Package 'GFPCA'

December 26, 2023

Title Graphical Functional Principal Component Analysis

Version 1.0

Author Jianbin Tan <jianbin.tan@duke.edu>

Description With the development of data collection methods, multivariate functional data that possess complex temporal correlation structures have become increasingly available. To analyze these data, we need to account for both multivariate and temporal correlations. These two-way dependencies may lead to inefficient dimension reduction when using classical FPCA methods. To solve this issue, we propose a novel graphical FPCA (GFPCA) for the dimension reduction of multivariate functional time series (MFTS) data, in which the multivariate dependencies are characterized by graphical models. In this package, the GFPCA() function is used for optimal reconstructions of signals from contaminated MFTS data using graphical-level information. In addition, the function can also be used for capturing and encoding the graphical structures of MFTS.

License MIT

Encoding UTF-8

RoxygenNote 7.2.3

R topics documented:

eigen_func_dyn *Estimating Functional Filters*

Description

A function to estimate functional filters for dynamic FPCA.

Usage

```
eigen_func_dyn(sp_matrix, grid_freq, sel_eig, max_comp)
```

Arguments

sp_matrix	An array of the estimated coefficients for spectral density kernels.
grid_freq	The frequencies of the estimated spectral density kernels.
sel_eig	The fraction of variance explained.
max_comp	The maximum number of components.

Value

Estimations of the number of components and the coefficients of functional filters.

eigen_func_sta *Estimating Eigenfunctions*

Description

A function to estimate eigenfunctions for static FPCA.

Usage

```
eigen_func_sta(cov_matrix_0, sel_eig, max_comp)
```

Arguments

cov_matrix_0	The estimated coefficients for lag-zero covariance functions.
sel_eig	The fraction of variance explained.
max_comp	The maximum number of components.

Value

Estimations of the number of components and the coefficients of eigenfunctions.

eigen_matrix_con *Estimating Eigen-matrices*

Description

A function to estimate eigen-matrices.

Usage

```
eigen_matrix_con(
  time_length,
  comp_num,
  subject_length,
  dmean_smo_fda,
  q,
  r,
  cov_matrix,
  Freq_eig,
  time_length_eig
)
```

Arguments

time_length	The time length of MFTS.
comp_num	The number of components.
subject_length	The dimension of MFTS.
dmean_smo_fda	The pre-smoothed coefficients after removing mean trends.
cov_matrix	An array of the estimated coefficients for the cross-covariance functions.
Freq_eig	The frequencies for the estimated eigenmatrices.
time_length_eig	The number of the evaluated frequencies.
p, r	The time lags for the estimated covariance functions.

Value

An array of the estimated eigenmatrices.

eigen_pre *Estimating Cross-covariance*

Description

A function to estimate cross-covariance for different time lags.

Usage

```
eigen_pre(subject_length, time_length, dmean_smo_fda, q, r)
```

Arguments

subject_length	The dimension of MFTS.
time_length	The time length of MFTS.
dmean_smo_fda	The pre-smoothed coefficients after removing mean trends.
p, r	The time lags for the estimated covariance functions.

Value

An array of the estimated coefficients for the cross-covariance functions.

GFPCA *Graphical Functional Principal Component Analysis (GFPCA)*

Description

A function to implement graphical FPCAs.

Usage

```
GFPCA(MFTS, Lt, Dynamic, Max.comp, FVE, Mean.zero)
```

Arguments

MFTS	A Z*p*J array containing the regularly observed MFTS data, where Z is the number of observed time points, p is the dimension of MFTS, and J is the time length of MFTS.
Lt	A vector containing the observed time points of MFTS. We require the time points to be contained in the interval [0,1].
Dynamic	A logical evaluating to TRUE or FALSE indicating whether the graphical DFPCA or SFPCA should be conducted.
Max.comp	The maximum number of components.
FVE	A numeric in [0,1] indicating the fraction of variance explained for determining the number of components. If FVE = 0, the number of component is selected by the ratio of variance explained.
Mean.zero	A logical evaluating to TRUE or FALSE indicating whether the mean functions of MFTS are zero.

Details

This is a generic function to implement graphical versions of DFPCA or SFPCA for multivariate functional time series (MFTS). The function requires that the MFTS data are densely and regularly observed.

