Package 'gdpc'

November 19, 2023

Type Package

Title Generalized Dynamic Principal Components

Version 1.1.4

Date 2023-11-19

Author Daniel Peña <daniel.pena@uc3m.es>,
Ezequiel Smucler <ezequiels.90@gmail.com>,
Victor Yohai <vyohai@dm.uba.ar>

Maintainer Ezequiel Smucler <ezequiels.90@gmail.com>

Description Functions to compute the Generalized Dynamic Principal Components
introduced in Peña and Yohai (2016) <DOI:10.1080/01621459.2015.1072542>. The implementation
includes an automatic procedure proposed in Peña, Smucler and Yohai (2020) <DOI:10.18637/jss.v092.c02>
for the identification of both the number of lags to be used
in the generalized dynamic principal components as well as the number of components required
for a given reconstruction accuracy.

License GPL (>= 2)

Imports xts, zoo, methods, Rcpp (>= 0.12.7), parallel, doParallel,
foreach

LinkingTo Rcpp, RcppArmadillo (>= 0.7.500.0.0)

Suggests testthat, R.rsp

Depends R (>= 3.3.0)

NeedsCompilation yes

Encoding UTF-8

VignetteBuilder R.rsp

Repository CRAN

Date/Publication 2023-11-19 15:20:02 UTC

R topics documented:

auto.gdpc *Automatic Fitting of Generalized Dynamic Principal Components*

Description

Computes Generalized Dynamic Principal Components. The number of components can be supplied by the user or chosen automatically so that a given proportion of variance is explained. The number of lags is chosen automatically using one of the following criteria: Leave-one-out cross-validation, an AIC type criterion, a BIC type criterion or a criterion based on a proposal of Bai and Ng (2002). See Peña, Smucler and Yohai (2020) for more details.

Usage

```
auto.gdpc(Z, crit = 'LOO', normalize = 1, auto_comp = TRUE, expl_var = 0.9,
          num_comp = 5, tol = 1e-4, k_max = 10,
          niter_max = 500, ncores = 1, verbose = FALSE)
```

Arguments

`Z`	Data matrix. Each column is a different time series.
`crit`	A string specifying the criterion to be used. Options are 'LOO', 'AIC', 'BIC' and 'BNG'. Default is 'LOO'. See Details below.
`normalize`	Integer. Either 1, 2 or 3. Indicates whether the data should be standardized. Default is 1. See Details below.
`auto_comp`	Logical. If TRUE compute components until the proportion of explained variance is equal to expl_var, otherwise use num_comp components. Default is TRUE.
`expl_var`	A number between 0 and 1. Desired proportion of explained variance (only used if auto_comp==TRUE). Default is 0.9.
`num_comp`	Integer. Number of components to be computed (only used if auto_comp==FALSE). Default is 5.
`tol`	Relative precision. Default is 1e-4.

`k_max`	Integer. Maximum possible number of lags. Default is 10.
`niter_max`	Integer. Maximum number of iterations. Default is 500.
`ncores`	Integer. Number of cores to be used for parallel computations. Default is 1.
`verbose`	Logical. Should progress be reported? Default is FALSE.

Details

Suppose the data matrix consists of m series of length T. Let $\boldsymbol{f}$ be the dynamic principal component defined using k lags, let R be the corresponding matrix of residuals and let $\Sigma = (R'R)/T$.

If crit = 'LOO' the number of lags is chosen among $0, \dots, k_{max}$ as the value k that minimizes the leave-one-out (LOO) cross-validation mean squared error, given by

$$LOO = \frac{1}{Tm} \sum_{i=1}^{m} \sum_{t=1}^{T} \frac{R_{t,i}^2}{(1 - h_{t,t})^2},$$

where $h_{t,t}$ are the diagonal elements of the hat matrix $H = F(F'F)^{-1}F'$, with F being the $T \times (k+2)$ matrix with rows $(f_{t-k}, f_{t-k+1}, \dots, f_t, 1)$.

If crit = 'AIC' the number of lags is chosen among $0, \dots, k_{max}$ as the value k that minimizes the following AIC type criterion

$$AIC = T \log(trace(\Sigma)) + 2m(k+2).$$

If crit = 'BIC' the number of lags is chosen among $0, \dots, k_{max}$ as the value k that minimizes the following BIC type criterion

$$BIC = T \log(trace(\Sigma)) + m(k+2)\log(T).$$

If crit = 'BNG' the number of lags is chosen among $0, \dots, k_{max}$ as the value k that minimizes the following criterion

$$BNG = \min(T, m) \log(trace(\Sigma)) + (k+1)\log(\min(T, m)).$$

This is an adaptation of a criterion proposed by Bai and Ng (2002).

