Package 'fdacluster'

May 9, 2022

Title Joint Clustering and Alignment of Functional Data

Version 0.1.1

Type Package

Description Revisited clustering approaches to accommodate functional data by allowing to jointly align the data during the clustering process. Currently, shift, dilation and affine transformations only are available to perform alignment. The k-mean algorithm has been extended to integrate alignment and is fully parallelized. Hierarchical clustering will soon be available as well. References: Sangalli L.M., Secchi P., Vantini S., Vitelli V. (2010) ``k-mean alignment for curve clustering'' <doi:10.1016/j.csda.2009.12.008>.

License GPL (>= 3)

Encoding UTF-8

LazyData true

LinkingTo Rcpp, RcppArmadillo, nloptr

RoxygenNote 7.1.2

Suggests testthat

Imports Rcpp, magrittr, tibble, dplyr, tidyr, purrr, ggplot2, nloptr

Depends R (>= 2.10)

URL `https://astamm.github.io/fdacluster/index.html`,
`https://github.com/astamm/fdacluster`

NeedsCompilation yes

Author Laura Sangalli [aut],
Piercesare Secchi [aut],
Aymeric Stamm [cre, ctb] (<`https://orcid.org/0000-0002-8725-3654`>),
Simone Vantini [aut],
Valeria Vitelli [aut],
Alessandro Zito [ctb]

Maintainer Aymeric Stamm <`aymeric.stamm@math.cnrs.fr`>

Repository CRAN

Date/Publication 2022-05-09 12:10:11 UTC

R topics documented:

`aneurisk65` *Subset of the AneuRisk65 benchmark data set*

Description

A data set containing the first derivative of the three-dimensional coordinates of the centerline of the internal carotid artery of 65 patients.

Usage

```
aneurisk65
```

Format

A list with 2 components:

x A 65 x 1380 matrix containing, in each row, the evaluation grid for each patient;

y A 65 x 3 x 1380 array containing, in each row, the values of the first derivative of each of the 3D coordinates of the ICA centerline, stored by a row in a matrix.

Source

This is a subset of the AneuRisk65 benchmark data set provided by the AneuRisk project.

References

1. Sangalli, L.M., Secchi, P. and Vantini, S. (2014), AneuRisk65: A dataset of three-dimensional cerebral vascular geometries, Electronic Journal of Statistics, 8 (2), 1879-1890.
2. Sangalli, L.M., Secchi, P. and Vantini, S. (2014), Analysis of AneuRisk65 data: K-mean Alignment, Electronic Journal of Statistics, 8 (2), 1891-1904.

`fdacluster` *K-mean alignment algorithm and variants for functional data*

Description

The **fdacluster** package allows to jointly perform clustering and alignment of functional data.

References

1. Sangalli, L.M., Secchi, P., Vantini, S. and Vitelli, V. (2010), K-mean alignment for curve clustering, Computational Statistics and Data Analysis, 54, 1219-1233.
2. Sangalli, L.M., Secchi, P. and Vantini, S. (2014), Analysis of AneuRisk65 data: K-mean Alignment, Electronic Journal of Statistics, 8 (2), 1891-1904.

`kma` *K-mean alignment and variants for functional data*

Description

K-mean alignment and variants for functional data

Usage

```
kma(
  x,
  y,
  seeds = NULL,
  warping_options = c(0.15, 0.15),
  n_clust = 1,
  maximum_number_of_iterations = 100,
  number_of_threads = 1,
  parallel_method = 0,
  distance_relative_tolerance = 0.001,
  use_fence = FALSE,
  check_total_dissimilarity = TRUE,
  use_verbose = TRUE,
  compute_overall_center = FALSE,
  warping_method = "affine",
  center_method = "mean",
  dissimilarity_method = "l2",
  optimizer_method = "bobyqa"
)
```

Arguments

x
: A matrix of size nObs x nPts storing the evaluation grid of each observation.

y
: An 3D array of size nObs x nDim x nPts storing the observation values.

seeds
: A vector of integers of size `n_clust` specifying the indices of the initial templates. Defaults to `NULL`, which boils down to randomly sampled indices.

warping_options
: A numeric vector supplied as a helper to the chosen `warping_method` to decide on warping parameter bounds.

n_clust
: An integer specifying the number of clusters (default: 1).

