster environment. The package makes use of S4 classes.

Maintainer Simone Giannerini <simone.giannerini@unibo.it>

License GPL (>= 2)

RoxygenNote 7.2.3

NeedsCompilation yes

Author Simone Giannerini [aut, cre] (<https://orcid.org/0000-0002-0710-668X>)

Repository CRAN

Date/Publication 2023-08-09 20:40:08 UTC

R topics documented:

`Srho` *Entropy Measure Of Serial And Cross Dependence*

Description

Entropy based measure of serial and cross dependence for integer or categorical data. Implements a normalized version of the Hellinger/Matusita distance. As shown in the references the metric measure is a proper distance.

Usage

```
Srho(x, y, lag.max, stationary = TRUE, plot = TRUE, version = c("FORTRAN","R"),
 nor = FALSE)
```

Arguments

`x, y`	integer or factor time series objects or vectors. (`y` is missing in the univariate case).
`lag.max`	maximum lag at which to calculate Srho; default is `round(N/4)` where N is the number of observations.
`stationary`	logical. If `TRUE` assumes stationarity and computes marginal probabilities by using N observations. If `FALSE` uses N-k observations where k is the lag.
`plot`	logical. If `TRUE` (the default) Srho is plotted.
`version`	either `"FORTRAN"` or `"R"`. `FORTRAN` version is the default and is preferred over the pure `R` version which is considerably slower but is included in case of portability issues.
`nor`	logical. If `TRUE` normalizes Srho with respect to its attainable maximum. Defaults to `FALSE`.

Details

Univariate version: serial entropy `Srho(x, lag.max,`
```
    stationary = TRUE, plot = TRUE, version = c("FORTRAN","R"), nor = FALSE)
```

Bivariate version: cross entropy `Srho(x, y, lag.max,`
```
    stationary = TRUE, plot = TRUE, version = c("FORTRAN","R"), nor = FALSE)
```

This implementation of the measure is normalized to take values in [0, 1]. Normalization is performed with respect to the maximum attainable value computed analytically. This makes the results of Srho comparable among different series.

Value

An object of S4 class "Srho", which is a list with the following elements:

`.Data`	vector of `lag.max` elements containing Srho computed at each lag.
`lags`	integer vector that contains the lags at which Srho is computed.
`stationary`	Object of class `"logical"`: `TRUE` if the stationary version is computed.
`data.type`	Object of class `"character"`: contains the data type.
`notes`	Object of class `"character"`: additional notes.

Warning

Unlike `ccf` the lag k value returned by `Srho(x,y)` estimates Srho between `x[t]` and `y[t+k]`. The result is returned invisibly if plot is TRUE.

Author(s)

Simone Giannerini<simone.giannerini@unibo.it>

References

Granger C. W. J., Maasoumi E., Racine J., (2004) A dependence metric for possibly nonlinear processes. *Journal of Time Series Analysis*, **25(5)**, 649–669.

Giannerini S., Maasoumi E., Bee Dagum E., (2015), Entropy testing for nonlinear serial dependence in time series, *Biometrika*, **102(3)**, 661–675 doi:10.1093/biomet/asv007.

Maasoumi E., (1993) A compendium to information theory in economics and econometrics. *Econometric Reviews*, **12(2)**, 137–181.

See Also

See Also `Srho.test`. The function `Srho.ts` implements the same measure for numeric data.

Examples

```
## UNIVARIATE VERSION
x <- as.integer(rbinom(n=20,size=4,prob=0.5))
Srho(x,lag.max=4)

## BIVARIATE VERSION
y <- as.integer(rbinom(n=20,size=4,prob=0.5))
Srho(x,y,lag.max=4)

## EXAMPLE  1: the effect of normalization
## computes the maximum attainable value by correlating x with itself

set.seed(12)
K    <- 5           # number of categories
smax <- 1-1/sqrt(K) # theoretical maximum under the uniform distribution
x    <- as.integer(sample(1:K,size=1e3,replace=TRUE)) # generates the sequence
```

```
S    <- Srho(x,x,lag.max=2,nor=FALSE,plot=FALSE)

plot(S,lwd=2,col=4)
abline(h=smax,col=2,lty=2)
text(x=-1,y=0.54,labels=paste("theoretical maximum = ",round(smax,4),sep=""),col=2)
text(x=-1,y=0.45,labels=paste("estimated maximum = ",round(S[3],4),sep=""),col=4)
```

`Srho-class` *Class "Srho"*

Description

A class for `Srho` and its extensions

Objects from the Class

Objects can be created by calls of the form `new("Srho", ...)`.

Slots

`.Data`**:** Object of class `"numeric"`: contains Srho computed on the data set.

`lags`**:** Object of class `"integer"`: contains the lags at which Srho is computed.

`stationary`**:** Object of class `"logical"`: `TRUE` if the stationary version is computed.

`data.type`**:** Object of class `"character"`: contains the data type.

`notes`**:** Object of class `"character"`: additional notes.

Methods

plot `signature(x = "Srho", y = "missing")`: ...

show `signature(object = "Srho")`: ...

Author(s)

Simone Giannerini <simone.giannerini@unibo.it>

See Also

See Also `Srho.test`

Examples

```
showClass("Srho")
```

Srho.test — *Entropy Test For Serial And Cross Dependence For Categorical Sequences*

Description

Bootstrap/permutation tests of serial and cross dependence for integer or categorical sequences.