For problems of relatively small dimension, say $T \geq m10$, 'AIC' can can give better results than the default 'LOO'.

If normalize = 1, the data is analyzed in the original units, without mean and variance standarization. If normalize = 2, the data is standardized to zero mean and unit variance before computing the principal components, but the intercepts and loadings are those needed to reconstruct the original series. If normalize = 3 the data are standardized as in normalize = 2, but the intercepts and the loadings are those needed to reconstruct the standardized series. Default is normalize = 1.

Value

An object of class `gdpcs`, that is, a list of length equal to the number of computed components. The i-th entry of this list is an object of class `gdpc`, that is, a list with entries

`expart`	Proportion of the variance explained by the first i components.

`mse` Mean squared error of the reconstruction using the first i components.

`crit` The value of the criterion of the reconstruction, according to what the user specified.

`k` Number of lags chosen.

`alpha` Vector of intercepts corresponding to f.

`beta` Matrix of loadings corresponding to f. Column number k is the vector of $k-1$ lag loadings.

`f` Coordinates of the i-th dynamic principal component corresponding to the periods $1, \ldots, T$.

`initial_f` Coordinates of the i-th dynamic principal component corresponding to the periods $-k+1, \ldots, 0$. Only for the case $k > 0$, otherwise 0.

`call` The matched call.

`conv` Logical. Did the iterations converge?

`niter` Integer. Number of iterations.

`components`, `fitted`, `plot` and `print` methods are available for this class.

Author(s)

Daniel Peña, Ezequiel Smucler, Victor Yohai

References

Bai J. and Ng S. (2002). “Determining the Number of Factors in Approximate Factor Models.” Econometrica, 70(1), 191–221.

Peña D., Smucler E. and Yohai V.J. (2020). “gdpc: An R Package for Generalized Dynamic Principal Components.” Journal of Statistical Software, 92(2), 1-23.

See Also

`gdpc`, `plot.gdpc`, `plot.gdpcs`, `fitted.gdpcs`, `components.gdpcs`

Examples

```
T <- 200 #length of series
m <- 200 #number of series
set.seed(1234)
f <- rnorm(T + 1)
x <- matrix(0, T, m)
u <- matrix(rnorm(T * m), T, m)
for (i in 1:m) {
   x[, i] <- 10 * sin(2 * pi * (i/m)) * f[1:T] + 10 * cos(2 * pi * (i/m)) * f[2:(T + 1)] + u[, i]
}
#Choose number of lags using the LOO criterion.
#k_max=3 to keep computation time low
autofit <- auto.gdpc(x, k_max = 3)
autofit
fit_val <- fitted(autofit, 1) #Get fitted values
```

```
resid <- x - fit_val #Residuals
plot(autofit, which_comp = 1) #Plot component
```

components *Generic Function for Getting Components From an Object*

Description

Generic function for getting components from an object.

Usage

```
components(object, which_comp)
```

Arguments

`object`	An object. Currently there is a method for objects of class `gdpcs`.
`which_comp`	Numeric vector indicating which components to get. Default is 1.

Value

A matrix whose columns are the desired components.

Author(s)

Daniel Peña, Ezequiel Smucler, Victor Yohai

components.gdpcs *Get Generalized Dynamic Principal Components From a gdpcs Object*

Description

Get Generalized Dynamic Principal Components from a gdpcs object.

Usage

```
## S3 method for class 'gdpcs'
components(object, which_comp = 1)
```

Arguments

`object`	An object of class gdpcs, usually the result of auto.gdpc.
`which_comp`	Numeric vector indicating which components to get. Default is 1.

Value

A matrix whose columns are the desired dynamic principal components.

Author(s)

Daniel Peña, Ezequiel Smucler, Victor Yohai

See Also

`gdpc`, `auto.gdpc`, `plot.gdpc`

Examples

```
T <- 200 #length of series
m <- 200 #number of series
set.seed(1234)
f <- rnorm(T + 1)
x <- matrix(0, T, m)
u <- matrix(rnorm(T * m), T, m)
for (i in 1:m) {
   x[, i] <- 10 * sin(2 * pi * (i/m)) * f[1:T] + 10 * cos(2 * pi * (i/m)) * f[2:(T + 1)] + u[, i]
}
#Choose number of lags using the LOO criterion.
#k_max=2 to keep computation time low
autofit <- auto.gdpc(x, k_max = 2, auto_comp = FALSE, num_comp = 2)
comps <- components(autofit, which_comp = c(1,2))
```

`fitted.gdpcs` *Get Reconstructed Time Series From a gdpcs Object*

Description

Get reconstructed time series from a gdpcs object.

