Package 'mGSFPCA'

May 8, 2026

Title Estimate Functional Principal Components from Sparse Data

Version 0.2.2

Description Implements functional principal component analysis (FPCA) for univariate and multivariate sparse functional data. The package estimates eigenfunctions, eigenvalues, and error variance simultaneously via maximum likelihood estimation (MLE), using a spline basis representation of the eigenfunctions. Orthonormality of the estimated eigenfunctions is enforced through a modified Gram-Schmidt (MGS) orthogonalization procedure applied iteratively during estimation, avoiding direct optimization over the Stiefel manifold and improving numerical stability. The optimal number of basis functions and principal components is selected via an Akaike Information Criterion (AIC)-type criterion, supporting both a full grid-search strategy and a computationally efficient sequential selection approach. Principal component scores are estimated by conditional expectation, enabling reconstruction of individual trajectories over the entire domain from sparse observations. Pointwise confidence intervals for reconstructed trajectories are also provided. Methods are described in Mbaka, Cao and Carey (2026) <doi:10.48550/arXiv.2603.18833> and Mbaka and Carey (2026) <doi:10.48550/arXiv.2603.19799>.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 2.10)

LazyData true

Imports fda, pracma, Rcpp, Metrics

LinkingTo Rcpp, RcppEigen

NeedsCompilation yes

Author Uche Mbaka [aut, cre] (ORCID: <https://orcid.org/0000-0002-1427-6388>), Michelle Carey [ctb] (ORCID: <https://orcid.org/0000-0002-5603-4264>)

Maintainer Uche Mbaka <uche.mbaka@ucd.ie>

Repository CRAN

Date/Publication 2026-05-08 14:30:02 UTC

Contents

`bspline_sim` *Simulated B-Spline Functional Data*

Description

A simulated dataset containing true and observed functional data based on B-spline functions

Format

A list with 6 components:

X A numeric matrix of dimension 100 x 51, representing the true functional data for 100 subjects evaluated at 51 time points.

Y A numeric matrix of dimension 617 x 3, representing observed data with the following columns:

ID Integer, subject identifiers (1 to 100).

time Numeric, observation time points in the range (0, 1).

measurement Numeric, observed values at the corresponding time points.

eigFun A numeric matrix of dimension 5 x 51, representing the true eigenfunctions (principal components) for 5 components evaluated on a 51-point grid.

eigVal A numeric vector of length 5, containing the true eigenvalues associated with the eigenfunctions, in descending order.

nbasis An integer, the true number of B-spline basis functions used to generate the data, equal to 10.

sig A numeric value, the true error variance (sigma^2) of the observation noise, equal to 0.167.

Source

Simulated data based on: Peng, J. and Paul, D. (2009). A geometric approach to maximum likelihood estimation of the functional principal components from sparse longitudinal data. Journal of Computational and Graphical Statistics. http://www.tandfonline.com/doi/abs/10.1198/jcgs.2009.08011

Examples

```
# example code
 bspline_sim
```

eval_mGSFPCA *Evaluate Principal Component Functions from mGSFPCA*

Description

Evaluates the principal component functions (eigenfunctions) from mGSFPCA() object at specified time points.

Usage

```
eval_mGSFPCA(mGSFPCA_obj, eval_pts = NULL, matrix_orth = FALSE)
```

Arguments

mGSFPCA_obj	A list object returned by the mGSFPCA function, containing the estimated principal components, basis functions, and other model parameters.
eval_pts	Numeric vector specifying the time points at which to evaluate the principal component functions. The time points should either be in the original data range or $\in [0, 1]$, in which case the output is equivalent to calling fda::eval.fd on mGSFPCA_obj$eigenfunction.
matrix_orth	Logical indicating whether to enforce strict orthogonality of the evaluated eigenfunctions $\Phi^\top \Phi = I_p$ under standard matrix multiplication (discrete inner product). Diagnostic comparison plots are displayed. Requires length(eval_pts) > p.

Details

The eval_mGSFPCA function evaluates the principal component functions (eigenfunctions) from an mGSFPCA object at the provided eval_pts.

Value

A matrix of dimension length(eval_pts) x p, where p is the number of principal components, containing the evaluated principal component functions at the specified eval_pts.

eval_spMultFPCA *Evaluate Principal Component Functions from spMultFPCA*

Description

Evaluates the principal component functions (eigenfunctions) from spMultFPCA() object at specified time points (eval_pts).

Usage

```
eval_spMultFPCA(spMultFPCA_obj, eval_pts = NULL)
```

Arguments

spMultFPCA_obj	A list object returned by the spMultFPCA function, containing the estimated principal components, basis functions, and other model parameters.
eval_pts	Numeric vector specifying the time points at which to evaluate the principal component functions.