Usage

```
Srho.test(x, y, lag.max=10, B = 1000, stationary = TRUE, plot = TRUE,
quant = c(0.95, 0.99), nor = FALSE)
```

Arguments

`x, y`	integer or factor time series objects or vectors. (`y` is missing in the univariate case).
`lag.max`	maximum lag at which to calculate Srho; the default is 10.
`B`	number of bootstrap/permutation replications.
`stationary`	logical. If `TRUE` assumes stationarity and computes marginal probabilities by using all the N observations. If `FALSE` uses N-k observations where k is the lag.
`plot`	logical. If `TRUE`(the default) produces a plot of Srho together with permutation confidence bands under the null hypothesis of independence.
`quant`	quantiles to be specified for the computation of the significant lags and the plot of confidence bands. Up to 2 quantiles can be specified. Defaults are 95% and 99%.
`nor`	logical. If `TRUE` normalizes Srho with respect to its attainable maximum. Defaults to `FALSE`.

Details

Univariate version: test for serial dependence `Srho.test(x, lag.max, B = 1000, stationary = TRUE, plot = TRUE, quant = c(0.95, 0.99), nor = FALSE)`

Bivariate version: test for cross dependence `Srho.test(x, y, lag.max, B = 1000, stationary = TRUE, plot = TRUE, quant = c(0.95, 0.99), nor = FALSE)`

Value

An object of class "Srho.test", which is a list with the following elements:

`.Data`	vector of `lag.max` elements containing Srho computed at each lag.
`quantiles`	Object of class `"matrix"`: contains the quantiles of the bootstrap/permutation distribution under the null hypothesis.
`test.type`	Object of class `"character"`: contains a description of the type of test performed.

`significant.lags`	Object of class `"list"`: contains the lags at which Srho exceeds the confidence bands at `quant`% under the null hypothesis.
`p.value`	Object of class `"numeric"`: contains the bootstrap p-value for each lag.
`lags`	integer vector that contains the lags at which Srho is computed.
`stationary`	Object of class `"logical"`: `TRUE` if the stationary version is computed.
`data.type`	Object of class `"character"`: contains the data type.
`notes`	Object of class `"character"`: additional notes.

Warning

Unlike `ccf` the lag k value returned by `Srho.test(x,y)` estimates Srho between `x[t]` and `y[t+k]`. The result is returned invisibly if plot is TRUE.

Author(s)

Simone Giannerini<simone.giannerini@unibo.it>

References

Granger C. W. J., Maasoumi E., Racine J., (2004) A dependence metric for possibly nonlinear processes. *Journal of Time Series Analysis*, **25(5)**, 649–669.

Maasoumi E., (1993) A compendium to information theory in economics and econometrics. *Econometric Reviews*, **12(2)**, 137–181.

See Also

See also `Srho`, `Srho.ts`. The function `Srho.test.ts` implements the same test for numeric data.

Examples

```
set.seed(12)
x <- as.integer(rbinom(n=30,size=4,prob=0.5))
y <- as.integer(rbinom(n=30,size=4,prob=0.5))
z <- as.integer(c(4,abs(x[-30]*2-2))-rbinom(n=30,size=1,prob=1/2))

# no dependence
Srho.test(x,lag.max=4)   # univariate
Srho.test(x,y,lag.max=4) # bivariate

# lag 1 dependence
Srho.test(x,z,lag.max=4) # bivariate
```

`Srho.test-class` *Class "Srho.test"*

Description

A class of tests for serial dependence and nonlinearity based upon `Srho`.

Objects from the Class

Objects can be created by calls of the form `new("Srho.test", ...)`.

Slots

`.Data`**:** Object of class `"numeric"`: contains Srho computed on the data set.

`call`**:** Object of class `"call"`: contains the call to the routine.

`call.h`**:** Object of class `"call"`: contains the call to the routine used for obtaining the surrogates or the bootstrap replicates under the null hypothesis.

`quantiles`**:** Object of class `"matrix"`: contains the quantiles of the bootstrap/permutation distribution under the null hypothesis.

`test.type`**:** Object of class `"character"`: contains a description of the type of test performed.

`significant.lags`**:** Object of class `"list"`: contains the lags at which Srho exceeds the confidence bands at `quant` under the null hypothesis.

`p.value`**:** Object of class `"numeric"`: contains the bootstrap p-value for each lag.

`lags`**:** Object of class `"integer"`: contains the lags at which Srho is computed.

`stationary`**:** `TRUE` if the stationary version is computed.

`data.type`**:** Object of class `"character"`: contains the data type.

`notes`**:** Object of class `"character"`: additional notes.

Extends

Class `"Srho"`, directly.

Methods

plot `signature(x = "Srho.test", y = "missing")`: ...

show `signature(object = "Srho.test")`: ...

Author(s)

Simone Giannerini <simone.giannerini@unibo.it>

See Also

See Also `Srho`

Examples

```
showClass("Srho.test")
```

Srho.test.AR.p *Entropy Tests For Nonlinearity In Time Series - Parallel Version*

Description

Entropy test of nonlinearity for time series based on Srho.ts and surrogate data obtained through the sieve bootstrap. The parallel version requires parallel.

