

Package ‘Sie2nts’

January 20, 2025

Type Package

Title Sieve Methods for Non-Stationary Time Series

Version 0.1.0

Description We provide functions for estimation and inference of locally-stationary time series using the sieve methods and bootstrapping procedure. In addition, it also contains functions to generate Daubechies and Coiflet wavelet by Cascade algorithm and to process data visualization.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

Repository CRAN

Suggests knitr, rmarkdown

VignetteBuilder knitr

Imports ggplot2, Matrix, plotly, stringr, RCurl, splines, methods,
utils, stats

NeedsCompilation no

Author Xiucan Ding [aut, cre, cph],
Chen Qian [aut, cph]

Maintainer Xiucan Ding <xiucaiding89@gmail.com>

Date/Publication 2023-01-18 08:30:09 UTC

Contents

auto.pacf.test	2
auto.test	3
bs.gene	4
bs.plot	5
fix.fit	6
fix.pacf	7
fix.pacf.test	8
fix.plot	8
fix.test	9

sie.auto.fit	10
sie.auto.pacf	11
sie.auto.plot	12
sie.plot.pacf	12
sie.predict	13

Index	14
--------------	-----------

auto.pacf.test	<i>The Test of Lag of Auto-Regressive (AR) Model Automatically</i>
----------------	--

Description

auto.pacf.test() generates a test of lag of AR model by choosing tuning parameter automatically.

Usage

```
auto.pacf.test(
  ts,
  lag = 3,
  b = 8,
  or = 4,
  type,
  alpha = 0.05,
  method = "LOOCV",
  threshold = 0,
  B.s = 1000
)
```

Arguments

ts	ts is the data set which is a time series data typically
lag	the lag for auto-regressive model, the default value is 3
b	the largest lag for auto-regressive model, the default value is 8, this parameter must be larger than lag
or	or indicates the order of spline and only used in Cspli type, default is 4 which indicates cubic spline
type	type indicates which type of basis is used. There are 31 types in this package
alpha	level of the test
method	method indicates which method used to choose optimal parameters, 3 methods in this package can be used.
threshold	threshold determines the bound for Elbow method
B.s	the number of statistics used in multiplier bootstrap, the default value is 1000

Details

In the parameter type, this package provides 32 types of basis including options "Legen" for Legendre polynomials, "Cheby" for first kind Chebyshev polynomials, "tri" for trigonometric polynomials, "cos" for cosine polynomials, "sin" for sine polynomials, "Cspli" for splines means Class of splines functions, in this option, the first input "c" is knots plus 2 that represent 0 and 1. "or" indicates the order of splines, so the number of basis is number of knots + 2 - 2 plus the number of order. When functions automatically choose the number of basis for splines, the number is not less than the order of spline. "db1" to "db20" for Daubechies1 wavelet basis to Daubechies20 wavelet basis and "cf1" to "cf5" for Coiflet1 wavelet basis to Coiflet5 wavelet basis. The package provides the wavelet tables are generated by Cascade algorithm using low-pass filter. If the exact values of wavelet are required, Recursion algorithm should be used. In the parameter method, it contains 3 options, the default option is "LOOCV", it uses Leave-One-Out Cross-Validation to choose the best tuning parameters. The second choice is "CV" which uses the Cross-Validation method, it takes $3 \cdot \log_2(n)$ size as validation set where n is the number of total observations. The third choice is "Elbow". This method similar as "IOOCV", however, it set the threshold manually. The function will choose the smallest tuning parameters once the value of LOOCV is less than threshold.

Value

p value of the test

References

- [1] Ding, Xiucui, and Zhou, Zhou. "Estimation and inference for precision matrices of nonstationary time series." *The Annals of Statistics* 48(4) (2020): 2455-2477.
- [2] Ding, Xiucui, and Zhou, Zhou. "Auto-regressive approximations to non-stationary time series, with inference and applications." Available online, 2021.

auto.test

The Test of Stability for Auto-Regressive (AR) Approximations Automatically

Description

auto.test() generates a test of Stability for AR Approximations by choosing tuning parameter automatically.