Details

The eval_spMultFPCA function evaluates the principal component functions (eigenfunctions) from an spMultFPCA object at the provided eval_pts.

Value

A list containing the following components:

- Ckk: A list containing the eigenfunctions, scores, and curve prediction for each variable at eval_pts.
- scores: A matrix of multivariate principal component scores for each subject.
- eigFunctions: A list of matrices, each containing the multivariate eigenfunctions for a variable.
- values: A vector of multivariate eigenvalues.
- vectors: A matrix representing the eigenvectors associated with the combined univariate score vectors
- npc: The number of multivariate principal components retained.
- Cov: The estimated multivariate covariance matrix.
- fullEigFun: A matrix of concatenated multivariate eigenfunctions across all variables.

---

get_xHat *Compute Fitted Trajectories and Scores from mGSFPCA*

Description

Estimates principal component scores and predicts trajectories for each subject using the results from mGSFPCA().

Usage

```
get_xHat(mGSFPCA_obj, eval_pts = NULL, alpha = 0.05)
```

Arguments

mGSFPCA_obj	A list object returned by the mGSFPCA function, containing the estimated principal components, mean function, and other model parameters.
eval_pts	Numeric vector specifying the time points at which to evaluate fitted trajectories. If NULL (default), the evaluation grid from the mGSFPCA_obj$pars$evalGrid is used.
alpha	Numeric value between 0 and 1 specifying the significance level for constructing point-wise confidence bands for the predicted trajectories. Default is 0.05.

Details

The get_xHat function computes the fitted trajectories and principal component scores for each subject based on the output of the mGSFPCA function. It uses the PACE (Principal Analysis by Conditional Expectation) method to estimate the scores.

Value

A list containing the following components:

- xHat: A matrix of dimension n x length(eval_pts), where n is the number of subjects, containing the fitted trajectories for each subject evaluated at eval_pts.
- Xi: A matrix of dimension n x p, where p is the number of principal components, containing the principal component scores for each subject.
- xHat_CI: A matrix of dimension n x length(eval_pts), containing the (1-α) point-wise confidence intervals for each subjects at eval_pts.

References

Yao, F., Müller, H.-G., & Wang, J.-L. (2005). Functional data analysis for sparse longitudinal data. Journal of the American Statistical Association, 100(470), 577–590.

---

mGSFPCA *Estimate Functional Principal Components from Sparse Data*

Description

Functional principal component analysis with modified Gram-Schmidt Orthornormalization and MLE.

Usage

```
mGSFPCA(
  data,
  p = 2:5,
  k = c(5, 10, 15),
  basis_type = "bspline",
```

```
  maxit = 500,
  optim_tol = 1e-05,
  optim_trace = 0,
  nRegGrid = 51,
  init_coeff = NULL,
  obs_range = NULL,
  mu_nbasis = 15,
  bin_size = 101,
  use_kp_grid = TRUE,
  alpha = 0,
  skip_check = FALSE
)
```

Arguments

data	A matrix or data frame with three columns: ID (subject ID), time (observation time points), and value (observed values).
p	Integer vector specifying the candidate number of principal components to consider. Default is 2:5.
k	Integer vector specifying the candidate number of basis functions to consider. Default is c(5, 10, 15).
basis_type	Character string specifying the type of basis functions to use. Options are "bspline" (default) or "fourier".
maxit	Integer specifying the maximum number of iterations for the optimization algorithm. Default is 500.
optim_tol	Numeric specifying the relative tolerance for the optimization convergence (Based on optim()). Default is 1e-5.
optim_trace	Integer specifying the level of tracing information from the optimization algorithm (Based on optim()). Default is 0 (no tracing).
nRegGrid	Integer specifying the number of points in the regular grid for evaluation. Default is 51.
init_coeff	Numeric vector of initial coefficients, or "LSQ". If "LSQ", initialization is based on a least-squares estimate of the covariance structure. If NULL (default), coefficients are initialized randomly.
obs_range	Numeric vector of length 2 specifying the observation range (aT, bT). If NULL (default), it is set to c(0,1).
mu_nbasis	Integer specifying the number of basis functions for the mean function. Default is 15.
bin_size	Integer specifying the number of bins for data binning. Default is 101.
use_kp_grid	Logical indicating whether to evaluate all combinations of p and k (TRUE, default) or use a stepwise model selection approach (FALSE).
alpha	Numeric value $\in [0, 0.5]$ controlling density-based weighting of the binned observations. Default is 0.
skip_check	Logical indicating whether to skip input validation checks. Default is FALSE.