Usage

```
Srho.test.AR(x, y, lag.max = 10, B = 100, plot = TRUE, quant = c(0.95, 0.99),
 bw = c("reference", "mlcv", "lscv", "scv", "pi"), bdiag=TRUE,
 method = c("integral", "summation"), tol = 0.001, order.max = NULL,
 fit.method=c("yule-walker", "burg", "ols", "mle", "yw"),  smoothed = TRUE ,...)

## Parallel version

Srho.test.AR.p(x, y, lag.max = 10, B = 100, plot = TRUE, quant = c(0.95, 0.99),
 bw = c("reference", "mlcv", "lscv", "scv", "pi"), bdiag=TRUE,
 method = c("integral", "summation"), tol = 0.001, order.max = NULL,
 fit.method=c("yule-walker", "burg", "ols", "mle", "yw"),  smoothed = TRUE,
 nwork=detectCores(),...)
```

Arguments

x, y	univariate numeric time series object or numeric vectors (y is missing in the univariate case).
lag.max	maximum lag at which to calculate Srho; the default is 10.
B	number of surrogate time series.
plot	logical. If TRUE (the default) produces a plot of Srho together with confidence bands under the null hypothesis of linearity at 95% and 99%.
quant	quantiles to be specified for the computation of the significant lags and the plot of confidence bands. Up to 2 quantiles can be specified. Defaults are 95% and 99%.
bw	see Srho.ts.
bdiag	see Srho.ts.
method	see Srho.ts.
tol	see Srho.ts.
order.max	see surrogate.ARs.
fit.method	see surrogate.ARs.

`smoothed`	logical. If `TRUE` (the default) uses the smoothed sieve bootstrap in `surrogate.ARs` to generate surrogates. Otherwise uses the classic sieve by calling `surrogate.AR`.
`nwork`	number of workers/processes to be used in parallel environments.
`...`	further arguments, typically passed to `hcubature`.

Details

For each lag from 1 to `lag.max` `Srho.test.AR` computes a test for nonlinearity for time series based on `Srho.ts`. The distribution under the null hypothesis of linearity is obtained through the sieve bootstrap. The routine requires the package parallel to spawn multiple workers.

Value

An object of class "Srho.test", which is a list with the following elements:

`.Data`	vector of `lag.max` elements containing Srho computed at each lag.
`call:`	Object of class `"call"`: contains the call to the routine.
`call.h:`	Object of class `"call"`: contains the call to the routine used for obtaining the surrogates or the bootstrap replicates under the null hypothesis
`quantiles`	Object of class `"matrix"`: contains the quantiles of the surrogate distribution under the null hypothesis.
`test.type`	Object of class `"character"`: contains a description of the type of test performed.
`significant.lags`	Object of class `"list"`: contains the lags at which Srho exceeds the confidence bands at `quant`% under the null hypothesis.
`p.value`	Object of class `"numeric"`: contains the bootstrap p-value for each lag.
`lags`	integer vector that contains the lags at which Srho is computed.
`stationary`	Object of class `"logical"`: `TRUE` if the stationary version is computed. Set to `FALSE` by default as only the non-stationary version is implemented.
`data.type`	Object of class `"character"`: contains the data type.
`notes`	Object of class `"character"`: additional notes.

Author(s)

Simone Giannerini<simone.giannerini@unibo.it>

References

Giannerini S., Maasoumi E., Bee Dagum E., (2015), Entropy testing for nonlinear serial dependence in time series, *Biometrika*, **102(3)**, 661–675 doi:10.1093/biomet/asv007.

See Also

See Also `Srho.ts`, `surrogate.AR`, `surrogate.ARs`, `Srho.test.AR`.

Examples

```
## Not run:
## ************************************************************
## WARNING: computationally intensive, increase B with caution
## ************************************************************

# modify nwork to match the number of available cores
set.seed(13)
x      <- arima.sim(n=120, model = list(ar=0.8));
result <- Srho.test.AR.p(x, lag.max = 5,  B = 100, bw='reference', method='integral', nwork=2)

## ** Compare timings **
system.time(Srho.test.AR.p(x, lag.max = 5,  B = 100, bw='reference', method='integral', nwork=4))
system.time(Srho.test.AR(x, lag.max = 5,  B = 100, bw='reference', method='integral'))

## End(Not run)
```

`Srho.test.ts.p` *Entropy Tests Of Serial And Cross Dependence For Time Series*

Description

Entropy test of serial and cross dependence for numeric time series (continuous state space) based on `Srho.ts`. The distribution under the null hypothesis of independence is obtained by means of bootstrap/permutations methods (see `ci.type`). The parallel version requires `parallel`.

Usage

```
Srho.test.ts(x, y, lag.max = 10,  B = 100, plot = TRUE, quant = c(0.95, 0.99),
 bw = c("reference","mlcv", "lscv", "scv", "pi"), bdiag=TRUE,
 method =c("integral","summation"), tol=1e-03, ci.type = c("mbb","perm"),...)

## Parallel version
Srho.test.ts.p(x, y, lag.max = 10,  B = 100, plot = TRUE, quant = c(0.95, 0.99),
 bw = c("reference","mlcv", "lscv", "scv", "pi"), bdiag=TRUE,
 method =c("integral","summation"), tol=1e-03, ci.type = c("mbb","perm"),
 nwork=detectCores(),...)
```

Arguments

`x, y`	univariate numeric time series object or numeric vectors (`y` is missing in the univariate case).
`lag.max`	maximum lag at which to calculate Srho; the default is 10.
`B`	number of bootstrap/permutation replications.
`plot`	logical. If `TRUE`(the default) produces a plot of Srho together with confidence bands under the null hypothesis at levels set by `quant`.

