Package ‘VLTimeCausality’

December 20, 2019

Title Variable-Lag Time Series Causality Inference Framework

Version 0.1.0

Description A framework to infer causality on a pair of time series of real numbers based on variable-lag Granger causality and transfer entropy. Typically, Granger causality and transfer entropy have an assumption of a fixed and constant time delay between the cause and effect. However, for a non-stationary time series, this assumption is not true. For example, considering two time series of velocity of person A and person B where B follows A. At some time, B stops tying his shoes, then running to catch up A. The fixed-lag assumption is not true in this case. We propose a framework that allows variable-lags between cause and effect in Granger causality and transfer entropy to allow them to deal with variable-lag non-stationary time series. Please see Chainarong Amornbunchornvej, Elena Zheleva, and Tanya Berger-Wolf (2019) <https://www.cs.uic.edu/~elena/pubs/amornbunchornvej-dsaa19.pdf> when referring to this package in publications.

License GPL-3

URL https://github.com/DarkEyes/VLTimeSeriesCausality

BugReports https://github.com/DarkEyes/VLTimeSeriesCausality/issues

Language en-US

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0), dtw, tseries, RTransferEntropy

Imports ggplot2 (>= 3.0)

Suggests knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.0.0

NeedsCompilation no

Author Chainarong Amornbunchornvej [aut, cre] (<https://orcid.org/0000-0003-3131-0370>)

Maintainer Chainarong Amornbunchornvej <grandca@gmail.com>

Repository CRAN

Date/Publication 2019-12-20 13:50:08 UTC

R topics documented:

checkMultipleSimulationVLtimeseries 2
followingRelation 3
GrangerFunc 4
MultipleSimulationVLtimeseries 5
multipleVLGrangerFunc 6
multipleVLTransferEntropy 7
plotTimeSeries 8
SimpleSimulationVLtimeseries 9
VLGrangerFunc 10
VLTransferEntropy 11

Index **13**

`checkMultipleSimulationVLtimeseries`
checkMultipleSimulationVLtimeseries

Description

checkMultipleSimulationVLtimeseries is a support function that can compare two adjacency matrices: groundtruth and inferred matrices. It re

Usage

```
checkMultipleSimulationVLtimeseries(trueAdjMat, adjMat)
```

Arguments

`trueAdjMat`	a groundtruth matrix.
`adjMat`	an inferred matrix.

Value

This function returns a list of precision `prec`, recall `rec`, and F1 score `F1` of inferred vs. groundtruth matrices.

Examples

```
# Generate simulation data
G<-matrix(FALSE,10,10) # groundtruth
G[1,c(4,7,8,10)]<-TRUE
G[2,c(5,7,9,10)]<-TRUE
G[3,c(6,8,9,10)]<-TRUE
TS <- MultipleSimulationVLtimeseries()
out<-multipleVLGrangerFunc(TS)
checkMultipleSimulationVLtimeseries(trueAdjMat=G,adjMat=out$adjMat)
```

`followingRelation` *followingRelation*

Description

followingRelation is a function that infers whether `Y` follows `X`.

Usage

```
followingRelation(Y, X, timeLagWindow, lagWindow = 0.2)
```

Arguments

`Y`	is a numerical time series of a follower
`X`	is a numerical time series of a leader
`timeLagWindow`	is a maximum possible time delay in the term of time steps.
`lagWindow`	is a maximum possible time delay in the term of percentage of length(X). If `timeLagWindow` is missing, then `timeLagWindow=ceiling(lagWindow*length(X))`. The default is 0.2.

Value

This function returns a list of following relation variables below.

`follVal`	is a following-relation value s.t. if `follVal` is positive, then `Y` follows `X`. If `follVal` is negative, then `X` follows `Y`. Otherwise, if `follVal` is zero, there is no following relation between `X,Y`.
`nX`	is a time series that is rearranged from `X` by applying the lags `optIndexVec` in order to imitate `Y`.
`optDelay`	is the optimal time delay inferred by cross-correlation of `X,Y`. It is positive if `Y` is simply just a time-shift of `X` (e.g. `Y[t]=X[t-optDelay]`).
`optCor`	is the optimal correlation of `Y[t]=X[t-optDelay]` for all `t`.
`optIndexVec`	is a time series of optimal warping-path from DTW that is corrected by cross correlation. It is approximately that `Y[t]=X[t-optIndexVec[t]]`).
`VLval`	is a percentage of elements in `optIndexVec` that is not equal to `optDelay`.
`ccfout`	is an output object of `ccf` function.

