

Package: pcdpca (via r-universe)

September 18, 2024

Title Dynamic Principal Components for Periodically Correlated Functional Time Series

Version 0.4

Description Method extends multivariate and functional dynamic principal components to periodically correlated multivariate time series. This package allows you to compute true dynamic principal components in the presence of periodicity. We follow implementation guidelines as described in Kidzinski, Kokoszka and Jouzdani (2017), in Principal component analysis of periodically correlated functional time series [<arXiv:1612.00040 >](https://arxiv.org/abs/1612.00040).

Depends R (>= 3.3.1)

Imports freqdom, fda

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Author Lukasz Kidzinski [aut, cre], Neda Jouzdani [aut], Piotr Kokoszka [aut]

Maintainer Lukasz Kidzinski <lukasz.kidzinski@stanford.edu>

Repository CRAN

Date/Publication 2017-09-03 05:17:18 UTC

Contents

pcdpca	2
pcdpca.inverse	3
pcdpca.scores	4

Index	5
--------------	----------

pcdpca

Compute periodically correlated DPCA filter coefficients

Description

For a given periodically correlated multivariate process X eigendecompose its spectral density and use an inverse fourier transform to get coefficients of the optimal filters.

Usage

```
pcdpca(X, period = NULL, q = 30, freq = (-1000:1000/1000) * pi)
```

Arguments

<code>X</code>	multivariate stationary time series
<code>period</code>	period of the periodic time series
<code>q</code>	window for spectral density estimation as in spectral.density
<code>freq</code>	frequency grid to estimate on as in spectral.density

Value

principal components series

References

Kidzinski, Kokoszka, Jouzdani Dynamic principal components of periodically correlated functional time series Research report, 2016

See Also

[pcdpca.inverse](#), [pcdpca.scores](#)

Examples

```
## Prepare some process
library(fda)
library(freedom)

MSE = function(X,Y=0){ sum((X-Y)**2) / nrow(X) }

d = 7
n = 100
A = t(t(matrix(rnorm(d*n),ncol=d,nrow=n))*7:1)
B = t(t(matrix(rnorm(d*n),ncol=d,nrow=n))*7:1)
C = t(t(matrix(rnorm(d*n),ncol=d,nrow=n))*7:1)

X = matrix(0,ncol=d,nrow=3*n)
X[3*(1:n) - 1,] = A
```

```

X[3*(1:n) - 2,] = A + B
X[3*(1:n) ,] = 2*A - B + C

basis = create.fourier.basis(nbasis=7)
X.fd = fd(t(Re(X)),basis=basis)
plot(X.fd)

## Hold out some datapoints
train = 1:(50*3)
test = (50*3) : (3*n)

## Static PCA ##
PR = prcomp(as.matrix(X[train,]))
Y1 = as.matrix(X) %*% PR$rotation
Y1[,-1] = 0
Xpca.est = Y1 %*% t(PR$rotation)

## Dynamic PCA ##
XI.est = dpca(as.matrix(X[train,]),
             q=3,
             freq=pi*(-150:150/150),
             Ndpc=1) # finds the optimal filter
Y.est = freqdom::filter.process(X, XI.est$filters )
Xdpca.est = freqdom::filter.process(Y.est, t(rev(XI.est$filters))) ) # deconvolution

## Periodically correlated PCA ##
XI.est.pc = pcdpca(as.matrix(X[train,]),
                 q=3,
                 freq=pi*(-150:150/150),period=3) # finds the optimal filter
Y.est.pc = pcdpca.scores(X, XI.est.pc) # applies the filter
Y.est.pc[,-1] = 0 # forces the use of only one component
Xpcdpca.est = pcdpca.inverse(Y.est.pc, XI.est.pc) # deconvolution

## Results
cat("NMSE PCA = ")
r0 = MSE(X[test,],Xpca.est[test,]) / MSE(X[test,],0)
cat(r0)
cat("\nNMSE DPCA = ")
r1 = MSE(X[test,],Xdpca.est[test,]) / MSE(X[test,],0)
cat(r1)
cat("\nNMSE PCDPCA = ")
r2 = MSE(X[test,],Xpcdpca.est[test,]) / MSE(X[test,],0)
cat(r2)
cat("\n")

```

Description

For given scores Y and dynamic principal components XI retrieve a series from which scores Y were calculated. This procedure should be seen as the inverse of [pcdpca.scores](#).

Usage

```
pcdpca.inverse(Y, XI)
```

Arguments

Y	scores process
XI	principal components series

Value

Retrieved process X

References

Kidzinski, Kokoszka, Jouzdani Dynamic principal components of periodically correlated functional time series Research report, 2016

See Also

[pcdpca.scores](#), [pcdpca](#)

pcdpca.scores

Compute periodically correlated DPCA scores, given the filters XI

Description

Compute periodically correlated DPCA scores, given the filters XI

Usage

```
pcdpca.scores(X, XI)
```

Arguments

X	multivariate time series
XI	series of filters returned from pcdpca

Index

* **pcdpca**

pcdpca.scores, 4

pcdpca, 2, 4

pcdpca.inverse, 2, 3

pcdpca.scores, 2, 4, 4

spectral.density, 2