`quant`	quantiles to be specified for the computation of the significant lags and the plot of confidence bands. Up to 2 quantiles can be specified. Defaults are 95% and 99%.
`bw`	see `Srho.ts`.
`bdiag`	see `Srho.ts`.
`method`	see `Srho.ts`.
`tol`	see `Srho.ts`.
`ci.type`	confidence interval type. determines how the distribution under the null hypothesis is obtained. `mbb` uses a moving block bootstrap with block length equal to `blag`, which is equal to `lag.max` by default. The option `perm` uses permutation methods (each resampled series is a random permutation of the original series). The option `mbb` makes sense only in the bivariate case for which is the default.
`nwork`	number of workers/processes to be used in parallel environments.
`...`	further arguments, typically, the MBB block length `blag` or the arguments passed to `hcubature`.

Details

Univariate version: test for serial dependence `Srho.test.ts.p(x, lag.max = 10,`

```
B = 100, plot = TRUE, quant = c(0.95, 0.99), bdiag=TRUE,
bw = c("reference", "mlcv", "lscv", "scv", "pi"), method =c("integral","summation"),
tol=1e-03, ci.type = c("perm"), nwork=detectCores())
```

Bivariate version: test for cross dependence `Srho.test.ts.p(x, y, lag.max = 10,`

```
B = 100, plot = TRUE, quant = c(0.95, 0.99), bdiag=TRUE,
bw = c("reference", "mlcv", "lscv", "scv", "pi"), method =c("integral","summation"),
tol=1e-03, ci.type = c("mbb","perm"), nwork=detectCores())
```

For each lag from 1 to `lag.max` (serial dependence) or from `-lag.max` to `lag.max` (cross dependence) `Srho.test.ts` computes a test for serial/cross dependence for time series based on `Srho.ts`. The distribution under the null hypothesis of independence is obtained through either permutation or bootstrap methods. If the option `mbb` is chosen (bivariate case only) the resampled series use a moving block bootstrap to acccount for the serial dependence of the original series so that the test will have better size than the permutation version.

Value

An object of class "Srho.test", which is a list with the following elements:

`.Data`	vector containing Srho computed at each lag.
`call:`	Object of class `"call"`: contains the call to the routine.
`call.h:`	Object of class `"call"`: contains the call to the routine used for obtaining the surrogates or the bootstrap replicates under the null hypothesis.
`quantiles`	Object of class `"matrix"`: contains the quantiles of the distribution under the null hypothesis.
`test.type`	Object of class `"character"`: contains a description of the type of test performed.

`significant.lags`	Object of class `"list"`: contains the lags at which Srho exceeds the confidence bands at `quant`% under the null hypothesis.
`p.value`	Object of class `"numeric"`: contains the bootstrap p-value for each lag.
`lags`	integer vector that contains the lags at which Srho is computed.
`stationary`	Object of class `"logical"`: `TRUE` if the stationary version is computed. Set to `FALSE` by default as only the non-stationary version is implemented.
`data.type`	Object of class `"character"`: contains the data type.
`notes`	Object of class `"character"`: additional notes.

Author(s)

Simone Giannerini<simone.giannerini@unibo.it>

References

Granger C. W. J., Maasoumi E., Racine J., (2004) A dependence metric for possibly nonlinear processes. *Journal of Time Series Analysis*, **25(5)**, 649–669.

Maasoumi E., (1993) A compendium to information theory in economics and econometrics. *Econometric Reviews*, **12(2)**, 137–181.

See Also

See Also `Srho.test.ts` and `Srho.ts`. The function `Srho.test` implements the same test for integer/categorical data. For a test for nonlinear serial dependence see `Srho.test.AR`, `Trho.test.AR`, `Trho.test.SA`, together with their parallel versions: `Srho.test.AR.p`, `Trho.test.AR`, `Trho.test.SA`.

Examples

```
## Not run:
## ************************************************************
## WARNING: computationally intensive, increase B with caution
## ************************************************************
set.seed(13)
n      <- 120
w      <- rnorm(n)
x      <- arima.sim(n, model = list(ar=0.8));
y      <- arima.sim(n, model = list(ar=0.8));
z      <- lag(x,-1) + rnorm(n,sd=2) # dependence at lag 1

# UNIVARIATE VERSION
res1 <- Srho.test.ts.p(w, lag.max = 5,  B = 40, ci.type="perm") # independence
res2 <- Srho.test.ts.p(x, lag.max = 5,  B = 40, ci.type="perm") # dependence

# BIVARIATE VERSION
res3 <- Srho.test.ts.p(x, y, lag.max = 5,  B = 40, ci.type="mbb") # independence
res4 <- Srho.test.ts.p(x, z, lag.max = 5,  B = 40, ci.type="mbb") # dependence

## End(Not run)
```

Srho.ts — *Entropy Measure Of Serial And Cross Dependence*

Description

Entropy based measure of serial and cross dependence for continuous data. For integer/categorical data see `Srho`. Implements a normalized version of the Hellinger/Matusita distance. As shown in the references the metric measure is a proper distance.