Examples

```
# Generate simulation data
TS <- SimpleSimulationVLtimeseries()
# Run the function
out<-followingRelation(Y=TS$Y,X=TS$X)
```

`GrangerFunc` *GrangerFunc*

Description

GrangerFunc is a Granger Causality function. It tests whether `X` Granger-causes `Y`.

Usage

```
GrangerFunc(
  Y,
  X,
  maxLag = 1,
  alpha = 0.05,
  autoLagflag = TRUE,
  gamma = 0.5,
  family = gaussian
)
```

Arguments

`Y`	is a numerical time series of effect
`X`	is a numerical time series of cause
`maxLag`	is a maximum possible time delay. The default is 1.
`alpha`	is a significance level of F-test to determine whether `X` Granger-causes `Y`. The default is 0.05.
`autoLagflag`	is a flag for enabling the automatic lag inference function. The default is true. If it is set to be true, then maxLag is set automatically using cross-correlation. Otherwise, if it is set to be false, then the function takes the maxLag value to infer Granger causality.
`gamma`	is a parameter to determine whether `X` Granger-causes `Y` using BIC difference ratio.
`family`	is a parameter of family of function for Generalized Linear Models function (glm). The default is `gaussian`.

Value

This function returns of whether `X` Granger-causes `Y`.

`ftest`	F-statistic of Granger causality.
`p.val`	A p-value from F-test.
`BIC_H0`	Bayesian Information Criterion (BIC) derived from `Y` regressing on `Y` past.
`BIC_H1`	Bayesian Information Criterion (BIC) derived from `Y` regressing on `Y`,`X` past.
`XgCsY`	The flag is true if `X` Granger-causes `Y` using BIC difference ratio where `BICDiffRatio >= gamma`.

XgCsY_ftest	The flag is true if X Granger-causes Y using F-test where p.val>=alpha.
XgCsY_BIC	The flag is true if X Granger-causes Y using BIC where BIC_H0>=BIC_H1.
maxLag	A maximum possible time delay.
H0	glm object of Y regressing on Y past.
H1	glm object of Y regressing on Y,X past.
BICDiffRatio	Bayesian Information Criterion difference ratio: (BIC_H0-BIC_H1)/BIC_H0.

Examples

```
# Generate simulation data
TS <- SimpleSimulationVLtimeseries()
# Run the function
out<-GrangerFunc(Y=TS$Y,X=TS$X)
```

MultipleSimulationVLtimeseries

MultipleSimulationVLtimeseries

Description

MultipleSimulationVLtimeseries is a support function for generating a set of time series TS[,1],...TS[,10]. TS[,1],TS[,2],TS[,3] are causes X time series that are generated independently. The rest of time series are Y time series that are effects of some causes TS[,1],TS[,2],TS[,3]. TS[,1] causes TS[,4],TS[,7],TS[,8], and TS[,10]. TS[,2] causes TS[,5],TS[,7],TS[,9], and TS[,10]. TS[,3] causes TS[,6],TS[,8],TS[,9], and TS[,10].

Usage

```
MultipleSimulationVLtimeseries(
  n = 200,
  lag = 5,
  YstFixInx = 111,
  YfnFixInx = 150,
  XpointFixInx = 100,
  arimaFlag = TRUE
)
```

Arguments

n	is length of time series.
lag	is a time lag between X and Y s.t. Y[t] is approximately X[t-lag].
YstFixInx	is the starting point of variable lag part.
YfnFixInx	is the end point of variable lag part.
XpointFixInx	is a point in X s.t. Y[YstFixInx:YfnFixInx]= X[XpointFixInx] .
arimaFlag	is ARMA model flag. If it is true, then X is generated by ARMA model. If it is false, then X is generated by sampling of the standard normal distribution.

Value

This function returns a list of time series `TS`.

Examples

```
# Generate simulation data
TS <- MultipleSimulationVLtimeseries()
```

`multipleVLGrangerFunc` *multipleVLGrangerFunc*

Description

multipleVLGrangerFunc is a function that infers Variable-lag Granger Causality of all pairwises of `m` time series `TS[,1],...TS[,m]`.

Usage

```
multipleVLGrangerFunc(
  TS,
  maxLag,
  alpha = 0.05,
  gamma = 0.3,
  autoLagflag = TRUE,
  causalFlag = 0,
  VLflag = TRUE,
  family = gaussian
)
```

Arguments

`TS`	is a numerical time series of effect where `TS[t,k]` is an element at time `t` of `k`th time series.
`maxLag`	is a maximum possible time delay. The default is 0.2*length(Y).
`alpha`	is a significance level of F-test to determine whether `X` Granger-causes `Y`. The default is 0.05.
`gamma`	is a parameter to determine whether `X` Granger-causes `Y` using BIC difference ratio. The default is 0.3.
`autoLagflag`	is a flag for enabling the automatic lag inference function. The default is true. If it is set to be true, then maxLag is set automatically using cross-correlation. Otherwise, if it is set to be false, then the function takes the maxLag value to infer Granger causality.
`causalFlag`	is a choice of criterion for inferring causality: `causalFlag=0` for BIC difference ratio, `causalFlag=1` for f-test, or `causalFlag=2` for BIC.

