Package ‘DChaos’

October 17, 2019

Type Package

Version 0.1-3

Date 2019-10-16

Title Chaotic Time Series Analysis

Author Julio E. Sandubete [aut, cre],
Lorenzo Escot [aut]

Maintainer Julio E. Sandubete <jsandube@ucm.es>

Imports xts, zoo, outliers, nnet, pracma, sandwich, NeuralNetTools, entropy

Description Provides several algorithms for the purpose of detecting chaotic signals inside univariate time series. We focus on methods derived from chaos theory which estimate the complexity of a dataset through exploring the structure of the attractor. We have taken into account the Lyapunov exponents as an ergodic measure. We have implemented the Jacobian method by a fit through neural networks in order to estimate both the largest and the spectrum of Lyapunov exponents. We have considered the full sample and three different methods of subsampling by blocks (non-overlapping, equally spaced and bootstrap) to estimate them. In addition, it is possible to make inference about them and know if the estimated Lyapunov exponents values are or not statistically significant. This library can be used with time series whose time-lapse is fixed or variable. That is, it considers time series whose observations are sampled at fixed or variable time intervals. For a review see David Ruelle and Floris Takens (1971) <doi:10.1007/BF01646553>, Ramazan Gencay and W. Davis Dechert (1992) <doi:10.1016/0167-2789(92)90210-E>, Jean-Pierre Eckmann and David Ruelle (1995) <doi:10.1103/RevModPhys.57.617>, Mototsugu Shintani and Oliver Linton (2004) <doi:10.1016/S0304-4076(03)00205-7>, Jeremy P. Huke and David S. Broomhead (2007) <doi:10.1088/0951-7715/20/9/011>.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-10-16 22:00:02 UTC

R topics documented:

`embedding` *Construction of embedding vectors using the method of delays*

Description

This function generates both the uniform and non-uniform embedding vectors set from an univariate time serie considering the argument set selected by the user.

Usage

```
embedding(x, m = 3, lag = 1, timelapse = c("FIXED", "VARIABLE"))
```

Arguments

`x`	a numeric vector, time serie, data frame or matrix depending on the method selected in `timelapse`.
`m`	a non-negative integer denoting the embedding dimension (Default 3).
`lag`	a non-negative integer denoting the reconstruction delay (Default 1).
`timelapse`	a character denoting if you consider that the observations are sampled at uniform time intervals `FIXED` or with a variable time-lapse between each observation `VARIABLE` (Default `FIXED`).

Details

If `FIXED` has been selected `x` must be a numeric vector or time serie. Otherwise `VARIABLE` has to be specified. In this case `x` must be a data frame or matrix with two columns. First, the date with the following format `YMD H:M:OS3` considering milliseconds e.g., 20190407 00:00:03.347. If you don't consider milliseconds you must put .000 after the seconds. It should be an object of class `Factor`. Second, the univariate time serie as a sequence of numerical values.

Value

A data frame with the uniform or non-uniform embedding vectors by columns from an univariate time serie considering the parameter set selected by the user.

Author(s)

Julio E. Sandubete, Lorenzo Escot

References

Ruelle, D., Takens, F. 1971 On the nature of turbulence. Communications in Mathematical Physics 20(3):167-192.

Packard, N.H., Crutchfield, J.P., Farmer, J.D., Shaw, R.S. 1980 Geometry from a time serie. Physical Review Letters 45:712-716.

Takens, F. 1981 Detecting strange attractors in turbulence. Springer Berlin Heidelberg.

Sauer, T., Yorke, J.A., Casdagli, M. 1991 Embedology. Journal of Statistical Physics 65(3):579-616.

Huke, J.P., Broomhead, D.S. 2007 Embedding theorems for non-uniformly sampled dynamical systems. Nonlinearity 20(9):205-244.

See Also

jacobi

Examples

```
## The first ten values corresponding to the uniform embedding
## vectors set for m=5 and lag=1 are showed by simulating
## time series from the logistic equation.
data<-logistic.ts(u.min=4,u.max=4,B=100,doplot=FALSE)
ts<-data$`Logistic 100`$time.serie
embed<-embedding(ts,m=5,lag=1,timelapse="FIXED")
show(head(embed, 10))
```

---

henon.ts *Simulation of time series from the Hénon map*

Description

This function simulates time series from the Hénon map considering the argument set selected by the user. The initial conditions of each time serie are two random numbers between -0.5 and 0.5. Some values for the parameters and initial conditions may lead to an unstable system that will tend to infinity.

