

# Package ‘TensorPreAve’

November 8, 2022

**Type** Package

**Title** Rank and Factor Loadings Estimation in Time Series Tensor Factor Models

**Version** 0.1.1

**Author** Weilin Chen [aut, cre]

**Description** A set of functions to estimate rank and factor loadings of time series tensor factor models. A tensor is a multidimensional array. To analyze high-dimensional tensor time series, factor model is a major dimension reduction tool. 'TensorPreAve' provides functions to estimate the rank of core tensors and factor loading spaces of tensor time series. More specifically, a pre-averaging method that accumulates information from tensor fibres is used to estimate the factor loading spaces. The estimated directions corresponding to the strongest factors are then used for projecting the data for a potentially improved re-estimation of the factor loading spaces themselves. A new rank estimation method is also implemented to utilizes correlation information from the projected data.

See Chen and Lam (2022) <[arXiv:2208.04012](https://arxiv.org/abs/2208.04012)> for more details.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**URL** <https://github.com/William-Chenwl/TensorPreAve>

**RoxygenNote** 7.2.1

**Imports** rTensor,MASS,stats,pracma

**Depends** R (>= 2.10)

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Maintainer** Weilin Chen <[w.chen56@lse.ac.uk](mailto:w.chen56@lse.ac.uk)>

**Repository** CRAN

**Date/Publication** 2022-11-08 14:20:21 UTC

## R topics documented:

|                               |   |
|-------------------------------|---|
| bs_cor_rank . . . . .         | 2 |
| equal_weight_tensor . . . . . | 3 |
| iter_proj . . . . .           | 4 |
| pre_est . . . . .             | 5 |
| rank_factors_est . . . . .    | 6 |
| tensor_data_gen . . . . .     | 8 |
| value_weight_tensor . . . . . | 9 |

|              |           |
|--------------|-----------|
| <b>Index</b> | <b>11</b> |
|--------------|-----------|

---

|             |                                   |
|-------------|-----------------------------------|
| bs_cor_rank | <i>Bootstrap Rank Estimation.</i> |
|-------------|-----------------------------------|

---

### Description

Function to estimate the rank of the core tensor by Bootstrapped Correlation Thresholding.

### Usage

```
bs_cor_rank(
  X,
  initial_direction,
  r_range = NULL,
  C_range = seq(0, 100, 0.1)[-1],
  B = 50
)
```

### Arguments

|                   |  |
|-------------------|--|
| X                 | A 'Tensor' object defined in package <b>rTensor</b> with $K + 1$ modes. Mode-1 should correspond to the time mode.   |
| initial_direction | Initial direction for projection, written in a list of $K$ vectors. This can be obtained from the iterative projection procedure.  |
| r_range           | Approximate range of $r_k$ (number of factors) to search from, written in a list of $K$ vectors (e.g. <code>z = list(c(1,10), c(1,10))</code> for $K = 2$ ). Default range is 1 to 10 for all modes. |
| C_range           | The range of constant C for calculating threshold. Default is <code>seq(0,100,0.1)</code> .  |
| B                 | Number of bootstrap samples. Default is 50.  |

### Details

Input a tensor time series and estimated projection directions, return the estimated rank of core tensor.

**Value**

A vector of length  $K$ , indicating estimated number of factors in each mode.

**Examples**

```
# Example of real data set
set.seed(10)
Q_PRE = pre_est(value_weight_tensor)
Q_PROJ = iter_proj(value_weight_tensor, initial_direction = Q_PRE)
bs_rank = bs_cor_rank(value_weight_tensor, Q_PROJ)
bs_rank

# Example using generated data
K = 2
T = 100
d = c(40,40)
r = c(2,2)
re = 10
eta = list(c(0,0),c(0,0))
u = list(c(-2,2),c(-2,2))
set.seed(10)
Data_test = tensor_data_gen(K,T,d,r,re,eta,u)
X = Data_test$X
Q_PRE = pre_est(X)
Q_PROJ = iter_proj(X, initial_direction = Q_PRE)
bs_rank = bs_cor_rank(X, Q_PROJ)
bs_rank
```

---

equal\_weight\_tensor     *Equal weight Fama-French portfolio returns data.*

---

**Description**

Equal weight Fama-French portfolio returns data formed on size and operating profitability of Chen and Lam (2022).

**Format**

A  $576 \times 10 \times 10$  'Tensor' object defined in package **rTensor**, where mode-1,2,3 correspond to time, OP levels and size levels, respectively.

**Details**

Stocks are categorized into 10 different sizes (market equity, using NYSE market equity deciles) and 10 different operating profitability (OP) levels (using NYSE OP deciles. OP is annual revenues minus cost of goods sold, interest expense, and selling, general, and administrative expenses divided by book equity for the last fiscal year end). The stocks in each of the  $10 \times 10$  categories form a

portfolio by equal weight. We use monthly data from July 1973 to June 2021, so that  $T = 576$ , and each data tensor we have thus has size  $10 \times 10 \times 576$ . Since the market factor is certainly pervasive in financial returns, we use the CAPM to remove its effects and facilitate detection of potentially weaker factors.

