Package 'localFDA'

July 22, 2025

Type Package

Title Localization Processes for Functional Data Analysis

Version 1.0.0

Date 2020-09-24

Description
Implementation of a theoretically supported alternative to k-nearest neighbors for functional data to solve problems of estimating unobserved segments of a partially observed functional data sample,
functional classification and outlier detection. The approximating neighbor curves are piecewise functions built from a functional sample.
Instead of a distance on a function space we use a locally defined distance function that satisfies stabilization criteria.
The package allows the implementation of the methodology and the replication of the results in Elías, A., Jiménez, R. and Yukich, J. (2020) <doi:10.48550/arXiv.2007.16059>.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

URL https://github.com/aefdz/localFDA

BugReports https://github.com/aefdz/localFDA

Imports stats, graphics

Depends R (>= 2.10)

NeedsCompilation no

Author Antonio Elías [aut, cre],
Raul Jiménez [aut],
Joe Yukich [aut]

Maintainer Antonio Elías <antonioefz91@gmail.com>

Repository CRAN

Date/Publication 2020-09-30 08:30:09 UTC

Contents

`classificationData` *Two groups of Gaussian processes with different mean values*

Description

Two groups of Gaussian processes with different mean values

Usage

```
classificationData
```

Format

A matrix with n = 100 functions by columns and t = 200 evaluation points by row. The first 50 are G1 and second 50 curves are G2 that differs in the mean value.

References

Elías, Antonio, Jiménez, Raúl and Yukich, Joe (2020). Localization processes for functional data analysis (submitted).

Examples

```
matplot(classificationData, type = "l")
```

exampleData *Functional Gaussian processes.*

Description

Functional Gaussian processes.

Usage

```
exampleData
```

Format

A matrix with n = 1000 functions by columns and t = 100 evaluation points by row.

References

Elías, Antonio, Jiménez, Raúl and Yukich, Joe (2020). Localization processes for functional data analysis (submitted).

Examples

```
matplot(exampleData, type = "l")
```

localizationClassifier *Localization classifier*

Description

Given a training sample with g groups, it predicts the group of the test sample.

Usage

```
localizationClassifier(trainingSample, testSample, classNames, k_opt, g_pi)
```

Arguments

trainingSample	matrix p by n, being n the number of functions and p the number of grid points. The colnames of the trainingSample matrix are i_groupName where i goes from 1 to the sample size of the group.
testSample	matrix p by n, being n the number of functions to classify and p the number of grid points.
classNames	character vector with the group names.
k_opt	Maximum order of the localization processes used in the classification rule.
g_pi	Vector of size g with a priori probabilities for the bayes classifier. If it is missing the probability is defined by the proportion of curves of each group.

Value

Two named training and test. Training contains the estimations made with the training sample (localization statistics and localization distances). Test contains the classification results (for each incoming data, localization distances in each group, prior probabilities used, likelihood in each group and the predicted_class).

References

Elías, Antonio, Jiménez, Raúl and Yukich, Joe (2020). Localization processes for functional data analysis (submitted).

Examples

```
X <- classificationData
ids_training <- sample(colnames(X), 90)
ids_testing <- setdiff(colnames(X), ids_training)
trainingSample <- X[,ids_training]
testSample <- X[,ids_testing]; colnames(testSample) <- NULL #blind
classNames <- c("G1", "G2")
classification_results <- localizationClassifier(trainingSample, testSample, classNames, k_opt = 3)
```

---

localizationDistances *Localization distances*

Description

Compute the localization distances of order k of the curve `y0`.

Usage

```
localizationDistances(y, y0)
```

Arguments

`y`	matrix p by n, being n the number of functions and p the number of grid points.
`y0`	focal curve (index or character name).

Value

a vector of length (n-1), being the localization distance of its corresponding order.

References

Elías, Antonio, Jiménez, Raúl and Yukich, Joe (2020). Localization processes for functional data analysis (submitted).

Examples

```
localizationDistances_1 <- localizationDistances(exampleData, y0 = "1")
```

`localizationProcesses` *Localization processes*

Description

Compute the localization processes of order k of the curve `y0`.

Usage

```
localizationProcesses(y, y0)
```

Arguments

`y`	matrix p by n, being n the number of functions and p the number of grid points.
`y0`	focal curve index or name

Value

a list with one element, `lc`, a matrix of size p x (n-1), being the (n-1) columns the localization processes of its corresponding order.

References

Elías, Antonio, Jiménez, Raúl and Yukich, Joe (2020). Localization processes for functional data analysis (submitted).

Examples

```
localizationProcesses_1 <- localizationProcesses(exampleData, y0 = "1")
```

localizationStatistics *Localization Distances Statistics*

Description

Estimate the mean and standard deviation of the localization distances mean.

Usage

```
localizationStatistics(y, robustify = TRUE, whiskerrule)
```

Arguments

y	matrix p by n, being n the number of functions and p the number of grid points.
robustify	if TRUE the mean and standard deviation are estimated with a the trimmed sample. Default is TRUE.
whiskerrule	Range parameter for the univariate boxplot detection rule. Default = 3.

Value

a list with the localization distances of each function (localizationDistances), the estimated mean (mean) and standard deviation (sd).

References

Elías, Antonio, Jiménez, Raúl and Yukich, Joe (2020). Localization processes for functional data analysis (submitted).

Examples

```
localizationStatistics_full <- localizationStatistics(exampleData[,1:101], robustify = TRUE)
localizationStatistics_full$trim_mean[c(1, 25, 50 ,75, 100)]
localizationStatistics_full$trim_sd[c(1, 25, 50 ,75, 100)]
```

outlierData *Functional Gaussian processes with outliers.*

Description

Functional Gaussian processes with outliers.

Usage

```
outlierData
```

Format

A matrix with n = 54 functions by columns and t = 200 evaluation points by row. The last 4 observations are two shape and two magnitude outliers.

References

Elías, Antonio, Jiménez, Raúl and Yukich, Joe (2020). Localization processes for functional data analysis (submitted).

Examples

```
matplot(outlierData, type = "l")
```

`outlierLocalizationDistance` *Outlier localization distances*

Description

Compute the localization distances of order k of the curve `y0`.

Usage

```
outlierLocalizationDistance(X, localrule = 0.9, whiskerrule = 3)
```

Arguments

`X`	matrix p by n, being n the number of functions and p the number of grid points.
`localrule`	Local distance rule: the method marks a curve as outlier if its k order localization distances are outliers in more than local_rulex100 percent of the k-order univariate boxplots. Default is 0.90 so a function must be at least an outlier in 90 percent of the k-order localization distances.
`whiskerrule`	Parameter for the whiskers of the univariate boxplot of the localization distances of order kth. Default value is 3.

Value

A list

References

Elías, Antonio, Jiménez, Raúl and Yukich, Joe (2020). Localization processes for functional data analysis (submitted).

Examples

```
outliers <- outlierLocalizationDistance(outlierData, localrule = 0.9, whiskerrule = 3)
outliers$outliers_ld_rule
```

Index