Usage

```
henon.ts(a.min = 0.4, a.max = 1.4, b.min = 0.1, b.max = 0.3,
  sample = 1000, transient = 100, B = 100, doplot = TRUE)
```

Arguments

`a.min`	a non-negative integer denoting a lower bound for the parameter `a` (Default 0.4).
`a.max`	a non-negative integer denoting an upper bound for the parameter `a` (Default 1.4).
`b.min`	a non-negative integer denoting a lower bound for the parameter `b` (Default 0.1).
`b.max`	a non-negative integer denoting an upper bound for the parameter `b` (Default 0.3).
`sample`	a non-negative integer denoting the length of each time serie (Default 1000).
`transient`	a non-negative integer denoting the number of observations that will be discarded to ensure that the values of each time serie are in the attractor (Default 100).
`B`	a non-negative integer denoting the number of series that will be generated for different values of the parameters `a` and `b`. The number of simulated series must be at least 100 (Default 100).
`doplot`	a logical value denoting if you want to draw a plot `TRUE` or not `FALSE`. If it is `TRUE` shows six graphs: the evolution of the temporal trajectories for the whole period, the attractor and its projections on the Cartesian plane and the bifurcation diagram. All of them consider both the 'x-coordinate' and the 'y-coordinate' (Default 'TRUE').

Value

A list containing as many items as series we want to simulate `B`. Each of them has the following attributes: the value of the parameter `a`, the value of the parameter `b`, the value of the initial condition `xo`, the value of the initial condition `yo` and a time serie from the iterated Hénon map with two columns corresponding to 'x-coordinate' and 'y-coordinate'.

Author(s)

Julio E. Sandubete, Lorenzo Escot

References

Hénon, M. 1976 A two-dimensional mapping with a strange attractor. Communications in Mathematical Physics 50(1):69-77.

See Also

`logistic.ts`, `rossler.ts`, `lorenz.ts`

Examples

```
## Simulates 100 time series from the Hénon map for different
## values of the parameters a and b in which this system exhibits
## a chaotic behaviour:
ts<-henon.ts(a.min=0.7,a.max=1.4,b.min=0.1,b.max=0.3,B=100,doplot=TRUE)
```

`infmutua` *Estimation of the Average Mutual Information function*

Description

This function estimates the Average Mutual Information function considering the argument set selected by the user.

Usage

```
infmutua(x, partitions = ceiling(1.5 + log(length(x))/log(2)),
  lag.max = 20, doplot = TRUE)
```

Arguments

`x`	a numeric vector or time serie.
`partitions`	a non-negative integer denoting the number of grouping of the set's elements into non-empty subsets, in such a way that every element is included in exactly one subset.
`lag.max`	a non-negative integer denoting an upper bound for the reconstruction delay (Default 20).
`doplot`	a logical value denoting if you want to draw a plot `TRUE` or not `FALSE`.

Value

The optimum lag which corresponds with the first minimum of the Average Mutual Information function.

Author(s)

Julio E. Sandubete, Lorenzo Escot

References

Fraser, A.M., Swinney, H.L. 1986 Independent coordinates for strange attractors from mutual information. Physical Review A 33(2):1134.

Examples

```
## The first minimum of the average mutual information
## function is showed by simulating a time series from
## the logistic equation.
data<-logistic.ts(u.min=4,u.max=4,B=100,doplot=FALSE)
ts<-data$`Logistic 100`$time.serie
lag.opt<-infmutua(ts,lag.max=10)
show(lag.opt$MutualInf)
show(lag.opt$FirstMin)
```

`jacobi` *Application of Jacobian method by a fit through neural networks*

Description

This function estimates the partial derivatives of the jacobian by a fit through a single-hidden-layer neural network considering the argument set selected by the user.