## References

Chen, W. and Lam, C. (2022). Rank and Factor Loadings Estimation in Time Series Tensor Factor Model by Pre-averaging. Manuscript.

---

|           |  |
|-----------|--|
| iter_proj | <i>Iterative Projection Estimator.</i> |
|-----------|--|

---

## Description

Function for Iterative Projection to re-estimate the factor loading matrices.

## Usage

```
iter_proj(X, initial_direction, proj_N = 30, z = rep(1, length(X@modes) - 1))
```

## Arguments

|                                |   |
|--------------------------------|---|
| <code>X</code>                 | A 'Tensor' object defined in package <b>rTensor</b> with $K + 1$ modes. Mode-1 should correspond to the time mode.  |
| <code>initial_direction</code> | Initial direction for projection, written in a list of $K$ vectors. This can be obtained from the pre-averaging procedure.  |
| <code>proj_N</code>            | Number of iterations, should be a positive integer. Default is 30.  |
| <code>z</code>                 | (Estimated) Rank of the core tensor, written as a vector of length $K$ . Can be set as 1's when we only need to do rank estimation based on projected data. Default is 1's. |

## Details

Input a tensor time series and initial projection direction, return the estimated factor loading matrices using iterative projection.

## Value

A list of  $K$  estimated factor loading matrices.

**Examples**

```

# Example of a real data set
set.seed(10)
Q_PRE = pre_est(value_weight_tensor)
Q_PROJ = iter_proj(value_weight_tensor, initial_direction = Q_PRE)
Q_PROJ

set.seed(10)
Q_PRE = pre_est(value_weight_tensor)
Q_PROJ_2 = iter_proj(value_weight_tensor, initial_direction = Q_PRE, z = c(2,2))
Q_PROJ_2

# Example using generated data
K = 2
T = 100
d = c(40,40)
r = c(2,2)
re = 10
eta = list(c(0,0),c(0,0))
u = list(c(-2,2),c(-2,2))
set.seed(10)
Data_test = tensor_data_gen(K,T,d,r,re,eta,u)
X = Data_test$X
Q_PRE = pre_est(X)
Q_PROJ = iter_proj(X, initial_direction = Q_PRE, z = r)
Q_PROJ

```

pre\_est

*Pre-Averaging Estimator***Description**

Function for the initial Pre-Averaging Procedure.

**Usage**

```
pre_est(X, z = rep(1, length(X@modes) - 1), M0 = 200, M = 5)
```

**Arguments**

|    |   |
|----|---|
| X  | A 'Tensor' object defined in package <b>rTensor</b> with $K + 1$ modes. Mode-1 should correspond to the time mode.  |
| z  | (Estimated) Rank of the core tensor, written as a vector of length $K$ . For iterative projection purpose, we only need this to be 1's. Default is 1's.         |
| M0 | Number of random samples to generate, should be a positive integer. Usually set as 200, or $\min(d_{-k}^2/4, 1000)$ for potential more samples. Default is 200. |
| M  | Number of chosen samples for pre-averaging, should be a positive integer. Usually can be set as constants (5 or 10) or 2.5 percents of M0. Default is 5.        |

**Details**

Input a tensor time series and return the estimated factor loading matrices using pre-averaging method.

**Value**

A list of  $K$  estimated factor loading matrices.

**Examples**

```
# Example of a real data set
set.seed(10)
Q_PRE = pre_est(value_weight_tensor)
Q_PRE

set.seed(10)
Q_PRE_2 = pre_est(value_weight_tensor, z = c(2,2))
Q_PRE_2

# Example using generated data
K = 2
T = 100
d = c(40,40)
r = c(2,2)
re = 10
eta = list(c(0,0),c(0,0))
u = list(c(-2,2),c(-2,2))
set.seed(10)
Data_test = tensor_data_gen(K,T,d,r,re,eta,u)
X = Data_test$X
Q_PRE = pre_est(X, z = r)
Q_PRE
```

---

rank\_factors\_est

*Rank and Factor Loadings Estimation*


---

**Description**

The complete procedure to estimate both rank and factor loading matrices simultaneously.