Usage

```
Srho.ts(x, y, lag.max = 10, bw = c("reference", "mlcv", "lscv", "scv", "pi"),
bdiag=TRUE, method = c("integral", "summation"), plot = TRUE, tol = 0.001, ...)
```

Arguments

`x, y`	univariate numeric time series object or numeric vectors (`y` is missing in the univariate case).
`lag.max`	maximum lag at which to calculate Srho; default is 10
`bw`	Object of class `"character"`: bandwidth selection method, can be `"reference"`, `"mlcv"`, `"lscv"`, `"scv"`, `"pi"`.
`bdiag`	Object of class `"logical"`: if `TRUE` uses the diagonal version of the bandwidth selectors `lscv`, `scv`, `pi`.
`method`	Object of class `"character"`: computation method, can be `"integral"` or `"summation"`.
`plot`	logical. If `TRUE` (the default) Srho is plotted.
`tol`	max. tolerance, passed to `hcubature`.
`...`	further arguments, typically passed to `hcubature`.

Details

Univariate version: serial entropy `Srho.ts(x, lag.max = 10,`
```
    bw = c("reference", "mlcv", "lscv", "scv", "pi"), bdiag=TRUE,
    method = c("integral", "summation"), plot = TRUE, tol = 0.001)
```
Bivariate version: cross entropy `Srho.ts(x, y, lag.max = 10,`
```
    bw = c("reference", "mlcv", "lscv", "scv", "pi"), bdiag=TRUE,
    method = c("integral", "summation"), plot = TRUE, tol = 0.001)
```

The bandwidth selection methods are the following:

`reference`: reference criterion.

`mlcv`: maximum likelihood cross-validation.

`lscv`: least-squares cross-validation, see `Hlscv`.

`scv`: smoothed cross-validation, see `Hscv`

`pi`: plugin, see `Hpi`

If `bdiag` = `TRUE` (the default), the diagonal bandwidth selectors `Hlscv.diag`, `Hscv.diag`, `Hpi.diag` are used.

Value

An object of class "Srho.ts", with the following slots:

`.Data`	Object of class `"numeric"`: contains Srho computed on the data set.
`method`	Object of class `"character"`: computation method
`bandwidth`	Object of class `"character"`: bandwidth selection method.
`lags`	Object of class `"integer"`: contains the lags at which Srho is computed.
`stationary`	Object of class `"logical"`: `TRUE` if the stationary version is computed.
`data.type`	Object of class `"character"`: contains the data type.
`notes`	Object of class `"character"`: additional notes.

Author(s)

Simone Giannerini<simone.giannerini@unibo.it>

References

Granger C. W. J., Maasoumi E., Racine J., (2004) A dependence metric for possibly nonlinear processes. *Journal of Time Series Analysis*, **25(5)**, 649–669.

Maasoumi E., (1993) A compendium to information theory in economics and econometrics. *Econometric Reviews*, **12(2)**, 137–181.

Giannerini S., Maasoumi E., Bee Dagum E., (2015), Entropy testing for nonlinear serial dependence in time series, *Biometrika*, **102(3)**, 661–675 doi:10.1093/biomet/asv007.

See Also

`Srho.test.ts`, `hcubature`, `ks`. The function `Srho` implements the same measure for integer/categorical data.

Examples

```
set.seed(11)
x <- arima.sim(list(order = c(1,0,0), ar = 0.8), n = 50)
S <- Srho.ts(x,lag.max=5,method="integral",bw="mlcv")

# creates a nonlinear dependence at lag 1
y <- c(runif(1),x[-50]^2*0.8-0.3)
S <- Srho.ts(x,y,lag.max=3,method="integral",bw="mlcv")
```

`Srho.ts-class` *Class "Srho.ts"*

Description

A class for `Srho` for continuous data/time series.

Objects from the Class

Objects can be created by calls of the form `new("Srho.ts", ...)`.

Slots

`.Data`**:** Object of class `"numeric"`: contains Srho computed on the data set.

`method`**:** Object of class `"character"`: computation method, can be "integral" or "summation".

`bandwidth`**:** Object of class `"character"`: bandwidth selection method.

`lags`**:** Object of class `"integer"`: contains the lags at which Srho is computed.

`stationary`**:** Object of class `"logical"`: `TRUE` if the stationary version is computed.

`data.type`**:** Object of class `"character"`: contains the data type.

`notes`**:** Object of class `"character"`: additional notes.

Extends

Class `"Srho"`, directly.

Methods

show `signature(object = "Srho.ts")`: ...

Author(s)

Simone Giannerini<simone.giannerini@unibo.it>

See Also

See Also `Srho.test`, `Srho`

Examples

```
showClass("Srho.ts")
```

surrogate.AR *Surrogate Time Series Through AR Modeling (Sieve Bootstrap)*

Description

Starting from a time series x given as input, the function generates surrogate series by means of the sieve bootstrap. The surrogates can be used for testing for non linearity in time series.

