Package ‘FADPclust’

January 20, 2025

Type Package

Title Functional Data Clustering Using Adaptive Density Peak Detection

Version 1.1.1

Maintainer Rui Ren <xmurr@stu.xmu.edu.cn>

Description An implementation of a clustering algorithm for functional data based on adaptive density peak detection technique, in which the density is estimated by functional k-nearest neighbor density estimation based on a proposed semi-metric between functions. The proposed functional data clustering algorithm is computationally fast since it does not need iterative process. (Alex Rodriguez and Alessandro Laio (2014) <doi:10.1126/science.1242072>; Xiao-Feng Wang and Yifan Xu (2016) <doi:10.1177/0962280215609948>).

License GPL (>= 2)

Depends R (>= 3.5.0)

Imports MFPCA, cluster, fpc, fda, fda.usc, funData, stats, graphics

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Author Rui Ren [aut, cre],
Kuangnan Fang [aut],
Qingzhao Zhang [aut],
Xiaofeng Wang [aut]

Repository CRAN

Date/Publication 2022-11-07 14:10:02 UTC

Contents

`FADPclust` *Functional Data Clustering Using Adaptive Density Peak Detection*

Description

Clustering of univariate or multivariate functional data by finding cluster centers from estimated density peaks. FADPclust is a non-iterative procedure that incorporates KNN density estimation algorithm. The number of clusters can also be selected by the user or selected automatically through an internal clustering criterion.

Usage

```
FADPclust(fdata, cluster = 2:10, method = "FADP1", proportion = NULL,
  f.cut = 0.15, pve = 0.9, stats = "Avg.silhouette")
```

Arguments

`fdata`	for univariate functional data clustering: a functional data object produced by fd() function of fda package; for multivariate functional data clustering: a list of functional data objects produced by fd() function of fda package.
`cluster`	integer, or a vector of integers specifying the pool of the number of clusters in automatic variation. The default is 2:10.
`method`	character string specifying the method used to calculate the pseudo functional k-nearest neighbor density. Valid options of are 'FADP1' and 'FADP2' (see details in references). The default is 'FADP1'.
`proportion`	numeric, a number or numeric vector of numbers within the range [0,1], specifying to automatically select the smoothing parameter k in density estimation (see details). The default is 0.1, 0.2, ... ,1.
`f.cut`	numeric, a number within the range [0,1], specified to automatically select cluster centroids from the decision plot. The default is 0.15.
`pve`	numeric, a number within the range [0,1], the proportion of variance explained: used to choose the number of functional principal components. The default is 0.9. When the method is chosen to be 'FADP1', there is no need to specify parameter 'pve' for univariate functional data clustering.
`stats`	character string specifying the distance based statistics for cluster validation and determining the number of clusters. Valid options are 'Avg.silhouette', 'Dunn', and 'CH' (See the description document of the cluster.stats function in the fpc R package for more details about these statistics). The default is "Avg.silhouette".

Details

Given n functional objects or curves, FADPclust() calculates f(x) and delta(x) for each object based on the semi-metric distance (see details in references), where f(x) is the local density calculated by the functional k-nearest neighbor density estimator of curve x, and delta(x) is the shortest semi-metric distance between sample curve x and y for all samples y such that f(x) <= f(y). Functional objects or curves with large f and large delta values are labeled class centroids. In other words, they appear as isolated points in the upper right corner of the f vs delta plot (the decision plot, see details in FADPplot). After cluster centroids are determined, other obejects are clustered according to their semi-metric distances to the closes centroids.

The smoothing parameter k in functional k-nearest neighbor density estimation must be explicitly provided. Following Lauter (1988)'s idea, suggest that the optimal size of k satisfies a certain proportion, k = a*n^(4/5), where a is a parameter about the optimal proportion to be determined. Here, users enters variable 'proportion' to specify the parameter a.

Value

An 'FADPclust' object that contains the list of the following items.

- nclust: number of clusters.
- para: smoothing parameter k selected automatically by KNN estimation.
- method: character string introducing the method used to calculate the smoothing parameter.
- clust: cluster assignments. A vector of the same length as the number of observations.
- density: final density vector f(x).
- delta: final delta vector delta(x).
- center: indices of the clustering centers.
- Avg.silhouette: average silhouette score from the final clustering result.
- Dunn: Dunn statistics from the final clustering result.
- CH: CH statistics from the final clustering result.

References

- Lauter, H. (1988), "Silverman, B. W.: "Density Estimation for Statistics and Data Analysis.," Biometrical Journal, 30(7), 876-877.
- Wang, X. F., and Xu, Y. (2016), "Fast Clustering Using Adaptive Density Peak Detection," Statistical Methods in Medical Research.
- Rodriguez, A., and Laio, A. (2014), "Machine learning. Clustering by fast search and find of density peaks," Science, 344(6191), 1492.
- Liu Y, Ma Z, and Yu F. (2017), "Adaptive density peak clustering based on K-nearest neighbors with aggregating strategy," Knowledge-Based Systems, 133(oct.1), 208-220.

See Also

FADPsummary, FADPplot.