`VLflag` is a flag of Granger causality choice: either `VLflag=TRUE` for VL-Granger or `VLflag=FALSE` for Granger causality.

`family` is a parameter of family of function for Generalized Linear Models function (glm). The default is `gaussian`.

Value

This function returns of a list of an adjacency matrix of causality where `adjMat[i,j]` is true if `TS[,i]` causes `TS[,j]`.

Examples

```
# Generate simulation data
TS <- MultipleSimulationVLtimeseries()
# Run the function
out<-multipleVLGrangerFunc(TS)
```

`multipleVLTransferEntropy`

multipleVLTransferEntropy

Description

multipleVLTransferEntropy is a function that infers Variable-lag Transfer Entropy of all pairwises of `m` time series `TS[,1],...TS[,m]`.

Usage

```
multipleVLTransferEntropy(
  TS,
  maxLag,
  nboot = 0,
  lx = 1,
  ly = 1,
  VLflag = TRUE,
  autoLagflag = TRUE
)
```

Arguments

`TS` is a numerical time series of effect where `TS[t,k]` is an element at time `t` of `k`th time series.

`maxLag` is a maximum possible time delay. The default is 0.2*length(Y).

`nboot` is a number of times of bootstrapping for RTransferEntropy::transfer_entropy() function.

`lx, ly`	are lag parameters of RTransferEntropy::transfer_entropy().
`VLflag`	is a flag of Granger causality choice: either `VLflag=TRUE` for VL-Granger or `VLflag=FALSE` for Granger causality.
`autoLagflag`	is a flag for enabling the automatic lag inference function. The default is true. If it is set to be true, then maxLag is set automatically using cross-correlation. Otherwise, if it is set to be false, then the function takes the maxLag value to infer Granger causality.

Value

This function returns of a list of an adjacency matrix of causality where `adjMat[i,j]` is true if `TS[,i]` causes `TS[,j]`.

Examples

```
# Generate simulation data
out1<-SimpleSimulationVLtimeseries()
TS<-cbind(out1$X,out1$Y)
# Run the function
out2<-multipleVLTransferEntropy(TS,maxLag=1)
```

`plotTimeSeries` *plotTimeSeries*

Description

plotTimeSeries is a function for visualizing time series

Usage

```
plotTimeSeries(X, Y, strTitle = "Time Series Plot", TSnames)
```

Arguments

`X`	is a 1st numerical time series
`Y`	is a 2nd numerical time series. If it is not supplied, the function plots only `X`.
`strTitle`	is a string of the plot title
`TSnames`	is a list of legend of `X,Y` where TSnames[1] is a legend of `X` and TSnames[2] is a legend of `Y`.

Value

This function returns an object of ggplot class.

Examples

```
# Generate simulation data
TS <- SimpleSimulationVLtimeseries()
# Run the function
plotTimeSeries(Y=TS$Y,X=TS$X)
```

`SimpleSimulationVLtimeseries` *SimpleSimulationVLtimeseries*

Description

SimpleSimulationVLtimeseries is a support function for generating time series `X,Y` where `X` VL-Granger-causes `Y`.

Usage

```
SimpleSimulationVLtimeseries(
  n = 200,
  lag = 5,
  YstFixInx = 110,
  YfnFixInx = 170,
  XpointFixInx = 100,
  arimaFlag = TRUE
)
```

Arguments

`n`	is length of time series.
`lag`	is a time lag between `X` and `Y` s.t. `Y[t]` is approximately `X[t-lag]`.
`YstFixInx`	is the starting point of variable lag part.
`YfnFixInx`	is the end point of variable lag part.
`XpointFixInx`	is a point in X s.t. `Y[YstFixInx:YfnFixInx]= X[XpointFixInx]` .
`arimaFlag`	is ARMA model flag. If it is true, then `X` is generated by ARMA model. If it is false, then `X` is generated by sampling of the standard normal distribution.

Value

This function returns a list of time series `X,Y` where `X` VL-Granger-causes `Y`.