Usage

```
jacobi(x, lag = 1, timelapse = "FIXED", M0 = 3, M1 = 10, H0 = 2,
  H1 = 10, I = 100, pre.white = TRUE, doplot = TRUE)
```

Arguments

`x`	a numeric vector, time serie, data frame or matrix depending on the method selected in `timelapse`.
`lag`	a non-negative integer denoting the reconstruction delay (Default 1).
`timelapse`	a character denoting if you consider that the observations are sampled at uniform time intervals `FIXED` or with a variable time-lapse between each observation `VARIABLE` (Default `FIXED`).
`M0`	a non-negative integer denoting a lower bound for the embedding dimension (Default 3).
`M1`	a non-negative integer denoting an upper bound for the embedding dimension (Default 10).
`H0`	a non-negative integer denoting a lower bound for the number of neurones in the hidden layer (Default 2).
`H1`	a non-negative integer denoting an upper bound for the number of neurones in the hidden layer (Default 10).
`I`	a non-negative integer denoting a number of neural networks iterations (Default 100).
`pre.white`	a character denoting if you want to use as points to evaluate the partial derivatives the delayed vectors filtered by the neural network `TRUE` or not `FALSE` (Default `TRUE`).
`doplot`	a logical value denoting if you want to draw a plot `TRUE` or not `FALSE`. If it is `TRUE` shows as many graphs as networks have been considered. Each of them represents the network structure by drawing the weights with positive values in black and the weights with negative values in grey. The thickness of the lines represents a greater or lesser value (Default `TRUE`).

Details

If `FIXED` has been selected `x` must be a numeric vector or time serie. Otherwise `VARIABLE` has to be specified. In this case `x` must be a data frame or matrix with two columns. First, the date with the following format `YMD H:M:OS3` considering milliseconds e.g., 20190407 00:00:03.347. If you don't consider milliseconds you must put .000 after the seconds. It should be an object of class `Factor`. Second, the univariate time serie as a sequence of numerical values.

Value

A list with several objects. The first output is a matrix called `Network.set`. It contains the networks that have the best fit for each embedding dimension `m`. That is, the neural networks that have the minimum bayesian information criterion (BIC) between all possible number of neurones in the hidden layer. Then, the partial derivatives of the jacobian are saved on a data frame for each neural network structure considered by keeping to `Jacobian.net`.

Author(s)

Julio E. Sandubete, Lorenzo Escot

References

Eckmann, J.P., Ruelle, D. 1985 Ergodic theory of chaos and strange attractors. Reviews of Modern Physics 57:617-656.

Eckmann, J.P., Kamphorst, S.O., Ruelle, D., Ciliberto, S. 1986 Liapunov exponents from time series. Physical Review A 34:971-979.

Hornik, K., Stinchcombe, M., White, H. 1989 Multilayer feedforward networks are universal approximators. Neural Networks 2(5):359-366.

Gencay, R., Dechert, W. 1992 An algorithm for the n lyapunov exponents of an n-dimensional unknown dynamical system. Physica D 59(1):142-157.

McCaffrey, D.F., Ellner, S., Gallant, A.R., Nychka, D.W. 1992 Estimating the lyapunov exponent of a chaotic system with nonparametric regression. Journal of the American Statistical Association 87(419):682-695.

Nychka, D., Ellner, S., Gallant, A.R., McCaffrey, D. 1992 Finding chaos in noisy systems. Journal of the Royal Statistical Society 54(2):399-426.

Kuan, C., Liu, T., Gencay, R. 2004 Netfile 4.01: Feedforward neural networks and Lyapunov exponents estimation. Ball State University.

See Also

`embedding`

Examples

```
## We show below an example considering time series from the
## logistic equation. The first objetc is a matrix called
## Network.set. It contains the networks that have the best
## fit for each embedding dimension (3<m<4).
```

```
data<-logistic.ts(u.min=4,u.max=4,B=100,doplot=FALSE)
ts<-data$`Logistic 100`$time.serie
jacob<-jacobi(ts,lag=1,timelapse="FIXED",M0=3,M1=4,
       H0=3,H1=7,I=10,pre.white=TRUE,doplot=FALSE)
show(jacob$Network.set)
## The partial derivatives of the jacobian are saved on a
## data frame for each neural network structure considered
## by keeping to Jacobian.net. The first ten jacobian values
## corresponding to the neural network for m=4 are showed.
show(head(jacob$Jacobian.net2, 10))
```

`logistic.ts` *Simulation of time series from the Logistic equation*

Description

This function simulates time series from the Logistic equation considering the argument set selected by the user. The initial condition of each time serie is a random number between 0 and 1.