Details

The mGSFPCA function implements functional principal component analysis using MLE with modified Gram-Schmidt orthornormalization.

Value

A list containing the following components:

- `Phi`: Matrix of estimated eigenfunctions.
- `Lambda`: Vector of estimated eigenvalues.
- `eigenfunctions`: List of eigenfunctions on $[0, 1]$ as fd object.
- `mu`: Vector of the estimated mean function evaluated on the grid.
- `sig2`: Estimated variance of the error term.
- `pars`: A list containing model parameters and results:
  - `p`: Optimal number of principal components.
  - `k`: Optimal number of basis functions.
  - `AIC`: Table of AIC values for different p and k combinations.
  - `coeffs`: Optimized coefficients.
  - `c_tilde`: Orthonormal coefficients.
  - `orthB`: Orthonormal basis functions evaluated over nRegGrid.
  - `mu_fdobj`: Functional data object for the mean function.
  - `mu_basis`: Basis object for the mean function.
  - `workGrid`: Grid points used for estimation.
  - `evalGrid`: Evaluation grid points.
  - `eigBasis`: Basis object for the eigenfunctions.
  - `binData`: Binned data used for estimation.
  - `range`: Observation range.
  - `convergence`: Convergence status of the optimization (Based on optim()). 0 indicates successful completion.

References

Mbaka, U., Cao, J., & Carey M., (2026). Estimation of Functional Principal Components from Sparse Functional Data.

Peng, J. and Paul, D. (2009). A geometric approach to maximum likelihood estimation of the functional principal components from sparse longitudinal data. Journal of Computational and Graphical Statistics.

Yao, F., Müller, H.-G., & Wang, J.-L. (2005). Functional data analysis for sparse longitudinal data. Journal of the American Statistical Association, 100(470), 577–590.

Examples

```
# Plot 3 random subjects
set.seed(111)
samp <- sample.int(100, 3)
# bspline_sim is automatically loaded
Y <- bspline_sim$Y; X <- bspline_sim$X
id <- unique(Y[,1])
grid <- seq(0, 1, length = 51)

plot(Y[,2], Y[,3], type = 'n', ylab = 'Y(t)', xlab = 't')
for (i in 1:3) {
  points(Y[id == samp[i], 2], Y[id == samp[i], 3], col = i, pch = 19)
  lines(grid, X[samp[i], ], col = i, lwd = 2)
}

# Estimate
fit <- mGSFPCA(Y, p = 3:5, k = c(5, 10, 15), basis_type = "bspline",
nRegGrid = 51, bin_size = 101)

c(fit$pars$p, fit$pars$k)

fit$Lambda

plot(grid, fit$Phi[,1], type ='l', lty = 2)
lines(grid, bspline_sim$eigFun[1,], col = 2) # change sign if needed
min(Metrics::rmse(fit$Phi[,1], bspline_sim$eigFun[1,]),
 Metrics::rmse(-fit$Phi[,1], bspline_sim$eigFun[1,]))

x_pred <- get_xHat(fit)
Metrics::rmse(x_pred$xHat, bspline_sim$X)
```

`spMultFPCA` *Sparse Multivariate Functional Principal Component Analysis*

Description

Performs sparse multivariate functional principal component analysis (spMultFPCA) on a list of functional datasets, estimating multivariate principal components using the modified Gram-Schmidt Functional Principal Component Analysis (mGSFPCA) for each variable and combining results via singular value decomposition (SVD).

Usage

```
spMultFPCA(
  dataCell,
  r = NULL,
```

```
  G = NULL,
  basis_type = "bspline",
  nRegGrid = 51,
  grid_range = NULL,
  mu_nbasis = 15,
  M_npc = NULL,
  maxit = 500,
  optim_tol = 1e-05,
  optim_trace = 0,
  bin_size = 51
)
```