maximum_number_of_iterations
: An integer specifying the maximum number of iterations before the algorithm stops (default: 100).

number_of_threads
: An integer specifying the number of threads used for parallelization (default: 1).

parallel_method
: An integer value specifying the type of desired parallelization for template computation, If 0 (default), templates are computed in parallel. If 1, parallelization occurs within a single template computation (only for the medoid method as of now).

distance_relative_tolerance
: A number specifying a relative tolerance on the distance update between two iterations. If all observations have not sufficiently improved in that sense, the algorithm stops. Defaults to 1e-3.

use_fence
: A boolean specifying whether the fence algorithm should be used to robustify the algorithm against outliers (default: `FALSE`).

check_total_dissimilarity
: A boolean specifying whether an additional stopping criterion based on improvement of the total dissimilarity should be used (default: `TRUE`).

use_verbose
: A boolean specifying whether the algorithm should output details of the steps to the console (default: `TRUE`).

compute_overall_center
: A boolean specifying whether the overall center should be also computed (default: `FALSE`).

warping_method
: A string specifying the warping method. Choices are `"none"`, `"shift"`, `"dilation"` and `"affine"` (default).

center_method
: A string specifying the center method. Choices are `"medoid"` and `"mean"` (default).

dissimilarity_method
: A string specifying the dissimilarity method. Choices are `"pearson"` and `"l2"` (default).

optimizer_method
: A string specifying the optimizer method. The only choice for now is `"bobyqa"`.

Value

The function output is a `kmap` object, which is a list with the following elements:

`x` As input.

`y` As input.

`seeds` Indices used in the algorithm.

`iterations` Number of iterations before the KMA algorithm stops.

`n_clust` As input.

`overall_center_grid` Overall center grid if `compute_overall_center` is set.

`overall_center_values` Overall center values if `compute_overall_center` is set.

`distances_to_overall_center` Distances of each observation to the overall center if `compute_overall_center` is set.

`x_final` Aligned observation grids.

`n_clust_final` Final number of clusters. Note that `n_clust_final` may differ from initial number of clusters `n_clust` if some clusters are empty.

`x_centers_final` Final center grids.

`y_centers_final` Final center values.

`template_grids` List of template grids at each iteration.

`template_values` List of template values at each iteration.

`labels` Cluster memberships.

`final_dissimilarity` Distances of each observation to the center of its assigned cluster.

`parameters_list` List of estimated warping parameters at each iteration.

`parameters` Final estimated warping parameters.

`timer` Execution time step by step.

`warping_method` As input.

`dissimilarity_method` As input.

`center_method` As input.

`optimizer_method` As input.

Examples

```
res <- kma(
  simulated30$x,
  simulated30$y,
  seeds = c(1, 21),
  n_clust = 2,
  center_method = "medoid",
  warping_method = "affine",
  dissimilarity_method = "pearson"
)
```

---

plot.kma *Plot for* kmap *objects*

Description

Plot for kmap objects

Usage

```
## S3 method for class 'kma'
plot(x, type = "data", number_of_displayed_points = 50, ...)
```

Arguments

`x` The kma object to be plotted.

`type` A string specifying the type of information to display. Choices are `"data"` for plotting the original and aligned curves (default) or `"warping"` for plotting the corresponding warping functions.

`number_of_displayed_points` The number of points to used for display. It is set as the minimum between this parameter and the number of points in the original data set. Defaults to 50.

`...` Other graphical parameters (see par). Ignored for now.

Value

A ggplot object invisibly.

Examples

```
res <- kma(
  simulated30$x,
  simulated30$y,
  seeds = c(1, 21),
  n_clust = 2,
  center_method = "medoid",
  warping_method = "affine",
  dissimilarity_method = "pearson"
)

plot(res, type = "data")
plot(res, type = "warping")
```

simulated30 *Simulated data for examples.*

Description

A data set containing 30 simulated uni-dimensional curves.

Usage

```
simulated30
```

Format

A list with abscissas x and values y:

x Matrix 30x200;

y Array 30x1x200.

simulated90 *Simulated data from the CSDA paper*

Description

A data set containing 90 simulated uni-dimensional curves.

Usage

```
simulated90
```

Format

A list with abscissas x and values y:

x Vector of size 100;

y Matrix if size 90x100.

Index