Usage

```
auto.test(
  ts,
  or = 4,
  type,
  alpha = 0.05,
  method = "LOOCV",
  threshold = 0,
  B.s = 1000
)
```

Arguments

ts	ts is the data set which is a time series data typically
or	indicates the order of spline and only used in Cspli type, default is 4 which indicates cubic spline
type	type indicates which type of basis is used. There are 31 types in this package
alpha	level of the test
method	method indicates which method used to choose optimal parameters, 3 methods in this package can be used
threshold	threshold determines the bound for Elbow method
B.s	the number of statistics used in multiplier bootstrap, the default value is 1000

Value

p value of the test

bs.gene

Generate Basis

Description

bs.gene() generates the value of k-th basis function. (The wavelet basis options return the full table)

Usage

```
bs.gene(type, k, point = 200, c = 10, or = 4, ops = "auto")
```

Arguments

type	type indicates which type of basis is used. There are 31 types in this package
k	k-th basis function
point	the number of values got from k-th basis function
c	c only used in Cspli which indicates the total number of knots to generate, the default is 10, c should not be less than k.(for splines, the true number of basis is c-2+or)
or	indicates the order of spline and only used in Cspli type, default is 4 which indicates cubic spline
ops	ops indicates the function uses existing table or theoretical way to generate, the default option is "auto"

Value

A data frame which contains the value of k-th basis function

References

[3] Chen, Xiaohong. "Large Sample Sieve Estimation of Semi-Nonparametric Models." Handbook of Econometrics, 6(B): 5549–5632,2007.

Examples

```
bs.gene("Legen", 2)
bs.gene("tri", 2, 300)
```

 bs.plot

Plots of Basis

Description

bs.plot() generates the plot of first k basis function.

Usage

```
bs.plot(type, k, or = 4, title = "")
```

Arguments

type	type indicates which type of basis is used (There are 31 types in this package)
k	The k is the number of basis functions represented (If wavelet are chosen, the real number of basis is 2^k . If Cspli is chosen, the real number of basis is $k-2+or$)
or	indicates the order of spline and only used in Cspli type, default is 4 which indicates cubic spline
title	give the title for the basis plot

Value

The plot of 1 to k basis functions

Examples

```
bs.plot("Legen", 2)
bs.plot("tri", 3)
```

fix.fit	<i>Estimate the Coefficients of Auto-Regressive (AR) Model by User Specifying</i>
---------	---

Description

fix.fit() estimates the coefficients of AR model by sieve methods with user specifying.

Usage

```
fix.fit(ts, c, b, type, or = 4, m = 500)
```

Arguments

ts	ts is the data set which is a time series data typically
c	c indicates the number of basis used to estimate (For wavelet, the real number of basis is 2^c . For Cubic Spline, the real number of basis is $c-2+or$)
b	b is the lag for auto-regressive model
type	type indicates which type of basis is used. There are 31 types in this package
or	indicates the order of spline and only used in Cspli type, default is 4 which indicates cubic spline
m	m indicates the number of points of coefficients to estimate

Value

A list contains 3 objects, the first is a matrix which contains estimates for each basis used in OLS, the second is a list contains estimates for coefficients in AR model and the last is a vector contains residuals

Examples

```
set.seed(137)
time.series = c()
n = 1024
v = 25
w = rnorm(n, 0, 1) / v
x_ini = runif(1,0,1)
for(i in 1:n){
  if(i == 1){
    time.series[i] = 0.2 + 0.6*cos(2*pi*(i/n))*x_ini + w[i] #
  } else{
    time.series[i] = 0.2 + 0.6*cos(2*pi*(i/n))*time.series[i-1] + w[i]
  }
}
res = fix.fit(time.series, c=5, b=1, type = "Legen")
cat(res$ols.coef)
plot.ts(res$ts.coef[[1]])
plot.ts(res$Residuals)
```

 fix.pacf

 Generate Partial Autocorrelation Function (PACF) by User Specifying

Description

fix.pacf() generates the PACF with fixed tuning parameters.