Usage

```
surrogate.AR(x, order.max = NULL, fit.method = c("yule-walker",
 "burg", "ols", "mle", "yw"), nsurr)
```

Arguments

`x`	a univariate numeric time series object or a numeric vector.
`order.max`	maximum order of the AR model to fit. Passed to `ar`
.	
`fit.method`	character string giving the method used to fit the AR model. It is passed to `ar` and has to be one of the strings in the default argument (partial matching works). Defaults to "yule-walker".
`nsurr`	number of surrogates.

Details

Let `N` be the length of the series `x`. The best AR model is chosen by means of the AIC criterion. The residuals of the model are resampled with replacement. Surrogate series are obtained by driving the fitted model with the resampled residuals.

Value

A list with the following elements:

`surr`	a matrix with `N` rows and `nsurr` columns, in each column is stored a surrogate.
`call`	contains the call to the routine.

Author(s)

Simone Giannerini<simone.giannerini@unibo.it>

References

Giannerini S., Maasoumi E., Bee Dagum E., (2015), Entropy testing for nonlinear serial dependence in time series, *Biometrika*, **102(3)**, 661–675 doi:10.1093/biomet/asv007.

Buhlmann, P., (1997). Sieve bootstrap for time series. *Bernoulli*, **3**, 123–148.

See Also

See also surrogate.AR, Trho.test.AR, surrogate.SA, Trho.test.SA.

Examples

```
set.seed(1345)
# Generates a AR(1) series
x      <- arima.sim(n=120, model = list(ar=0.8));
x.surr <- surrogate.AR(x, nsurr=3);
plot.ts(x.surr$surr,col=4);


## Check that the surrogates have the same ACF of x
corig <- acf(x,10,plot=FALSE)$acf[,,1];
csurr <- acf(x.surr$surr[,1],10,plot=FALSE)$acf[,,1];
round(cbind(corig,csurr,"abs(difference)"=abs(corig-csurr)),3)
```

---

surrogate.ARs — *Surrogate Time Series Through A Modeling (Smoothed Sieve Bootstrap)*

Description

Starting from a time series x given as input, the function generates surrogate series by means of the smoothed sieve bootstrap. The surrogates can be used for testing for non linearity in time series.

Usage

```
surrogate.ARs(x, order.max = NULL,
 fit.method = c("yule-walker","burg", "ols", "mle", "yw"), nsurr)
```

Arguments

x	a univariate numeric time series object or a numeric vector.
order.max	maximum order of the AR model to fit. Passed to ar.
fit.method	character string giving the method used to fit the AR model. It is passed to ar and has to be one of the strings in the default argument (partial matching works). Defaults to "yule-walker".
nsurr	number of surrogates.

Details

Let N be the length of the series x. The best AR model is chosen by means of the AIC criterion. Surrogate series are obtained by driving the fitted model with the smoothed resampled residuals. Smoothing is performed through Kernel density estimation with a Gaussian Kernel by using the dafaults of density.

Value

A list with the following elements:

`surr`	a matrix with `N` rows and `nsurr` columns, in each column is stored a surrogate.
`call`	contains the call to the routine.

Author(s)

Simone Giannerini<simone.giannerini@unibo.it>

References

Giannerini S., Maasoumi E., Bee Dagum E., (2015), Entropy testing for nonlinear serial dependence in time series, *Biometrika*, **102(3)**, 661–675 doi:10.1093/biomet/asv007.

Bickel, P., Buhlmann, P., (1999). A new mixing notion and functional central limit theorems for a sieve bootstrap in time series. *Bernoulli* **5**, 413–446.

See Also

See also `surrogate.AR`, `Trho.test.AR`, `surrogate.SA`, `Trho.test.SA`.

Examples

```
set.seed(1345)
# Generates a AR(1) series
x      <- arima.sim(n=120, model = list(ar=0.8));
x.surr <- surrogate.ARs(x, order.max=NULL, nsurr=3);
plot.ts(x.surr$surr,col=4);


## Check that the surrogates have the same ACF of x
corig <- acf(x,10,plot=FALSE)$acf[,,1];
csurr <- acf(x.surr$surr[,1],10,plot=FALSE)$acf[,,1];
round(cbind(corig,csurr,"abs(difference)"=abs(corig-csurr)),3)
```

`surrogate.SA` *Surrogate Time Series Through Simulated Annealing*

Description

Starting from a time series `x` given as input, the function generates surrogate series through Simulated Annealing. Each surrogate series is a constrained random permutation having the same autocorrelation function (up to `nlag` lags) of the original series `x`. The surrogates can be used for testing for non linearity in time series.

Usage

```
surrogate.SA(x, nlag, nsurr, Te = 0.0015, RT = 0.9, eps.SA = 0.05, nsuccmax = 30,
 nmax = 300, che = 1e+05)
```

Arguments

`x`	a univariate numeric time series object or a numeric vector.
`nlag`	minimization is performed w.r.t. to the first `nlag` lags.
`nsurr`	number of surrogates.
`Te`	starting value for the temperature.
`RT`	reduction factor for the temperature `Te`.
`eps.SA`	target tolerance.
`nsuccmax`	Te is decreased after `nsuccmax*N` successes.
`nmax`	Te is decreased after `nmax*N` successes.
`che`	after `che*2N` global iterations the algorithm starts again.

Details

Let `N` be the length of the series `x`. Sensible (`N`-dependent) defaults are derived for the parameters of the algorithm, there should not be the need to change them. In case, the user could try increasing `eps.SA`.