Usage

```
logistic.ts(u.min = 1, u.max = 4, sample = 1000, transient = 100,
  B = 100, doplot = TRUE)
```

Arguments

u.min	a non-negative integer denoting a lower bound for the parameter u (Default 1).
u.max	a non-negative integer denoting an upper bound for the parameter u (Default 4).
sample	a non-negative integer denoting the length of each time serie (Default 1000).
transient	a non-negative integer denoting the number of observations that will be discarded to ensure that the values of each time serie are in the attractor (Default 100).
B	a non-negative integer denoting the number of series that will be generated for several u-parameter values. The number of simulated series must be at least 100 (Default 100).
doplot	a logical value denoting if you want to draw a plot TRUE or not FALSE. If it is TRUE shows four graphs: the evolution of the temporal trajectories for an initial period, the evolution of these for the whole period, the attractor and its projections on the Cartesian plane and the bifurcation diagram (Default TRUE).

Value

A list containing as many items as series we want to simulate B. Each of them has the following attributes: the value of the parameter u, the value of the initial condition xo and a time serie from the iterated Logistic equation.

Author(s)

Julio E. Sandubete, Lorenzo Escot

References

May, R.M. 1976 Simple mathematical models with very complicated dynamics. Nature (261):459-467.

See Also

henon.ts, rossler.ts, lorenz.ts

Examples

```
## Simulates 100 time series from the logistic equation for
## u-parameter values in which this system exhibits a chaotic
## behaviour:
ts<-logistic.ts(u.min=3.57,u.max=4,B=100,doplot=TRUE)
```

`lorenz.ts` *Simulation of time series from the Lorenz system*

Description

This function simulates time series from the Lorenz system considering the argument set selected by the user. Some values for the parameters and initial conditions may lead to an unstable system that will tend to infinity.

Usage

```
lorenz.ts(sigma.min = 8, sigma.max = 10, rho.min = 25,
  rho.max = 27, beta.min = 1, beta.max = 2.67, xo.min = -14,
  xo.max = -10, yo.min = -13, yo.max = -10, zo.min = 3,
  zo.max = 10, time = seq(0, 100, 0.01), transient = 1000, B = 100,
  doplot = TRUE)
```

Arguments

sigma.min	a non-negative integer denoting a lower bound for the parameter sigma (Default 8).
sigma.max	a non-negative integer denoting an upper bound for the parameter sigma (Default 10).
rho.min	a non-negative integer denoting a lower bound for the parameter rho (Default 25).
rho.max	a non-negative integer denoting an upper bound for the parameter rho (Default 27).

beta.min	a non-negative integer denoting a lower bound for the parameter beta (Default 1).
beta.max	a non-negative integer denoting an upper bound for the parameter beta (Default 2.67).
xo.min	a non-negative integer denoting a lower bound for the initial condition xo (Default -14).
xo.max	a non-negative integer denoting an upper bound for the initial condition xo (Default -10).
yo.min	a non-negative integer denoting a lower bound for the initial condition yo (Default -13).
yo.max	a non-negative integer denoting an upper bound for the initial condition yo (Default -10).
zo.min	a non-negative integer denoting a lower bound for the initial condition zo (Default 3).
zo.max	a non-negative integer denoting an upper bound for the initial condition zo (Default 10).
time	a numeric vector denoting the time-lapse and the time-step (Default 'time-lapse' equal to 10001 with a 'time-step' of 0.01 seconds).
transient	a non-negative integer denoting the number of observations that will be discarded to ensure that the values of each time serie are in the attractor (Default 1000).
B	a non-negative integer denoting the number of series that will be generated for different values of parameters sigma, rho and beta. The number of simulated series must be at least 100 (Default 100).
doplot	a logical value denoting if you want to draw a plot TRUE or not FALSE. If it is TRUE shows six graphs: the evolution of the temporal trajectories for the whole period, the attractor and its projections on the Cartesian plane. All of them consider the 'x-coordinate', the 'y-coordinate' and the 'z-coordinate' (Default TRUE).

