Package ‘MEFM’

June 6, 2024

Type Package

Title Perform MEFM Estimation on Matrix Time Series

Version 0.1.1

Date 2024-06-04

Author Zetai Cen [aut, cre]

Maintainer Zetai Cen <z.cen@lse.ac.uk>

Description To perform main effect matrix factor model (MEFM) estimation for a given matrix time series as described in Lam and Cen (2024) <doi:10.48550/arXiv.2406.00128>. Estimation of traditional matrix factor models is also supported. Supplementary functions for testing MEFM over factor models are included.

License GPL-3

Imports tensorMiss, stats

Encoding UTF-8

RoxygenNote 7.2.3

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2024-06-06 17:10:09 UTC

Contents

est_FM *Estimation of factor models on matrix time series*

Description

Estimate the FM structure on the given matrix time series

Usage

```
est_FM(Yt, r = 0, delta = 0.2)
```

Arguments

Yt	demeaned matrix time series, written in an array with dimension 3 and the first dimension for time.
r	Rank of core factors for the common component, written in a vector of length 2. First value as 0 is to denote unknown rank which would be automatically estimated using ratio-based estimators. Default is 0.
delta	Non-negative number as the correction parameter for rank estimation. Default is 0.2.

Value

A list containing the following: r: a vector representing either the given rank or the estimated rank, with length 2; A: a list of the estimated row and column factor loading matrices; Ft: the estimated core factor series, as multi-dimensional array with dimension 3, where mode-1 is the time mode; Ct: the estimated common component time series, as multi-dimensional array with dimension 3, where mode-1 is the time mode; covMatrix: a list of the estimated row and column covariance matrices which are used to estimate loading matrices;

Examples

```
TT = 40;
d = c(40,40);
r = c(2,2);
re = c(2,2);
eta = list(c(0,0), c(0,0));
coef_f = c(0.7, 0.3, -0.4, 0.2, -0.1);
coef_fe = c(-0.7, -0.3, -0.4, 0.2, 0.1);
coef_e = c(0.8, 0.4, -0.4, 0.2, -0.1);
param_mu = c(0,1);
param_alpha = c(0,1);
param_beta = c(0,1);
data_example = gen_MEFM(TT,d,r,re,eta, coef_f, coef_fe, coef_e, param_mu, param_alpha, param_beta);
est_FM(data_example$FM);
```

est_MEFM *Estimation of MEFM on matrix time series*

Description

Estimate the MEFM structure on the given matrix time series

Usage

```
est_MEFM(Yt, r = 0, delta = 0.2)
```

Arguments

`Yt`	demeaned matrix time series, written in an array with dimension 3 and the first dimension for time.
`r`	Rank of core factors for the common component, written in a vector of length 2. First value as 0 is to denote unknown rank which would be automatically estimated using ratio-based estimators. Default is 0.
`delta`	Non-negative number as the correction parameter for rank estimation. Default is 0.2.

Value

A list containing the following: r: a vector representing either the given rank or the estimated rank, with length 2; mu: a vector representing the estimated time-varying grand mean series; alpha: a matrix representing the estimated time-varying row effect series, where the row index denotes time index; beta: a matrix representing the estimated time-varying column effect series, where the row index denotes time index; A: a list of the estimated row and column factor loading matrices; Ft: the estimated core factor series, as multi-dimensional array with dimension 3, where mode-1 is the time mode; Ct: the estimated common component time series, as multi-dimensional array with dimension 3, where mode-1 is the time mode; Yt: the estimated matrix time series, as multi-dimensional array with dimension 3, where mode-1 is the time mode; covMatrix: a list of the estimated row and column covariance matrices which are used to estimate loading matrices;

