Package ‘RTFA’

January 20, 2025

Type Package

Title Robust Factor Analysis for Tensor Time Series

Version 0.1.0

Author Matteo Barigozzi [aut],
Yong He [aut],
Lorenzo Trapani [aut],
Lingxiao Li [aut, cre]

Maintainer Lingxiao Li <lilingxiao@mail.sdu.edu.cn>

Description Tensor Factor Models (TFM) are appealing dimension reduction tools for high-order tensor time series, and have wide applications in economics, finance and medical imaging. We propose an one-step projection estimator by minimizing the least-square loss function, and further propose a robust estimator with an iterative weighted projection technique by utilizing the Huber loss function. The methods are discussed in Barigozzi et al. (2022) <arXiv:2206.09800>, and Barigozzi et al. (2023) <arXiv:2303.18163>.

License GPL (>= 2)

Depends R (>= 3.5.0)

Imports rTensor, tensor

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2023-04-10 14:00:05 UTC

Contents

TFM_est *Estimation of Factor Model for High-Dimensional Tensor Time Series*

Description

This function is to estimate the tensor factor model via four different methods, namely the initial estimation without initial (IE), one-step projection estimation (PE), iterative projection estimation (iPE) and iterative weighted projection estimation by Huber loss (HUBER).

Usage

```
TFM_est(x, r, method = "PE", tol = 1e-04, maxiter = 100)
```

Arguments

`x`	$T \times p_1 \times \cdots \times p_K$ tensor-valued time series.
`r`	input rank of the factor tensor.
`method`	character string, specifying the type of the estimation method to be used. `"IE",` Initial estimation, without projection. `"PE",` One-step projection estimation. `"iPE",` Iterative projection estimation. `"HUBER",` Iterative weighted projection estimation based on huber loss function.
`tol`	tolerance in terms of the Frobenius norm.
`maxiter`	maximum number of iterations if error stays above `tol`.

Details

See Barigozzi et al. (2022) and Barigozzi et al. (2023) for details.

Value

return a list containing the following:

`Ft` estimated factor processes of dimension $T \times r_1 \times r_2 \times \cdots \times r_K$.

`Ft.all` Summation of factor processes over time, of dimension $r_1, r_2, \cdots, r_K$.

`Q` a list of estimated factor loading matrices $Q_1, Q_2, \cdots, Q_K$.

`x.hat` fitted signal tensor, of dimension $T \times p_1 \times p_2 \times \cdots \times p_K$.

`niter` number of iterations.

`fnorm.resid` Frobenius norm of residuals, divide the Frobenius norm of the original tensor.

Author(s)

Matteo Barigozzi, Yong He, Lingxiao Li, Lorenzo Trapani.

References

Barigozzi M, He Y, Li L, Trapani L. Robust Estimation of Large Factor Models for Tensor-valued Time Series. <arXiv:2206.09800>

Barigozzi M, He Y, Li L, Trapani L. Statistical Inference for Large-dimensional Tensor Factor Model by Iterative Projection. <arXiv:2303.18163>

Examples

```
library(rTensor)
set.seed(1234)
p <- c(12,16,20)  # dimensions of tensor time series
r <- c(3,4,5)  # dimensions of factor series
A<-list()
Q<-list()
for(i in 1:3){
  A[[i]]<-matrix(rnorm(p[i]*r[i],0,1),p[i],r[i])
  Q[[i]]<-eigen(A[[i]]%*%t(A[[i]]))$vectors
}
T<-100
F<-array(NA,c(T,r))
E<-array(NA,c(T,p))
S<-array(NA,c(T,p))
X<-array(NA,c(T,p))
for(t in 1:T){
  F[t,,,]<-array(rnorm(prod(r),0,1),r)
  E[t,,,]<-array(rnorm(prod(p),0,1),p)
  S[t,,,]<-ttl(as.tensor(F[t,,,]),A,c(1,2,3))@data
  X[t,,,]<-S[t,,,]+E[t,,,]
}
result <- TFM_est(X,r,method='PE')
Q.hat<-result$Q
Ft.hat <- result$Ft
```

TFM_FN *Estimation Factor Numbers via Eigenvalue-Ratio Criterion*

Description

This function is to estimate factor numbers via eigenvalue-ratio criterion corresponding to initial estimation without projection, one-step projection estimation, iterative projection estimation and iterative weighted projection estimation by Huber loss.

Usage

```
TFM_FN(x, r = NULL, method = "PE", tol = 1e-04, maxiter = 100)
```

Arguments

`x`	$T \times p_1 \times \cdots \times p_K$ tensor-valued time series.
`r`	input rank of the factor tensor.
`method`	character string, specifying the type of the factor estimation method to be used. `"IE"`, Initial estimation, without projection. `"PE"`, One-step projection estimation. `"iPE"`, Iterative projection estimation. `"HUBER"`, Iterative weighted projection estimation based on huber loss function.
`tol`	tolerance in terms of the Frobenius norm.
`maxiter`	maximum number of iterations if error stays above `tol`.

Details

See Barigozzi et al. (2022) and Barigozzi et al. (2023) for details.

Value

return a list containing the following:

`path` a $K \times (\text{niter} + 1)$ matrix of the estimated Tucker rank of the factor process as a path of the maximum number of iteration (niter) used. The i-th column is the estimated rank $\hat{r}_1, \hat{r}_2, \cdots, \hat{r}_K$ at $(i-1)$-th iteration.

`factor.num` final solution of the estimated Tucker rank of the factor process $\hat{r}_1, \hat{r}_2, \cdots, \hat{r}_K$.

Author(s)

Matteo Barigozzi, Yong He, Lingxiao Li, Lorenzo Trapani.

References

Barigozzi M, He Y, Li L, Trapani L. Robust Estimation of Large Factor Models for Tensor-valued Time Series. <arXiv:2206.09800>

Barigozzi M, He Y, Li L, Trapani L. Statistical Inference for Large-dimensional Tensor Factor Model by Iterative Projection. <arXiv:2303.18163>

Examples

```
library(rTensor)
set.seed(1234)
p <- c(12,16,20) # dimensions of tensor time series
r <- c(3,4,5)  # dimensions of factor series
A<-list()
Q<-list()
for(i in 1:3){
  A[[i]]<-matrix(rnorm(p[i]*r[i],0,1),p[i],r[i])
  Q[[i]]=eigen(A[[i]]%*%t(A[[i]]))$vectors
}
```

```
T<-100
F<-array(NA,c(T,r))
E<-array(NA,c(T,p))
S<-array(NA,c(T,p))
X<-array(NA,c(T,p))
for(t in 1:T){
  F[t,,,]<-array(rnorm(prod(r),0,1),r)
  E[t,,,]<-array(rnorm(prod(p),0,1),p)
  S[t,,,]<-ttl(as.tensor(F[t,,,]),A,c(1,2,3))@data
  X[t,,,]<-S[t,,,]+E[t,,,]
}
rank<-TFM_FN(X,r=NULL,method='PE')
```

Index