Value

A list containing as many items as series we want to simulate B. Each of them has the following attributes: the value of the parameter sigma, the value of the parameter rho, the value of the parameter beta, the value of the initial condition xo, the value of the initial condition yo, the value of the initial condition zo and a time serie from the iterated Lorenz system with three columns corresponding to 'x-coordinate', 'y-coordinate' and 'z-coordinate'.

Author(s)

Julio E. Sandubete, Lorenzo Escot

References

Lorenz, E. 1963 Deterministic nonperiodic flow. Journal of the Atmospheric Sciences 20(2):130-141.

See Also

logistic.ts, henon.ts, rossler.ts

Examples

```
## Simulates 100 time series from the Lorenz system for different
## values of the parameters sigma, rho and beta in which this system
## exhibits a chaotic behaviour:
ts<-lorenz.ts(sigma.min=10,sigma.max=10,rho.min=27,rho.max=27,beta.min=2.67,
              beta.max=2.67,time=seq(0,10,0.01),transient=100,B=100, doplot=TRUE)
```

---

lyapunov — *Estimation of the Lyapunov exponent through several methods*

Description

This function estimates both the largest Lyapunov exponent through the Norma-2 method and the Lyapunov exponent spectrum through the QR decomposition method taking into account the full sample and three different methods of subsampling by blocks considering the argument set selected by the user.

Usage

```
lyapunov(x, lag = 1, timelapse = "FIXED", M0 = 3, M1 = 10,
  H0 = 2, H1 = 10, I = 100, lyapmethod = c("LLE", "SLE", "ALL"),
  blocking = c("FULL", "NOVER", "EQS", "BOOT", "ALL"),
  pre.white = TRUE, B = 100, netplot = TRUE, doplot = TRUE)
```

Arguments

`x`	a numeric vector, time serie, data frame or matrix depending on the method selected in `timelapse`.
`lag`	a non-negative integer denoting the reconstruction delay (Default 1).
`timelapse`	a character denoting if you consider that the observations are sampled at uniform time intervals `FIXED` or with a variable time-lapse between each observation `VARIABLE` (Default `FIXED`).
`M0`	a non-negative integer denoting a lower bound for the embedding dimension (Default 3).
`M1`	a non-negative integer denoting an upper bound for the embedding dimension (Default 10).
`H0`	a non-negative integer denoting a lower bound for the number of neurones in the hidden layer (Default 2).
`H1`	a non-negative integer denoting an upper bound for the number of neurones in the hidden layer (Default 10).

`I`	a non-negative integer denoting a number of neural networks iterations (Default 100).
`lyapmethod`	a character denoting if you want to estimate the largest Lyapunov exponent `LLE`, the Lyapunov exponent spectrum `SLE` or both `ALL` (Default `LLE`).
`blocking`	a character denoting if you consider the full sample `FULL`, the non-overlapping sample `NOVER`, the equally spaced sample `EQS`, the bootstrap sample `BOOT` or all of them `ALL` (Default `FULL`).
`pre.white`	a character denoting if you want to use as points to evaluate the partial derivatives the delayed vectors filtered by the neural network `TRUE` or not `FALSE` (Default `TRUE`).
`B`	a non-negative integer denoting the number of bootstrap iterations (Default 100).
`netplot`	a logical value denoting if you want to draw a plot `TRUE` or not `FALSE`. If it is `TRUE` shows as many graphs as networks have been considered. Each of them represents the network structure by drawing the weights with positive values in black and the weights with negative values in grey. The thickness of the lines represents a greater or lesser value (Default `TRUE`).
`doplot`	a logical value denoting if you want to draw a plot `TRUE` or not `FALSE`. If it is `TRUE` the evolution of the Lyapunov exponent values are represented for the whole period considering the lyapunov methods and the blocking methods selected by the user (Default `TRUE`).