Examples

```
TT = 40;
d = c(40,40);
r = c(2,2);
re = c(2,2);
eta = list(c(0,0), c(0,0));
coef_f = c(0.7, 0.3, -0.4, 0.2, -0.1);
coef_fe = c(-0.7, -0.3, -0.4, 0.2, 0.1);
coef_e = c(0.8, 0.4, -0.4, 0.2, -0.1);
param_mu = c(0,1);
param_alpha = c(0,1);
param_beta = c(0,1);
data_example = gen_MEFM(TT,d,r,re,eta, coef_f, coef_fe, coef_e, param_mu, param_alpha, param_beta);
```

```
est_MEFM(data_example$MEFM);
```

---

gen_MEFM *Data generation of matrix time series with MEFM structure*

Description

Generate a matrix time series with MEFM at each time t, with the first mode as the time mode, the second as the row mode and the third as the column mode

Usage

```
gen_MEFM(
  TT,
  d,
  r,
  re,
  eta,
  coef_f,
  coef_fe,
  coef_e,
  param_mu,
  param_alpha,
  param_beta,
  heavy_tailed = FALSE,
  t_df = 3,
  rademacher = FALSE,
  seed = 2024
)
```

Arguments

TT	Length of time series.
d	Dimensions of the matrix at time t, written in a vector of length 2 where the first number denotes the number of rows p and the second denoted the number of columns q.
r	Rank of the core factors, written in a vector of length 2.
re	re: Rank of the cross-sectional common error core factors, written in a vector of length 2.
eta	Quantities controlling factor strengths in each factor loading matrix, written in a list of 2 vectors.
coef_f	AR(5) coefficients for the factor series, written in a vector of length 5.
coef_fe	AR(5) coefficients for the common component in error series, written in a vector of length 5.

`coef_e` AR(5) coefficients for the idiosyncratic component in error series, written in a vector of length 5.

`param_mu` If rademacher = TRUE, represent parameters of normal distribution to generate grand mean series mu_t, written in a vector of length 2 representing the mean and standard deviation. Otherwise written in a scalar multiplied by the generated Rademacher random variable.

`param_alpha` If rademacher = TRUE, represent parameters of normal distribution to generate row effect series alpha_t, written in a vector of length 2 representing the mean and standard deviation. Otherwise written in a scalar multiplied by the generated Rademacher random variable.

`param_beta` If rademacher = TRUE, represent parameters of normal distribution to generate column effect series beta_t, written in a vector of length 2 representing the mean and standard deviation. Otherwise written in a scalar multiplied by the generated Rademacher random variable.

`heavy_tailed` Whether to generate data from heavy-tailed distribution. If FALSE, generate from N(0,1); if TRUE, generate from t-distribution. Default is FALSE.

`t_df` The degree of freedom for t-distribution if heavy_tailed = TRUE. Default is 3.

`rademacher` Mechanism to generate mu, alpha and beta. If FALSE, generate from normal with param_mu, param_alpha and param_beta; if TRUE, generate from Rademacher distribution and scaled by param_mu, param_alpha and param_beta. Default is FALSE.

`seed` Random seed required for reproducibility. Default is 2024.

Value

A list containing the following: mu: the generated time-varying grand mean series, as a vector of length TT; alpha: the generated time-varying row effect series, as a matrix of dimension (TT,p); beta: the generated time-varying column effect series, as a matrix of dimension (TT,q); A: a list of 2 factor loading matrices; C: the generated common component time series, as multi-dimensional array with dimension 3, where mode-1 is the time mode, mode-2 is for rows and mode-3 is for columns; Ft: the generated core factor series, as multi-dimensional array with dimension 3, where mode-1 is the time mode, mode-2 is for rows and mode-3 is for columns; MEFM: the generated matrix time series with MEFM structure, as multi-dimensional array with dimension 3, where mode-1 is the time mode, mode-2 is for rows and mode-3 is for columns; FM: the generated matrix time series with only traditional factor structure, as multi-dimensional array with dimension 3, where mode-1 is the time mode, mode-2 is for rows and mode-3 is for columns; E: the generated error time series with factor structure, as multi-dimensional array with dimension 3, where mode-1 is the time mode, mode-2 is for rows and mode-3 is for columns;