Usage

```
  ## S3 method for class 'gdpcs'
fitted(object, num_comp = 1, ...)
```

Arguments

`object`	An object of class gdpcs, usually the result of auto.gdpc.
`num_comp`	Integer indicating how many components to use for the reconstruction. Default is 1.
`...`	Additional arguments for compatibility.

Value

A matrix that is the reconstruction of the original series.

Author(s)

Daniel Peña, Ezequiel Smucler, Victor Yohai

See Also

gdpc, auto.gdpc, plot.gdpc

Examples

```
T <- 200 #length of series
m <- 200 #number of series
set.seed(1234)
f <- rnorm(T + 1)
x <- matrix(0, T, m)
u <- matrix(rnorm(T * m), T, m)
for (i in 1:m) {
 x[, i] <- 10 * sin(2 * pi * (i/m)) * f[1:T] + 10 * cos(2 * pi * (i/m)) * f[2:(T + 1)] + u[, i]
}
#Choose number of lags using the LOO criterion.
#k_max=2 to keep computation time low
autofit <- auto.gdpc(x, k_max = 2, auto_comp = FALSE, num_comp = 2)
recons <- fitted(autofit, num_comp = 2)
```

gdpc *Generalized Dynamic Principal Components*

Description

Computes a single Generalized Dynamic Principal Component with a given number of lags.

Usage

```
gdpc(Z, k, f_ini = NULL, tol = 1e-4, niter_max = 500, crit = 'LOO')
```

Arguments

`Z` Data matrix. Each column is a different time series.

`k` Integer. Number of lags to use.

`f_ini` (Optional). Numeric vector. Starting point for the iterations. If no argument is passed the ordinary (non-dynamic) first principal component completed with k lags is used.

`tol` Relative precision. Default is 1e-4.

`niter_max` Integer. Maximum number of iterations. Default is 500.

`crit` A string specifying the criterion to be used to evaluate the fitted model. Options are 'LOO', 'AIC', 'BIC' and 'BNG'. Default is 'LOO'.

Details

See auto.gdpc for the definition of criterion that is part of the output of this function.

Value

An object of class gdpc, that is, a list with entries:

expart Proportion of the variance explained.

mse Mean squared error.

crit The value of the criterion of the reconstruction, according to what the user specified.

k Number of lags used.

alpha Vector of intercepts corresponding to f.

beta Matrix of loadings corresponding to f. Column number k is the vector of $k-1$ lag loadings.

f Coordinates of the first dynamic principal component corresponding to the periods $1,\ldots,T$.

initial_f Coordinates of the first dynamic principal component corresponding to the periods $-k+1,\ldots,0$. Only for the case $k>0$, otherwise 0.

call The matched call.

conv Logical. Did the iterations converge?

niter Integer. Number of iterations.

fitted, plot and print methods are available for this class.

Author(s)

Daniel Peña, Ezequiel Smucler, Victor Yohai

See Also

auto.gdpc, plot.gdpc

Examples

```
T <- 200 #length of series
m <- 500 #number of series
set.seed(1234)
f <- rnorm(T + 1)
x <- matrix(0, T, m)
u <- matrix(rnorm(T * m), T, m)
for (i in 1:m) {
   x[, i] <- 10 * sin(2 * pi * (i/m)) * f[1:T] + 10 * cos(2 * pi * (i/m)) * f[2:(T + 1)] + u[, i]
}
fit <- gdpc(x, k = 1) #find first DPC with one lag
fit
par(mfrow = c(1, 2)) #plot loadings
plot(fit, which = 'Loadings', which_load = 0, xlab = '', ylab = '')
plot(fit, which = 'Loadings', which_load = 1, xlab = '', ylab = '')
```

ipi91 *Industrial Production Index (IPI) of France, Germany, Italy, United Kingdom, USA and Japan*

Description

Six series corresponding to the Industrial Production Index (IPI) of France, Germany, Italy, United Kingdom, USA and Japan. Monthly data from January 1991 to December 2012.

Usage

```
data(ipi91)
```

Format

A matrix time series with 264 observations on the following 6 variables.

`France` IPI of France.

`Germany` IPI of Germany.

`Italy` IPI of Italy.