Examples

```
###univariate functional data
data("simData1")
plot(simData1, xlab = "x", ylab = "y")
FADP1.sil.ans <- FADPclust(fdata = simData1, cluster = 2:5, method = "FADP1",
                           proportion = seq(0.02, 0.2, 0.02), f.cut = 0.15,
```

```
                           stats = "Avg.silhouette")
FADPsummary(FADP1.sil.ans); FADPplot(FADP1.sil.ans)

FADP1.dunn.ans <- FADPclust(fdata = simData1, cluster = 2:5, method = "FADP1",
                            proportion = seq(0.02, 0.2, 0.02), f.cut = 0.15,
                            stats = "Dunn")
FADPsummary(FADP1.dunn.ans); FADPplot(FADP1.dunn.ans)

FADP1.ch.ans <- FADPclust(fdata = simData1, cluster = 2:5, method = "FADP1",
                          proportion = seq(0.02, 0.2, 0.02), f.cut = 0.15,
                          stats = "CH")
FADPsummary(FADP1.ch.ans); FADPplot(FADP1.ch.ans)

FADP2.ans <- FADPclust(fdata = simData1, cluster = 2:5, method = "FADP2",
                       proportion = seq(0.02, 0.2, 0.02), f.cut = 0.15,
                       pve = 0.9, stats = "Avg.silhouette")
FADPsummary(FADP2.ans); FADPplot(FADP2.ans)

###multivariate functional data
data("simData2")
FADP1.ans <- FADPclust(fdata = simData2, cluster = 2:5, method = "FADP1",
                       proportion = seq(0.02, 0.2, 0.02), f.cut = 0.15,
                       pve = 0.9, stats = "Avg.silhouette")
FADPsummary(FADP1.ans); FADPplot(FADP1.ans)

FADP2.ans <- FADPclust(fdata = simData2, cluster = 2:5, method = "FADP2",
                       proportion = seq(0.02, 0.2, 0.02), f.cut = 0.15,
                       pve = 0.9, stats = "Avg.silhouette")
FADPsummary(FADP2.ans); FADPplot(FADP2.ans)
```

`FADPplot` *Visualize the result of FADPclust*

Description

Plot the f vs delta plot with selected centroids.

Usage

```
FADPplot(object, cols = "default")
```

Arguments

`object`	object of class 'FADPclust' that is returned from FADPclust().
`cols`	vector of colors used to distinguish different clusters. Ten default colors are given.

See Also

FADPclust, FADPsummary.

Examples

```
###univariate functional data
data("simData1")
plot(simData1, xlab = "x", ylab = "y")
FADP1.ans <- FADPclust(fdata = simData1, cluster = 2:5, method = "FADP1",
                       proportion = seq(0.02, 0.2, 0.02), f.cut = 0.15,
                       stats = "Avg.silhouette")
FADPsummary(FADP1.ans); FADPplot(FADP1.ans)

FADP2.ans <- FADPclust(fdata = simData1, cluster = 2:5, method = "FADP2",
                       proportion = seq(0.02, 0.2, 0.02), f.cut = 0.15,
                       pve = 0.9, stats = "Avg.silhouette")
FADPsummary(FADP2.ans); FADPplot(FADP2.ans)

###multivariate functional data
data("simData2")
FADP1.ans <- FADPclust(fdata = simData2, cluster = 2:5, method = "FADP1",
                       proportion = seq(0.02, 0.2, 0.02), f.cut = 0.15,
                       pve = 0.9, stats = "Avg.silhouette")
FADPsummary(FADP1.ans); FADPplot(FADP1.ans)

FADP2.ans <- FADPclust(fdata = simData2, cluster = 2:5, method = "FADP2",
                       proportion = seq(0.02, 0.2, 0.02), f.cut = 0.15,
                       pve = 0.9, stats = "Avg.silhouette")
FADPsummary(FADP2.ans); FADPplot(FADP2.ans)
```

FADPsummary — *Summary of FADPclust*

Description

Summarize the result obetained from the FADPclust() function.

Usage

```
FADPsummary(object)
```

Arguments

object	object of class 'FADPclust' that is returned from FADPclust().

See Also

FADPclust, FADPplot.

Examples

```
###univariate functional data
data("simData1")
plot(simData1, xlab = "x", ylab = "y")
FADP1.ans <- FADPclust(fdata = simData1, cluster = 2:5, method = "FADP1",
                       proportion = seq(0.02, 0.2, 0.02), f.cut = 0.15,
                       stats = "Avg.silhouette")
FADPsummary(FADP1.ans); FADPplot(FADP1.ans)

FADP2.ans <- FADPclust(fdata = simData1, cluster = 2:5, method = "FADP2",
                       proportion = seq(0.02, 0.2, 0.02), f.cut = 0.15,
                       pve = 0.9, stats = "Avg.silhouette")
FADPsummary(FADP2.ans); FADPplot(FADP2.ans)

###multivariate functional data
data("simData2")
FADP1.ans <- FADPclust(fdata = simData2, cluster = 2:5, method = "FADP1",
                       proportion = seq(0.02, 0.2, 0.02), f.cut = 0.15,
                       pve = 0.9, stats = "Avg.silhouette")
FADPsummary(FADP1.ans); FADPplot(FADP1.ans)

FADP2.ans <- FADPclust(fdata = simData2, cluster = 2:5, method = "FADP2",
                       proportion = seq(0.02, 0.2, 0.02), f.cut = 0.15,
                       pve = 0.9, stats = "Avg.silhouette")
FADPsummary(FADP2.ans); FADPplot(FADP2.ans)
```

---

simData1 — *Simulated univariate functional data for method FADPclust*

Description

Simulated univariate functional data, with 2 clusters each containing 100 sample curves, were for users to apply the method FADPclust.

Format

fd, see FDA R package for details.

---

simData2 — *Simulated multivariate functional data for method FADPclust*

Description

Simulated three-dimensional multivariate functional data, with 2 clusters each containing 100 sample curves, were for users to apply the method FADPclust.

Format

fd, see FDA R package for details.

Index