Details

If `FIXED` has been selected `x` must be a numeric vector or time serie. Otherwise `VARIABLE` has to be specified. In this case `x` must be a data frame or matrix with two columns. First, the date with the following format `YMD H:M:OS3` considering milliseconds e.g., 20190407 00:00:03.347. If you don't consider milliseconds you must put .000 after the seconds. It should be an object of class `Factor`. Second, the univariate time serie as a sequence of numerical values.

Value

A list containing the largest Lyapunov exponent, the Lyapunov exponent spectrum or both for each neural network structure considered by keeping to `Lyapunov.net`. The dataset saved by each blocking method are the estimated Lyapunov exponent value, the standard error, the z-test value and the p-value. If the user chooses the non-overlapping sample, the equally spaced sample or the bootstrap sample the mean and median values of the Lyapunov exponent are showed. Also some details about the embedding dimension, the sample size, the block length and the block number are recorded.

Author(s)

Julio E. Sandubete, Lorenzo Escot

References

Ellner, S., Gallant, A., McCaffrey, D., Nychka, D. 1991 Convergence rates and data requirements for jacobian-based estimates of lyapunov exponents from data. Physics Letters A 153(6):357-363.

McCaffrey, D.F., Ellner, S., Gallant, A.R., Nychka, D.W. 1992 Estimating the lyapunov exponent of a chaotic system with nonparametric regression. Journal of the American Statistical Association 87(419):682-695.

Nychka, D., Ellner, S., Gallant, A.R., McCaffrey, D. 1992 Finding chaos in noisy systems. Journal of the Royal Statistical Society 54(2):399-426.

Whang, Y.J., Linton, O. 1999 The asymptotic distribution of nonparametric estimates of the lyapunov exponent for stochastic time series. Journal of Econometrics 91(1):1-42.

Shintani, M., Linton, O. 2004 Nonparametric neural network estimation of Lyapunov exponents and a direct test for chaos. Journal of Econometrics 120(1):1-33.

See Also

embedding, jacobi, lyapunov.max, lyapunov.spec

Examples

```
## We show below an example considering time series from the
## logistic equation. We have estimated the Lyapunov exponent
## spectrum by each blocking method for an embedding dimension (m=4).
data<-logistic.ts(u.min=4,u.max=4,B=100,doplot=FALSE)
ts<-data$`Logistic 100`$time.serie
lyapu<-lyapunov(ts,lag=1,timelapse="FIXED",M0=4,M1=4,H0=2,H1=7,I=10,
       lyapmethod="SLE",blocking="ALL",pre.white=TRUE,B=30,netplot=FALSE,
       doplot=FALSE)
show(lyapu$Lyapunov.net$Spectrum.full$Exponent)
show(lyapu$Lyapunov.net$Spectrum.nonoverlap$Exponent)
show(lyapu$Lyapunov.net$Spectrum.equally$Exponent)
show(lyapu$Lyapunov.net$Spectrum.bootstrap$Exponent)
```

---

lyapunov.max *Estimation of the largest Lyapunov Exponent*

Description

This function estimates the largest Lyapunov exponent through the Norma-2 method considering the argument set selected by the user.

Usage

```
lyapunov.max(x, blocking = c("FULL", "NOVER", "EQS", "BOOT"), B = 100,
  doplot = TRUE)
```

Arguments

`x`	a matrix or data frame containing the partial derivatives of jacobian.
`blocking`	a character denoting if you consider the full sample `FULL`, the non-overlapping sample `NOVER`, the equally spaced sample `EQS` or the bootstrap sample `BOOT` (Default `FULL`).

B	a non-negative integer denoting the number of bootstrap iterations (Default 100).
doplot	a logical value denoting if you want to draw a plot `TRUE` or not `FALSE`. If it is `TRUE` shows as many graphs as embedding dimensions have been considered. Each of them represents the evolution of the largest Lyapunov exponent values for the whole period considering the blocking method selected by the user (Default `TRUE`).

Value

A list containing the largest Lyapunov exponent considering the parameter set selected by the user. The dataset saved by each blocking method are the estimated Lyapunov exponent value, the standard error, the z-test value and the p-value. If the user chooses the non-overlapping sample, the equally spaced sample or the bootstrap sample the mean and median values of the Lyapunov exponent are showed. Also some details about the embedding dimension, the sample size, the block length and the block number are recorded.