Value

A list with components

`comp_num`	Number of components.
`xi_dyn_IN` or `xi_sta_IN`	A list of FPC scores computed by integration.
`xi_dyn_CE` or `xi_sta_CE`	A list of FPC scores computed by conditional expectation.
`mean_function`	A list of mean functions of MFTS, where Lt contains the time grid of the estimated values.
`Functional filters` or `eigenfunctions`	A list of the estimated functional filters (or eigenfunctions), where Lt contains the time grid of the estimated values.
`eigen_matrix`	A p*p*(J/2)*comp_num array containing the estimated eigenmatrices for different frequences.
`Phi`	A p*p*(J/2)*comp_num array containing the estimated inverse of eigenmatrices by incorporating graph constraints.
`mea_error`	A vector containing the estimated variances of the measurement errors.
`dmean_smo_fda`	The pre-smoothed coefficients after removing mean trends.

---

gra_selection *Regularized Estimator for Precision Matrices*

Description

Joint regularized estimator for precision matrices by ADMM.

Usage

```
gra_selection(S, lambda)
```

Arguments

`S`	An array of eigenmatrices.
`lambda`	A tuning parameter to the joint regularized estimator.

Value

A list of two collections of precision matrices.

score_est *Score Extraction*

Description

A function to extract FPCA scores for Graphical FPCA.

Usage

```
score_est(
  xi,
  fre_tra,
  des_mat,
  dmean_fda,
  eigen_matrix,
  squ_des_mat,
  squ_fre_tra,
  sigma_est,
  subject_length,
  time_length,
  comp_num
)
```

Arguments

xi	An initial value of the scores.
fre_tra, des_mat, dmean_fda, squ_des_mat, squ_fre_tra	Some auxiliary objects for the score extraction.
eigen_matrix	An array of eigenmatrices.
sigma_est	A vector of the estimated variances for measurement errors.
subject_length	The dimension of MFTS.
time_length	The time length of MFTS.
comp_num	The number of components.

Value

A list containing the estimated scores and other objects relating to the convergence of the algorithm.

Sel_lam *Selection of Tuning Parameters for Joint Graphical Lasso*

Description

A function to select tuning parameters for joint graphical Lasso via AIC.

Usage

```
Sel_lam(eigen_matrix)
```

Arguments

eigen_matrix	An array of eigenmatrices.

Value

A selected tuning parameter.

---

smooth_fda *Pre-smoothing of Multivariate Functional Time Series (MFTS)*

Description

A function to pre-smooth MFTS data.

Usage

```
smooth_fda(fda, x_fda, subject_length, time_length, X_basis, tra_mat)
```

Arguments

fda	An array containing the observed MFTS data.
x_fda	The observed time points of MFTS.
subject_length	The dimension of MFTS.
time_length	The time length of MFTS.
X_basis	A basis object for pre-smoothing.
tra_mat	A transformation matrix for the basis functions.

Value

A list with the pre-smoothed coefficients and the estimated variances of measurement errors.

---

sp_tran *Constructing Spectral Density Kernels*

Description

A function to construct spectral densities given cross-covariance.

Usage

```
sp_tran(r, grid_freq, cov_matrix)
```

Arguments

r	The time lags for estimating the spectral density kernels.
grid_freq	The frequencies of the estimated spectral density kernels.
cov_matrix	An array of the estimated coefficients for the cross-covariance functions.

Value

An array of the estimated coefficients for spectral density kernels.

`test_weaksep` *Testing Dynamic Weak Separability*

Description

A function to illustrate the validity of dynamic weak separability.

Usage

```
test_weaksep(
  time_length,
  comp_num,
  subject_length,
  dmean_smo_fda,
  q,
  r,
  Freq_eig,
  time_length_eig
)
```

Arguments

`time_length`	The time length of MFTS.
`comp_num`	The number of components.
`subject_length`	The dimension of MFTS.
`dmean_smo_fda`	The pre-smoothed coefficients after removing mean trends.
`Freq_eig`	The frequencies for the estimated eigenmatrices.
`time_length_eig`	The number of the evaluated frequencies.
`p, r`	The time lags for the estimated covariance functions.

Value

An object to illustrate the validity of dynamic weak separability.

Index