Examples

```
TT = 40;
d = c(40,40);
r = c(2,2);
re = c(2,2);
eta = list(c(0,0), c(0,0));
coef_f = c(0.7, 0.3, -0.4, 0.2, -0.1);
```

```
coef_fe = c(-0.7, -0.3, -0.4, 0.2, 0.1);
coef_e = c(0.8, 0.4, -0.4, 0.2, -0.1);
param_mu = c(0,1);
param_alpha = c(0,1);
param_beta = c(0,1);
gen_MEFM(TT,d,r,re,eta, coef_f, coef_fe, coef_e, param_mu, param_alpha, param_beta);
```

---

make_gamma *Aggregation of estimated error*

Description

Computing the aggregated estimated error at some index for constructing asymptotic normality

Usage

```
make_gamma(E, type = "mu", ind = 1)
```

Arguments

E	A matrix representing the estimated error matrix at some time t of dim (p,q).
type	Character input, choice from one of 'mu', 'alpha' and 'beta'. Default is 'mu'.
ind	integer denoting the index of interest, only used when type is 'alpha' or 'beta'. Default is 1.

Value

A numeric number

---

make_xy *Construction of series for testing MEFM*

Description

Constructing x or y series for the MEFM testing

Usage

```
make_xy(E, type = "alpha")
```

Arguments

E	An array representing the sequence of estimated error matrix of dim (T,p,q).
type	Character input, either 'alpha' or 'beta'. Default is 'alpha'.

Value

A vector representing the constructed x or y series

`qHat` *Estimated theta quantile based on a given series*

Description

Computing the estimated quantile according to a given series for a given level of theta

Usage

```
qHat(xt, theta = 0.95)
```

Arguments

`xt`	A vector representing a series.
`theta`	A value from 0 to 1. Default is 0.95.

Value

A numeric number

`sigmaD_MEFM` *HAC covariance estimator for asymptotic normality on each row j of loading matrix estimator*

Description

Computing the HAC covariance estimator for asymptotic normality on each row j of the row or column loading matrix estimator

Usage

```
sigmaD_MEFM(k, D, Q, C, E, j, beta = 0)
```

Arguments

k	Integer to choose the mode of loading matrix, either 1 or 2.
D	Eigenvalue matrix of sample covariance matrix, with dimension rk by rk.
Q	Estimated row (k=1) or column (k=2) loading matrix, with dimension p (for k=1) or q (for k=2) by rk.
C	Estimated common component series, written in an array with dimension (T,p,q) where the first dimension denotes time.
E	Estimated error matrix time series, written in an array with the same dimension as C.
j	Integer representing the row of loading matrix. Value should be integers from minimum 1 to maximum p (for k=1) or q (for k=2).
beta	Lag parameter of the HAC type. Default is 0.

Value

A matrix of dimension rk by rk

Examples

```
TT = 40;
d = c(40,40);
r = c(2,2);
re = c(2,2);
eta = list(c(0,0), c(0,0));
coef_f = c(0.7, 0.3, -0.4, 0.2, -0.1);
coef_fe = c(-0.7, -0.3, -0.4, 0.2, 0.1);
coef_e = c(0.8, 0.4, -0.4, 0.2, -0.1);
param_mu = c(0,1);
param_alpha = c(0,1);
param_beta = c(0,1);
data_example = gen_MEFM(TT,d,r,re,eta, coef_f, coef_fe, coef_e, param_mu, param_alpha, param_beta);
est_result = est_MEFM(data_example$MEFM, r=r);
D2 <- diag(x=(svd(est_result$covMatrix[[2]])$d)[1:r[2]], nrow=r[2], ncol=r[2]);
sigmaD_MEFM(2, D2, est_result$A[[2]], est_result$Ct, data_example$MEFM - est_result$Yt, 1, 0);
```

Index