`United Kingdom` IPI of United Kingdom.

`USA` IPI of USA.

`Japan` IPI of Japan.

Examples

```
data(ipi91)
plot(ipi91, plot.type = 'multiple', main = 'Industrial Production Index')
## Not run:
#Compute first GDPC with nine lags; this may take a bit.
gdpc_ipi <- gdpc(ipi91, 9, niter_max = 1500)
#Plot the component
plot(gdpc_ipi, which = 'Component', ylab = '')
#Get reconstruction of the time series and plot
recons <- fitted(gdpc_ipi)
colnames(recons) <- colnames(ipi91)
plot(recons, main = 'Fitted values')

## End(Not run)
```

`plot.gdpc` *Plot Generalized Dynamic Principal Components*

Description

Plots a gdpc object.

Usage

```
## S3 method for class 'gdpc'
plot(x, which = 'Component', which_load = 0, ...)
```

Arguments

x	An object of class gdpc, usually the result of gdpc or one of the entries of the result of auto.gdpc.
which	String. Indicates what to plot, either 'Component' or 'Loadings'. Default is 'Component'.
which_load	Lag number indicating which loadings should be plotted. Only used if which = 'Loadings'. Default is 0.
...	Additional arguments to be passed to the plotting functions.

Author(s)

Daniel Peña, Ezequiel Smucler, Victor Yohai

See Also

gdpc, auto.gdpc, plot.gdpcs

Examples

```
T <- 200 #length of series
m <- 200 #number of series
set.seed(1234)
f <- rnorm(T + 1)
x <- matrix(0, T, m)
u <- matrix(rnorm(T * m), T, m)
for (i in 1:m) {
   x[, i] <- 10 * sin(2 * pi * (i/m)) * f[1:T] + 10 * cos(2 * pi * (i/m)) * f[2:(T + 1)] + u[, i]
}
#Choose number of lags using the LOO type criterion.
#k_max=3 to keep computation time low
autofit <- auto.gdpc(x, k_max = 3)
plot(autofit[[1]], xlab = '', ylab = '')
```

plot.gdpcs *Plot Generalized Dynamic Principal Components*

Description

Plots a gdpcs object.

Usage

```
## S3 method for class 'gdpcs'
plot(x, which_comp = 1, plot.type = 'multiple', ...)
```

Arguments

`x` An object of class gdpcs, usually the result of auto.gdpc.

`which_comp` Numeric vector indicating which components to plot. Default is 1.

`plot.type` Argument to be passed to `plot.zoo`. Used only when the original data set was stored in an object of class `zoo`. Default is 'multiple'.

`...` Additional arguments to be passed to the plotting functions.

Author(s)

Daniel Peña, Ezequiel Smucler, Victor Yohai

See Also

`gdpc`, `auto.gdpc`, `plot.gdpc`

Examples

```
T <- 200 #length of series
m <- 200 #number of series
set.seed(1234)
f <- rnorm(T + 1)
x <- matrix(0, T, m)
u <- matrix(rnorm(T * m), T, m)
for (i in 1:m) {
   x[, i] <- 10 * sin(2 * pi * (i/m)) * f[1:T] + 10 * cos(2 * pi * (i/m)) * f[2:(T + 1)] + u[, i]
}
#Choose number of lags using the LOO criterion.
#k_max=2 to keep computation time low
autofit <- auto.gdpc(x, k_max = 2, auto_comp = FALSE, num_comp = 2)
autofit
plot(autofit, which_comp = c(1,2), xlab = '', ylab = '')
```

pricesSP50 *Stock Prices of the First 50 Components of S&P500*

Description

Fifty series corresponding to the stock prices of the first 50 components of the Standard&Poor's 500 index. Five hundred daily observations starting 1/1/2010.

Usage

```
data(pricesSP50)
```

Format

A matrix time series with 500 observations on the stock prices of the first 50 components of the Standard&Poor's 500 index.

Examples

```
data(pricesSP50)
## Not run:
#Plot the first four series
plot(pricesSP50[, 1:4], main = 'Four components of the S&P500 index')
#Compute GDPCs; this may take a bit.
fit_SP <- auto.gdpc(pricesSP50, normalize = 2, niter_max = 1000, ncores= 4)
fit_SP
#Get reconstruction and plot
recons <- fitted(fit_SP, num_comp = 2)
colnames(recons) <- colnames(pricesSP50)
plot(recons[, 1:4], main = 'Reconstruction of four components of the S&P500 index')

## End(Not run)
```

Index