**Usage**

```
rank_factors_est(
  X,
  proj_N = 30,
  r_range = NULL,
  C_range = seq(0, 100, 0.1)[-1],
```

```

M0 = 200,
M = 5,
B = 50,
input_r = NULL
)

```

### Arguments

|         |  |
|---------|--|
| X       | A 'Tensor' object defined in package <b>rTensor</b> with $K + 1$ modes. Mode-1 should correspond to the time mode.   |
| proj_N  | Number of iterations for iterative projection. Default is 30.  |
| r_range | Approximate range of $r_k$ (number of factors) to search from, written in a list of $K$ vectors (e.g. <code>z = list(c(1, 10), c(1, 10))</code> for $K = 2$ ). Default range is 1 to 10 for all modes. |
| C_range | The range of constant C for calculating threshold. Default is <code>seq(0, 100, 0.1)</code> .  |
| M0      | Number of random samples to generate in pre-averaging procedure. Usually set as 200, or $\min(d_{-k}^2/4, 1000)$ for potential more samples. Default is 200.   |
| M       | Number of chosen samples for pre-averaging. Usually can be set as constants (5 or 10) or 2.5 percents of M0. Default is 5.   |
| B       | Number of bootstrap samples for estimating rank of core tensor by bootstrapped correlation thresholding. Default is 50.  |
| input_r | The rank of core tensor if it is already know, written as a vector of length $K$ . If no input, it will be estimated. Default is NULL.   |

### Details

Input a tensor time series and return the estimated factor loading matrices and rank of core tensor.

### Value

A list containing the following:  
rank: A vector of  $K$  elements, indicating the estimated number of factors in each mode  
loadings: A list of  $K$  estimated factor loading matrices.

### Examples

```

# Example of real data set
set.seed(10)
results = rank_factors_est(value_weight_tensor)
results

```

```

# Example using generated data
K = 2
T = 100
d = c(40, 40)
r = c(2, 2)
re = 10

```

```

eta = list(c(0,0),c(0,0))
u = list(c(-2,2),c(-2,2))
set.seed(10)
Data_test = tensor_data_gen(K,T,d,r,re,eta,u)
X = Data_test$X
results = rank_factors_est(X)
results

```

---

tensor\_data\_gen      *Tensor time series data generation.*

---

### Description

Function to generate a random sample of time series tensor factor model, based on econometrics assumptions.

### Usage

```
tensor_data_gen(K, n, d, r, re, eta, u)
```

### Arguments

|     |   |
|-----|---|
| K   | The number of modes for the tensor time series.   |
| n   | Length of time series.  |
| d   | Dimensions of each mode of the tensor, written in a vector of length K.                                 |
| r   | Rank of core tensors, written in a vector of length K.  |
| re  | Dimension of the cross-sectional common error.  |
| eta | Quantities controlling factor strengths in each factor loading matrix, written in a list of K vectors.  |
| u   | Quantities controlling range of elements in each factor loading matrix, written in a list of K vectors. |

### Details

Input tensor dimension and rank of core tensor, return a sample of tensor time series generated by factor model.

### Value

A list containing the following:

- X: the generated tensor time series, stored in a 'Tensor' object defined in **rTensor**, where mode-1 is the time mode
- A: a list of K factor loading matrices
- F\_ts: time series of core tensor, stored in a 'Tensor' object, where mode-1 is the time mode
- E\_ts: time series of error tensor, stored in a 'Tensor' object, where mode-1 is the time mode



**Examples**

```

set.seed(10)
K = 2
n = 100
d = c(40,40)
r = c(2,2)
re = 10
eta = list(c(0,0),c(0,0))
u = list(c(-2,2),c(-2,2))
Data_test = tensor_data_gen(K,n,d,r,re,eta,u)

X = Data_test$X
A = Data_test$A
F_ts = Data_test$F_ts
E_ts = Data_test$E_ts

X@modes
F_ts@modes
E_ts@modes
dim(A[[1]])

```

---

value\_weight\_tensor     *Value weighted Fama-French portfolio returns data.*

---

**Description**

Value weighted Fama-French portfolio returns data formed on size and operating profitability of Chen and Lam (2022).

**Format**

A  $576 \times 10 \times 10$  'Tensor' object defined in package **rTensor**, where mode-1,2,3 correspond to time, OP levels and size levels, respectively.

**Details**

Stocks are categorized into 10 different sizes (market equity, using NYSE market equity deciles) and 10 different operating profitability (OP) levels (using NYSE OP deciles. OP is annual revenues minus cost of goods sold, interest expense, and selling, general, and administrative expenses divided by book equity for the last fiscal year end). The stocks in each of the  $10 \times 10$  categories form a portfolio using value weighted. We use monthly data from July 1973 to June 2021, so that  $T = 576$ , and each data tensor we have thus has size  $10 \times 10 \times 576$ . Since the market factor is certainly pervasive in financial returns, we use the CAPM to remove its effects and facilitate detection of potentially weaker factors.

**References**

Chen, W. and Lam, C. (2022). Rank and Factor Loadings Estimation in Time Series Tensor Factor Model by Pre-averaging. Manuscript.

# Index

`bs_cor_rank`, [2](#)

`equal_weight_tensor`, [3](#)

`iter_proj`, [4](#)

`pre_est`, [5](#)

`rank_factors_est`, [6](#)

`tensor_data_gen`, [8](#)

`value_weight_tensor`, [9](#)