QMDPCA [QMDPCA]

Quadratic Moving Dynamic Principal Component Analysis for Non-Stationary Multivariate Time Series

Description
This function reduces the dimension of non-stationary (and stationary) multivariate time series by performing eigenanalysis on the quadratic moving cross-covariance matrix of the extended data matrix up to some specified lag. Notice that the following libraries are needed to be installed before using the MDPCA function: library(roll); library(expm).

Usage
QMDPCA(x, w, l)

Arguments

x
a T-by-m data matrix, where the rows are "T" time points, and the columns are "m" variables

w
window width (i.e. window length)

l
number of lagged series to be included in the calculation of QMDPCA

Value

xdata
returns the extended data matrix of x

F
returns the quadratic moving cross-covariance matrix of the extended data matrix xdata

Lambda
returns the eigenvalues of the matrix F

U
returns the eigenvectors of the matrix F

Note

step1: Build the extended data matrix (i.e. xdata) and obtain the eigenvalues and eigenvectors of its quadratic moving cross-covariance matrix. step2: Transfer extended data matrix (i.e. xdata) using eigenvectors (i.e. U) that correspond to the largest eigenvalues (i.e. Lambda). For example, if we find the first two eigenvalues to be large enough, then we can choose the corresponding two eigenvectors to obtain the final results (i.e. two QMDPCs).

Author(s)

Fayed Alshammri

References


Examples

## This is Example 2 of Alshammri and Pan (2020).
## The data matrix X is a non-stationary time series with m=5 and T=1000.

m=5; T=1000

# Generate x_t
X=mat.or.vec(m, T)

a1=arima.sim(list(order=c(1,1,1),ar=0.55,ma=-0.8),n=T,sd=1)
X[1,]=a1[2:(T+1)]

a2=arima.sim(list(order=c(1,1,1),ar=-0.65,ma=0.45),n=T,sd=1)
X[2,]=a2[2:(T+1)]

a3=arima.sim(list(order=c(1,1,1),ar=0.45,ma=1.6),n=T,sd=1)
X[3,]=a3[2:(T+1)]

a4=arima.sim(list(order=c(1,1,1),ar=-0.8,ma=-0.9),n=T,sd=1)
X[4,]=a4[2:(T+1)]

a5=arima.sim(list(order=c(1,1,1),ar=0.85,ma=-2.2),n=T,sd=1)
X[5,]=a5[2:(T+1)]

X=t(X)

X=ts(X)

## apply QMDPCA with 100 window length and one lagged series.
Analysis=QMDPCA(X, 100, 1)

U=Analysis$U

Lambda=Analysis$Lambda

F=Analysis$xdata

# Example: If we find the first two eigenvalues to be large enough, then we can choose the corresponding two eigenvectors to obtain the final results (i.e. two QMDPCs)
Lambda[1:2]/sum(Lambda) * 100

# Final results (i.e. two QMDPCs)
Transform=ts(Transform)

[Package QMDPCA version 0.1.0 Index]