Author(s)

Julio E. Sandubete, Lorenzo Escot

References

Ellner, S., Gallant, A., McCaffrey, D., Nychka, D. 1991 Convergence rates and data requirements for jacobian-based estimates of lyapunov exponents from data. Physics Letters A 153(6):357-363.

McCaffrey, D.F., Ellner, S., Gallant, A.R., Nychka, D.W. 1992 Estimating the lyapunov exponent of a chaotic system with nonparametric regression. Journal of the American Statistical Association 87(419):682-695.

Nychka, D., Ellner, S., Gallant, A.R., McCaffrey, D. 1992 Finding chaos in noisy systems. Journal of the Royal Statistical Society 54(2):399-426.

Whang, Y.J., Linton, O. 1999 The asymptotic distribution of nonparametric estimates of the lyapunov exponent for stochastic time series. Journal of Econometrics 91(1):1-42.

Shintani, M., Linton, O. 2004 Nonparametric neural network estimation of Lyapunov exponents and a direct test for chaos. Journal of Econometrics 120(1):1-33. ## We show below an example considering time series from the ## logistic equation. We have estimated the largest Lyapunov ## exponent considering the bootstrap sample for an embedding ## dimension (m=4). First of all, we need to estimates the ## partial derivatives of the jacobian. data<-logistic.ts(u.min=4,u.max=4,B=100,doplot=FALSE) ts<-data$‘Logistic 100‘$time.serie jacob<-jacobi(ts,M0=4,M1=4,doplot=FALSE) deriv<-jacob$Jacobian.net lyapu<-lyapunov.max(deriv,blocking="BOOT",B=100) show(lyapu$Exponent)

See Also

lyapunov.spec, lyapunov

`lyapunov.spec` *Estimation of the Lyapunov exponent spectrum*

Description

This function estimates the Lyapunov exponent spectrum through the QR decomposition method considering the argument set selected by the user.

Usage

```
lyapunov.spec(x, blocking = c("FULL", "NOVER", "EQS", "BOOT"), B = 100,
  doplot = TRUE)
```

Arguments

`x`	a matrix or data frame containing the partial derivatives of jacobian.
`blocking`	a character denoting if you consider the full sample `FULL`, the non-overlapping sample `NOVER`, the equally spaced sample `EQS` or the bootstrap sample `BOOT` (Default `FULL`).
`B`	a non-negative integer denoting the number of bootstrap iterations (Default 100).
`doplot`	a logical value denoting if you want to draw a plot `TRUE` or not `FALSE`. If it is `TRUE` shows as many graphs as embedding dimensions have been considered. Each of them represents the evolution of the Lyapunov exponent spectrum values for the whole period considering the blocking method selected by the user (Default `TRUE`).

Value

A list containing the Lyapunov exponent spectrum considering the parameter set selected by the user. The dataset saved by each blocking method are the estimated Lyapunov exponent value, the standard error, the z-test value and the p-value. If the user chooses the non-overlapping sample, the equally spaced sample or the bootstrap sample the mean and median values of the Lyapunov exponent are showed. Also some details about the embedding dimension, the sample size, the block length and the block number are recorded.

Author(s)

Julio E. Sandubete, Lorenzo Escot

References

Ellner, S., Gallant, A., McCaffrey, D., Nychka, D. 1991 Convergence rates and data requirements for jacobian-based estimates of lyapunov exponents from data. Physics Letters A 153(6):357-363.

McCaffrey, D.F., Ellner, S., Gallant, A.R., Nychka, D.W. 1992 Estimating the lyapunov exponent of a chaotic system with nonparametric regression. Journal of the American Statistical Association 87(419):682-695.

Nychka, D., Ellner, S., Gallant, A.R., McCaffrey, D. 1992 Finding chaos in noisy systems. Journal of the Royal Statistical Society 54(2):399-426.

Whang, Y.J., Linton, O. 1999 The asymptotic distribution of nonparametric estimates of the lyapunov exponent for stochastic time series. Journal of Econometrics 91(1):1-42.

Shintani, M., Linton, O. 2004 Nonparametric neural network estimation of Lyapunov exponents and a direct test for chaos. Journal of Econometrics 120(1):1-33.

