Package 'fdakma'

December 23, 2022

Title Functional Data Analysis: K-Mean Alignment

Version 1.3.0

Description It performs simultaneously clustering and alignment of a multidimensional or unidimensional functional dataset by means of k-mean alignment.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

URL https://github.com/astamm/fdakma

BugReports https://github.com/astamm/fdakma/issues

Imports cli, fdacluster

NeedsCompilation no

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 2022-12-23 19:10:11 UTC

R topics documented:

`kma` *K-Means Alignment Algorithm*

Description

This is an implementation of the k-means alignment algorithm originally described in Sangalli et al. (2010), with improvements as proposed in Vantini (2012).

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(
  fdacluster::simulated30$x,
  fdacluster::simulated30$y,
  seeds = c(1, 21),
  n_clust = 2,
  center_method = "medoid",
  warping_method = "affine",
  dissimilarity_method = "pearson"
)
```

Index