Arguments

dataCell	A list of matrices or data frames, each with three columns: ID (subject identifier), time (observation time points), and value (observed values) for each functional variable.
r	A list of integer vectors specifying candidate numbers of principal components for each variable. If NULL, defaults to 3:5 for each variable.
G	A list of integer vectors specifying candidate numbers of basis functions for each variable. If NULL, defaults to c(5, 10, 15) for each variable.
basis_type	Character string or vector specifying the type of basis functions ("bspline" or "fourier") for each variable. If a single string, it is applied to all variables. Defaults to "bspline".
nRegGrid	Integer or list of integers specifying the number of points in the regular grid for evaluation for each variable. If a single integer, it is applied to all variables. If NULL, defaults to 50 for each variable.
grid_range	A list of numeric vectors of length 2 specifying the observation range [aT, bT] for each variable. If a single range is provided, it is applied to all variables. If NULL, ranges are determined from the data.
mu_nbasis	Integer or list of integers specifying the number of basis functions for the mean function of each variable. If a single integer, it is applied to all variables. If NULL, defaults to 15 for each variable.
M_npc	Integer specifying the number of multivariate principal components to retain. If NULL, determined automatically using the elbow method.
maxit	Integer specifying the maximum number of iterations for the optimization algorithm. Default is 500.
optim_tol	Numeric specifying the tolerance for optimization convergence. Default is 1e-5.
optim_trace	Integer specifying the level of tracing information from the optimization algorithm. Default is 0 (no tracing).
bin_size	Integer or list of integers specifying the number of bins for data binning for each variable. If a single integer, it is applied to all variables. If NULL, defaults to 51 for each variable.

Details

The `spMultFPCA` function performs multivariate functional principal component analysis on sparse functional data by applying `mGSFPCA` to each variable in `dataCell`.

Value

A list containing the following components:

- `Ckk`: A list of results from `mGSFPCA` for each variable.
- `scores`: A matrix of multivariate principal component scores for each subject.
- `eigFunctions`: A list of matrices, each containing the multivariate eigenfunctions for a variable.
- `values`: A vector of multivariate eigenvalues.
- `vectors`: A matrix representing the eigenvectors associated with the combined univariate score vectors
- `npc`: The number of multivariate principal components retained.
- `Cov`: The estimated multivariate covariance matrix.
- `fullEigFun`: A matrix of concatenated multivariate eigenfunctions across all variables.

References

Mbaka, U., & Carey M. (2026). Estimation of Multivariate Functional Principal Components from Sparse Functional Data

Happ, C., & Greven, S. (2018). Multivariate functional principal component analysis for data observed on different dimensional domains. Journal of the American Statistical Association, 113(522), 649–659.

Examples

```
sp_mult_sim
tmp <- spMultFPCA(sp_mult_sim$obs_data,
                  r = rep(list(3:5), 3),
                  G = rep(list(5:10), 3),
                  nRegGrid = 100)
Metrics::rmse(tmp$Cov, sp_mult_sim$True_Cov)
grid <- seq(0, 1, length = 300)
plot(grid, sp_mult_sim$True_Eigs$funcs[,1], type ='l', ylab = 'phi')
lines(grid, -tmp$fullEigFun[,1], col = 2)
```

`sp_mult_sim` *Simulated Sparse Multivariate Functional Data*

Description

A simulated dataset for testing sparse multivariate functional principal component analysis (e.g., `spMultFPCA`). It contains observed data for three functional variables, true covariance, true trajectories, eigenfunctions, eigenvalues, scores, and error variance for 100 subjects evaluated over a 100-point grids.

Format

A list with 6 components:

obs_data A list of 3 data frames, each representing observed data for one functional variable:

- **y1** A data frame with 482 observations and 3 variables:
  - **subj** Integer, subject identifiers (1 to 100).
  - **argvals** Numeric, observation time points in the range (0, 1).
  - **y** Numeric, observed values for the first functional variable.
- **y2** A data frame with 489 observations and 3 variables:
  - **subj** Integer, subject identifiers (1 to 100).
  - **argvals** Numeric, observation time points in the range (0, 1).
  - **y** Numeric, observed values for the second functional variable.
- **y3** A data frame with 510 observations and 3 variables:
  - **subj** Integer, subject identifiers (1 to 100).
  - **argvals** Numeric, observation time points in the range (0, 1).
  - **y** Numeric, observed values for the third functional variable.

True_Cov A numeric matrix of dimension 300 x 300, representing the true multivariate covariance matrix across all variables evaluated on a 300-point grid.

True_obs A list of 3 numeric matrices, each of dimension 100 x 100, representing the true trajectories for 100 subjects for each of the three functional variables.

True_Eigs A list of 2 components:

- **funcs** A numeric matrix of dimension 300 x 300, representing the true multivariate eigenfunctions evaluated on a 300-point grid.
- **vals** A numeric vector of length 300, containing the true multivariate eigenvalues.

True_scores A numeric matrix of dimension 100 x 9, containing the true principal component scores for 100 subjects across 9 multivariate components.

sigma A numeric value, the true error variance (sigma^2) of the observation noise.

Source

Simulated data based on: Li, C., Xiao, L., and Luo, S. (2020). Fast covariance estimation for multivariate sparse functional data. Stat, 9(1):e245. https://onlinelibrary.wiley.com/doi/10.1002/sta4.245

Examples

```
# example code
 sp_mult_sim
```

Index