See Also

`lyapunov.max`, `lyapunov`

Examples

```
## We show below an example considering time series from the
## logistic equation. We have estimated the Lyapunov exponent
## spectrum considering the bootstrap sample for an embedding
## dimension (m=4). First of all, we need to estimates the
## partial derivatives of the jacobian.
data<-logistic.ts(u.min=4,u.max=4,B=100,doplot=FALSE)
ts<-data$`Logistic 100`$time.serie
jacob<-jacobi(ts,M0=4,M1=4,doplot=FALSE)
deriv<-jacob$Jacobian.net
lyapu<-lyapunov.spec(deriv,blocking="BOOT",B=10)
show(lyapu$Exponent)
```

`rossler.ts` *Simulation of time series from the Rössler system*

Description

This function simulates time series from the Rössler system considering the argument set selected by the user. Some values for the parameters and initial conditions may lead to an unstable system that will tend to infinity.

Usage

```
rossler.ts(a.min = 0.1, a.max = 0.2, b.min = 0.1, b.max = 0.2,
  c.min = 4, c.max = 5.7, xo.min = -2, xo.max = 2, yo.min = -10,
  yo.max = 10, zo.min = -0.2, zo.max = 0.2, time = seq(0, 100,
  0.01), transient = 1000, B = 100, doplot = TRUE)
```

Arguments

`a.min`	a non-negative integer denoting a lower bound for the parameter `a` (Default 0).
`a.max`	a non-negative integer denoting an upper bound for the parameter `a` (Default 0.2).
`b.min`	a non-negative integer denoting a lower bound for the parameter `b` (Default 0).

b.max	a non-negative integer denoting an upper bound for the parameter b (Default 0.2).
c.min	a non-negative integer denoting a lower bound for the parameter c (Default 4).
c.max	a non-negative integer denoting an upper bound for the parameter c (Default 5.7).
xo.min	a non-negative integer denoting a lower bound for the initial condition xo (Default -2).
xo.max	a non-negative integer denoting an upper bound for the initial condition xo (Default 2).
yo.min	a non-negative integer denoting a lower bound for the initial condition yo (Default -10).
yo.max	a non-negative integer denoting an upper bound for the initial condition yo (Default 10).
zo.min	a non-negative integer denoting a lower bound for the initial condition zo (Default -0.2).
zo.max	a non-negative integer denoting an upper bound for the initial condition zo (Default 0.2).
time	a numeric vector denoting the time-lapse and the time-step (Default 'time-lapse' equal to 10001 with a 'time-step' of 0.01 seconds).
transient	a non-negative integer denoting the number of observations that will be discarded to ensure that the values of each time serie are in the attractor (Default 1000).
B	a non-negative integer denoting the number of series that will be generated for different values of parameters a, b and c. The number of simulated series must be at least 100 (Default 100).
doplot	a logical value denoting if you want to draw a plot TRUE or not FALSE. If it is TRUE shows six graphs: the evolution of the temporal trajectories for the whole period, the attractor and its projections on the Cartesian plane. All of them consider the 'x-coordinate', the 'y-coordinate' and the 'z-coordinate' (Default TRUE).

Value

A list containing as many items as series we want to simulate B. Each of them has the following attributes: the value of the parameter a, the value of the parameter b, the value of the parameter c, the value of the initial condition xo, the value of the initial condition yo, the value of the initial condition zo and a time serie from the iterated Rössler system with three columns corresponding to 'x-coordinate', 'y-coordinate' and 'z-coordinate'.

Author(s)

Julio E. Sandubete, Lorenzo Escot

References

Rössler, O. 1976 An equation for continuous chaos. Physics Letters A 57(5):397-398.

See Also

logistic.ts, henon.ts, lorenz.ts

Examples

```
## Simulates 100 time series from the Rössler system for different
## values of the parameters a, b and c in which this system exhibits
## a chaotic behaviour:
ts<-rossler.ts(a.min=0.2,a.max=0.2,b.min=0.2,b.max=0.2,c.min=5.7,c.max=5.7,
               time=seq(0,10,0.01),transient=100,B=100,doplot=TRUE)
```

Index