Value

A list with the following elements:

`surr`	a matrix with `N` rows and `nsurr` columns, in each column is stored a surrogate.
`call`	contains the call to the routine.

Author(s)

Simone Giannerini<simone.giannerini@unibo.it>

References

Giannerini S., Maasoumi E., Bee Dagum E., (2015), Entropy testing for nonlinear serial dependence in time series, *Biometrika*, **102(3)**, 661–675 doi:10.1093/biomet/asv007.

Schreiber T., Schmitz A.,(2000) Surrogate time series. *Physica D*, **142(3-4)**, 346–382.

See Also

See Also `Trho.test.SA`, `surrogate.AR`, `Trho.test.AR`.

Examples

```
set.seed(1345)
# Generates a AR(1) series
x      <- arima.sim(n=120, model = list(ar=0.8));
x.surr <- surrogate.SA(x, nlag=10, nsurr=3);
plot.ts(x.surr$surr,col=4);
```

```
## Check that the surrogates have the same ACF of x
corig <- acf(x,10,plot=FALSE)$acf[,,1];
csurr <- acf(x.surr$surr[,1],10,plot=FALSE)$acf[,,1];
round(cbind(corig,csurr,"abs(difference)"=abs(corig-csurr)),3)
```

`Trho.test.AR.p` *Entropy Tests For Nonlinearity In Time Series - Parallel Version*

Description

Entropy test of nonlinearity for time series based on `Srho.ts` and surrogate data obtained through the sieve bootstrap (AR modeling). The parallel version requires `parallel`.

Usage

```
Trho.test.AR(x, y, lag.max = 10, B = 100, plot = TRUE, quant = c(0.95, 0.99),
 bw = c("reference", "mlcv", "lscv", "scv", "pi"), bdiag=TRUE,
 method = c("integral", "summation"), tol = 0.001, order.max = NULL,
 fit.method=c("yule-walker", "burg", "ols", "mle", "yw"), smoothed = TRUE,...)

## Parallel version

Trho.test.AR.p(x, y, lag.max = 10, B = 100, plot = TRUE, quant = c(0.95, 0.99),
 bw = c("reference", "mlcv", "lscv", "scv", "pi"), bdiag=TRUE,
 method = c("integral", "summation"), tol = 0.001, order.max = NULL,
 fit.method=c("yule-walker", "burg", "ols", "mle", "yw"),  smoothed = TRUE,
 nwork=detectCores(),...)
```

Arguments

`x, y`	univariate numeric time series object or numeric vectors (`y` is missing in the univariate case).
`lag.max`	maximum lag at which to calculate Trho; the default is 10.
`B`	number of surrogate time series.
`plot`	logical. If `TRUE` (the default) produces a plot of Trho together with confidence bands under the null hypothesis of linearity at 95% and 99%.
`quant`	quantiles to be specified for the computation of the significant lags and the plot of confidence bands. Up to 2 quantiles can be specified. Defaults are 95% and 99%.
`bw`	see `Srho.ts`.
`bdiag`	see `Srho.ts`.
`method`	see `Srho.ts`.
`tol`	see `Srho.ts`.
`order.max`	see `surrogate.ARs`.

`fit.method`	see `surrogate.ARs`.
`smoothed`	logical. If `TRUE` (the default) uses the smoothed sieve bootstrap in `surrogate.ARs` to generate surrogates. Otherwise uses the classic sieve by calling `surrogate.AR`.
`nwork`	number of workers/processes to be used in parallel environments.
`...`	further arguments, typically passed to `hcubature`.

Details

For each lag from 1 to `lag.max` `Trho.test.AR` computes a test for nonlinearity for time series based on `Srho.ts`. The distribution under the null hypothesis of a linear Gaussian process is obtained through the sieve bootstrap. The routine requires the package parallel to spawn multiple workers.

Value

An object of class "Srho.test", which is a list with the following elements:

`.Data`	vector of `lag.max` elements containing Trho computed at each lag.
`call:`	Object of class `"call"`: contains the call to the routine.
`call.h:`	Object of class `"call"`: contains the call to the routine used for obtaining the surrogates or the bootstrap replicates under the null hypothesis
`quantiles`	Object of class `"matrix"`: contains the quantiles of the surrogate distribution under the null hypothesis.
`test.type`	Object of class `"character"`: contains a description of the type of test performed.
`significant.lags`	Object of class `"list"`: contains the lags at which Trho exceeds the confidence bands at `quant`% under the null hypothesis.
`p.value`	Object of class `"numeric"`: contains the bootstrap p-value for each lag.
`lags`	integer vector that contains the lags at which Trho is computed.
`stationary`	Object of class `"logical"`: `TRUE` if the stationary version is computed. Set to `FALSE` by default as only the non-stationary version is implemented.
`data.type`	Object of class `"character"`: contains the data type.
`notes`	Object of class `"character"`: additional notes.

Author(s)

Simone Giannerini<simone.giannerini@unibo.it>

References

Giannerini S., Maasoumi E., Bee Dagum E., (2015), Entropy testing for nonlinear serial dependence in time series, *Biometrika*, **102(3)**, 661–675 doi:10.1093/biomet/asv007.