Usage

```
fix.pacf(ts, c, lag, type, or = 4, m = 500)
```

Arguments

ts	ts is the data set which is a time series data typically
c	c indicates the number of basis used to Estimate (For wavelet, the number of basis is 2^c . If Cspli is chosen, the real number of basis is $c-2+or$)
lag	lag b is the lag for auto-regressive model
type	type indicates which type of basis is used (There are 31 types in this package)
or	or indicates the order of spline and only used in Cspli type, default is 4 which indicates cubic spline
m	m indicates the number of points of coefficients to Estimate

Value

A vector which contains the PACF with specific lag

Examples

```
set.seed(137)
time.series = c()
n = 1024
v = 25
w = rnorm(n, 0, 1) / v
x_ini = runif(1,0,1)
for(i in 1:n){
  if(i == 1){
    time.series[i] = 0.2 + 0.6*cos(2*pi*(i/n))*x_ini + w[i] #
  } else{
    time.series[i] = 0.2 + 0.6*cos(2*pi*(i/n))*time.series[i-1] + w[i]
  }
}
fix.pacf(time.series, c=5, lag = 1, type = "Legen")
```

fix.pacf.test	<i>Testing Lag of Auto-Regressive (AR) Model</i>
---------------	--

Description

fix.pacf.test() generates a test of lags for AR Approximations.

Usage

```
fix.pacf.test(ts, c, type, or = 4, lag = 3, b = 8, B.s = 1000, m = 0)
```

Arguments

ts	ts is the data set which is a time series data typically
c	c indicates the number of basis used to estimate (For wavelet, the number of basis is 2^c . If Cspli is chosen, the real number of basis is $c-2+or$)
type	type indicates which type of basis is used. There are 31 types in this package
or	or indicates the order of spline and only used in Cspli type, default is 4 which indicates cubic spline
lag	lag determine the lag of AR Approximations. The default is 3
b	the largest lag for auto-regressive model, the default value is 8, this parameter must be larger than lag
B.s	the number of statistics used in multiplier bootstrap, the default value is 1000
m	the number of window size used in multiplier bootstrap, the default value is 0 which uses the minimum volatility method to determine the number

Value

It returns a list contains p value for each lag

fix.plot	<i>Plot Results of Estimating</i>
----------	-----------------------------------

Description

fix.plot() visualizes the estimation of coefficient.

Usage

```
fix.plot(res.fix.fit, type, title = "")
```


Arguments

res.fix.fit the output from fix.fit() function
 type type indicates which type of basis is used (There are 31 types in this package)
 title give the title for the fixed estimate plot

Value

A list which contains 3 plot related to the estimation of coefficient, Elbow point and cross validation in order

Examples

```
set.seed(137)
time.series = c()
n = 1024
v = 25
w = rnorm(n, 0, 1) / v
x_ini = runif(1,0,1)
for(i in 1:n){
  if(i == 1){
    time.series[i] = 0.2 + 0.6*cos(2*pi*(i/n))*x_ini + w[i] #
  } else{
    time.series[i] = 0.2 + 0.6*cos(2*pi*(i/n))*time.series[i-1] + w[i]
  }
}

res1 = fix.fit(time.series, 5, 1, type = "Legen")
fix.plot(res1, "Legen")
```

fix.test	<i>The Test of Stability for Auto-Regressive (AR) Approximations With Fixed Parameters</i>
----------	--

Description

fix.test() generates a test of Stability for AR Approximations with fixed parameters.

Usage

```
fix.test(ts, c, b, type, or = 4, B.s = 1000, m = 0)
```

Arguments

ts ts is the data set which is a time series data typically
 c c indicates the number of basis used to estimate (For wavelet, the number of basis is 2^c . If Cspli is chosen, the real number of basis is $c-2+or$)
 b b is the lag for auto-regressive model

type	type indicates which type of basis is used. There are 31 types in this package
or	indicates the order of spline and only used in Cspli type, default is 4 which indicates cubic spline
B.s	the number of statistics used in multiplier bootstrap, the default value is 1000
m	the number of window size used in multiplier bootstrap, the default value is 0 which uses the minimum volatility method to determine the number

Value

p value of the test

sie.auto.fit	<i>Estimate the Coefficients of Auto-Regressive (AR) Model Automatically</i>
--------------	--

Description

sie.auto.fit() estimates the coefficients of AR model by sieve methods with 2 cross validation methods and elbow method.

Usage

```
sie.auto.fit(ts, type, or = 4, method = "LOOCV", m = 500, threshold = 0)
```

Arguments

ts	ts is the data set which is a time series data typically
type	type indicates which type of basis is used. There are 31 types in this package
or	indicates the order of spline and only used in Cspli type, default is 4 which indicates cubic spline
method	method indicates which method used to choose optimal parameters, 3 methods in this package can be used
m	m indicates the number of points of coefficients to estimate
threshold	threshold determines the bound for Elbow method

Value

A list contains 4 objects, the first is estimates for coefficients in AR model, the second is cross validation table, the third is estimates for each basis used in OLS and the last is optimal parameters

 sie.auto.pacf

Generate Partial Autocorrelation Function (PACF) Automatically

Description

sie.auto.pacf() generates the PACF from 1 to lag automatically.