Examples

```
# Generate simulation data
TS <- SimpleSimulationVLtimeseries()
```

VLGrangerFunc *VLGrangerFunc*

Description

VLGrangerFunc is a Variable-lag Granger Causality function. It tests whether `X` VL-Granger-causes `Y`.

Usage

```
VLGrangerFunc(
  Y,
  X,
  alpha = 0.05,
  maxLag,
  gamma = 0.5,
  autoLagflag = TRUE,
  family = gaussian
)
```

Arguments

`Y`	is a numerical time series of effect
`X`	is a numerical time series of cause
`alpha`	is a significance level of f-test to determine whether `X` Granger-causes `Y`. The default is 0.05.
`maxLag`	is a maximum possible time delay. The default is 0.2*length(Y).
`gamma`	is a parameter to determine whether `X` Granger-causes `Y` using BIC difference ratio. The default is 0.5.
`autoLagflag`	is a flag for enabling the automatic lag inference function. The default is true. If it is set to be true, then maxLag is set automatically using cross-correlation. Otherwise, if it is set to be false, then the function takes the maxLag value to infer Granger causality.
`family`	is a parameter of family of function for Generalized Linear Models function (glm). The default is `gaussian`.

Value

This function returns of whether `X` Granger-causes `Y`.

`ftest`	F-statistic of Granger causality.
`p.val`	A p-value from F-test.
`BIC_H0`	Bayesian Information Criterion (BIC) derived from `Y` regressing on `Y` past.
`BIC_H1`	Bayesian Information Criterion (BIC) derived from `Y` regressing on `Y`,`X` past.

`XgCsY`	The flag is true if `X` Granger-causes `Y` using BIC difference ratio where `BICDiffRatio >= gamma`.
`XgCsY_ftest`	The flag is true if `X` Granger-causes `Y` using f-test where `p.val>=alpha`.
`XgCsY_BIC`	The flag is true if `X` Granger-causes `Y` using BIC where `BIC_H0>=BIC_H1`.
`maxLag`	A maximum possible time delay.
`H0`	glm object of `Y` regressing on `Y` past.
`H1`	glm object of `Y` regressing on `Y,X` past.
`follOut`	is a list of variables from function `followingRelation`.
`BICDiffRatio`	Bayesian Information Criterion difference ratio: `(BIC_H0-BIC_H1)/BIC_H0`.

Examples

```
# Generate simulation data
TS <- SimpleSimulationVLtimeseries()
# Run the function
out<-VLGrangerFunc(Y=TS$Y,X=TS$X)
```

`VLTransferEntropy` *VLTransferEntropy*

Description

VLTransferEntropy is a Variable-lag Transfer Entropy function. It tests whether `X` VL-Transfer-Entropy-causes `Y`.

Usage

```
VLTransferEntropy(
  Y,
  X,
  maxLag,
  nboot = 0,
  lx = 1,
  ly = 1,
  VLflag = TRUE,
  autoLagflag = TRUE
)
```

Arguments

`Y`	is a numerical time series of effect
`X`	is a numerical time series of cause
`maxLag`	is a maximum possible time delay. The default is 0.2*length(Y).

nboot	is a number of times of bootstrapping for RTransferEntropy::transfer_entropy() function.
lx, ly	are lag parameters of RTransferEntropy::transfer_entropy().
VLflag	is a flag of Transfer Entropy choice: either VLflag=TRUE for VL-Transfer Entropy or VLflag=FALSE for Transfer Entropy.
autoLagflag	is a flag for enabling the automatic lag inference function. The default is true. If it is set to be true, then maxLag is set automatically using cross-correlation. Otherwise, if it is set to be false, then the function takes the maxLag value to infer Granger causality.

Value

This function returns of whether X (VL-)Transfer-Entropy-causes Y.

TEratio	is a Transfer Entropy ratio. If it is greater than one , then X causes Y.
res	is an object of output from RTransferEntropy::transfer_entropy()
follOut	is a list of variables from function followingRelation.
XgCsY_trns	The flag is true if X (VL-)Transfer-Entropy-causes Y using Transfer Entropy ratio ratio where TEratio >1 if X causes Y.

Examples

```
# Generate simulation data
TS <- SimpleSimulationVLtimeseries()
# Run the function
out<-VLTransferEntropy(Y=TS$Y,X=TS$X)
```

Index

checkMultipleSimulationVLtimeseries, 2

followingRelation, 3

GrangerFunc, 4

MultipleSimulationVLtimeseries, 5
multipleVLGrangerFunc, 6
multipleVLTransferEntropy, 7

plotTimeSeries, 8

SimpleSimulationVLtimeseries, 9

VLGrangerFunc, 10
VLTransferEntropy, 11