See Also

See Also `Srho.ts`, `surrogate.AR`, `surrogate.ARs`, `Trho.test.AR`.

Examples

```
## Not run:

# modify nwork to match the number of available cores
set.seed(13)
b      <- 100
x      <- arima.sim(n=120, model = list(ar=0.8));
result <- Trho.test.AR.p(x, lag.max = 5, B=b, nwork=2)

## ** Compare timings **
system.time(Trho.test.AR.p(x,lag.max = 5,B=b, nwork=4))
system.time(Trho.test.AR(x,  lag.max = 5,B=b))

## End(Not run)
```

`Trho.test.SA.p` *Entropy Tests For Nonlinearity In Time Series - Parallel Version*

Description

Entropy test of nonlinearity for time series based on `Srho.ts` and surrogate data obtained through Simulated Annealing. The parallel version requires `parallel`.

Usage

```
Trho.test.SA(x, y, lag.max = 10,  B = 100, plot = TRUE, quant = c(0.95, 0.99),
 bw = c("reference","mlcv", "lscv", "scv", "pi"), bdiag=TRUE,
 method =c("integral","summation"), tol=1e-03, nlag=trunc(length(x)/4),
 Te=0.0015, RT=0.9, eps.SA=0.05, nsuccmax=30, nmax=300, che=100000,...)

Trho.test.SA.p(x, y, lag.max = 10,  B = 100, plot = TRUE, quant = c(0.95, 0.99),
 bw = c("reference","mlcv", "lscv", "scv", "pi"), bdiag=TRUE,
 method =c("integral","summation"), tol=1e-03, nlag=trunc(length(x)/4), Te=0.0015,
RT=0.9, eps.SA=0.05, nsuccmax=30, nmax=300, che=100000, nwork=detectCores(),...)
```

Arguments

`x, y`	univariate numeric time series object or numeric vectors (y is missing in the univariate case).
`lag.max`	maximum lag at which to calculate Trho; the default is 10.
`B`	number of surrogate time series.
`plot`	logical. If `TRUE`(the default) produces a plot of Trho together with confidence bands under the null hypothesis of linearity at 95% and 99%.
`quant`	quantiles to be specified for the computation of the significant lags and the plot of confidence bands. Up to 2 quantiles can be specified. Defaults are 95% and 99%.

`bw`	see `Srho.ts`.
`bdiag`	see `Srho.ts`.
`method`	see `Srho.ts`.
`tol`	see `Srho.ts`.
`nlag`	see `surrogate.SA`.
`Te`	see `surrogate.SA`.
`RT`	see `surrogate.SA`.
`eps.SA`	see `surrogate.SA`.
`nsuccmax`	see `surrogate.SA`.
`nmax`	see `surrogate.SA`.
`che`	see `surrogate.SA`.
`nwork`	number of workers/processes to be used in parallel environments.
`...`	further arguments, typically passed to `hcubature`.

Details

For each lag from 1 to `lag.max` `Trho.test.SA` computes a test for nonlinearity for time series based on `Srho.ts`. The distribution under the null hypothesis of a linear Gaussian process is obtained through a generalization of surrogate data methods. Surrogate time series are obtained through Simulated Annealing (SA). Sensible (N-dependent) defaults are derived for the parameters of the SA algorithm, there should not be the need to change them. The routine requires the package parallel to spawn multiple workers.

Value

An object of class "Srho.test", which is a list with the following elements:

`.Data`	vector of `lag.max` elements containing Trho computed at each lag.
`call:`	Object of class `"call"`: contains the call to the routine.
`call.h:`	Object of class `"call"`: contains the call to the routine used for obtaining the surrogates or the bootstrap replicates under the null hypothesis.
`quantiles`	Object of class `"matrix"`: contains the quantiles of the surrogate distribution under the null hypothesis.
`test.type`	Object of class `"character"`: contains a description of the type of test performed.
`significant.lags`	Object of class `"list"`: contains the lags at which Trho exceeds the confidence bands at `quant`% under the null hypothesis.
`p.value`	Object of class `"numeric"`: contains the bootstrap p-value for each lag.
`lags`	integer vector that contains the lags at which Trho is computed.
`stationary`	Object of class `"logical"`: `TRUE` if the stationary version is computed. Set to `FALSE` by default as only the non-stationary version is implemented.
`data.type`	Object of class `"character"`: contains the data type.
`notes`	Object of class `"character"`: additional notes.

Author(s)

Simone Giannerini<simone.giannerini@unibo.it>

References

Giannerini S., Maasoumi E., Bee Dagum E., (2015), Entropy testing for nonlinear serial dependence in time series, *Biometrika*, **102(3)**, 661–675 doi:10.1093/biomet/asv007.

See Also

See Also `Srho.ts`, `surrogate.SA`, `Trho.test.SA`.

Examples

```
## Not run:
# modify nwork to match the number of available cores
set.seed(13)
x      <- arima.sim(n=120, model = list(ar=0.8));
result <- Trho.test.SA.p(x, lag.max = 5, B = 100, bw='reference', nwork=2)

## ** Compare timings **
system.time(Trho.test.SA.p(x, lag.max = 5, B = 100, bw='reference', nwork=4))
system.time(Trho.test.SA(x, lag.max = 5, B = 100, bw='reference'))

## End(Not run)
```

Index