Usage

```
sie.auto.pacf(ts, c, lag, type, or = 4, m = 500)
```

Arguments

ts	ts is the data set which is a time series data typically
c	c indicates the number of basis used to estimate (For wavelet, the number of basis is 2^c . If Cspli is chosen, the real number of basis is $c-2+or$)
lag	lag b is the lag for auto-regressive model
type	type indicates which type of basis is used (There are 31 types in this package)
or	indicates the order of spline and only used in Cspli type, default is 4 which indicates cubic spline
m	m indicates the number of points of coefficients to estimate

Value

A list contains the PACF in each lag

Examples

```
set.seed(137)
time.series = c()
n = 1024
v = 25
w = rnorm(n, 0, 1) / v
x_ini = runif(1,0,1)
for(i in 1:n){
  if(i == 1){
    time.series[i] = 0.2 + 0.6*cos(2*pi*(i/n))*x_ini + w[i] #
  } else{
    time.series[i] = 0.2 + 0.6*cos(2*pi*(i/n))*time.series[i-1] + w[i]
  }
}
sie.auto.pacf(time.series, 5, 1, "Legen")
```

sie.auto.plot *Plot the Estimate Results by Automatic Fitting*

Description

sie.auto.plot() visualizes the estimation of coefficient, gives the elbow plot and represents the cross validation result.

Usage

```
sie.auto.plot(res.auto.fit, type, title = "")
```

Arguments

res.auto.fit	the output from sie.auto.fit() function
type	type indicates which type of basis is used (There are 31 types in this package)
title	give the title for the auto estimate plot

Value

A list which contains 3 plot related to the estimation of coefficient, Elbow point and cross validation in order

sie.plot.pacf *Plot Partial Autocorrelation Function (PACF)*

Description

sie.plot.pacf() shows the PACF with different lag.

Usage

```
sie.plot.pacf(ts, c, lag, type, ops = "2d", title = "", m = 500, or = 4)
```

Arguments

ts	ts is the data set which is a time series data typically
c	c indicates the number of basis used to estimate (For wavelet, the number of basis is 2^c . If Cspli is chosen, the real number of basis is $c-2+or$)
lag	lag b is the lag for auto-regressive model
type	type indicates which type of basis is used (There are 32 types in this package)
ops	choose 2D plot or 3D plot ("2d" indicates 2D plot and "3d" indicates 3D plot)
title	give the title for the pacf plot
m	m indicates the number of points of coefficients to estimate
or	or indicates the order of spline, default is 4 which indicates cubic spline

Value

The plot of pacf basis on the time series data

 sie.predict

Predicting Time Series With H Steps

Description

predict.nts() predicts the time series data basis on the estimation.

Usage

```
sie.predict(ts, esti.li, h)
```

Arguments

ts	The data set which is a time series data typically
esti.li	The output from fix.fit() or sie.auto.fit() function
h	h indicates the number of forecasting points

Value

A vector which contains h forecasting points

Examples

```
set.seed(137)
time.series = c()
n = 1024
v = 25
w = rnorm(n, 0, 1) / v
x_ini = runif(1,0,1)
for(i in 1:n){
  if(i == 1){
    time.series[i] = 0.2 + 0.6*cos(2*pi*(i/n))*x_ini + w[i] #
  } else{
    time.series[i] = 0.2 + 0.6*cos(2*pi*(i/n))*time.series[i-1] + w[i]
  }
}
res1.2 = fix.fit(time.series, 5, 1, "Legen")
sie.predict(time.series, res1.2, 5)
```

Index

auto.pacf.test, 2
auto.test, 3

bs.gene, 4
bs.plot, 5

fix.fit, 6
fix.pacf, 7
fix.pacf.test, 8
fix.plot, 8
fix.test, 9

sie.auto.fit, 10
sie.auto.pacf, 11
sie.auto.plot, 12
sie.plot.pacf, 12
sie.predict, 13