Package 'DoTC'

June 17, 2016

Type Package

Title Distribution of Typicality Coefficients

Version 0.2

Date 2016-06-17

Author Philip Wadewitz, Julia Fischer, Kurt Hammerschmidt, Demian Battaglia, Holger Sennhenn-Reulen

Maintainer Holger Sennhenn-Reulen <hsennhenn-reulen@dpz.eu>

Description
Calculation of cluster typicality coefficients as being generated by fuzzy k-means clustering.

Depends ggplot2, plyr

Imports methods

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2016-06-17 21:20:50

R topics documented:

extract 2
getStart 2
plotCS 3
plotNcluster 4
plotTC 5
wrapFKM 5

Index **8**

extract *Auxiliary Function for Extracting Elements from the Result of wrapFKM*

Description

Function for extracting elements from the result of wrapFKM, in especially if m is a vector.

Usage

```
extract(x, what)
```

Arguments

x	result from wrapFKM
what	what should be extracted (possible values are U, combined, remaining, combinations, and n_cluster).

Value

The argument that should be extracted from fkm.

Author(s)

Holger Sennhenn-Reulen

Examples

```
## Not run: extract(x, what)
```

getStart *Calculate Starting Values for Fuzzy k-means Clustering*

Description

This function calculates starting values as used in the wrapFKM function.

Usage

```
getStart(d, K = 10, nrep = 100, safety = TRUE, ...)
```

Arguments

d	data-set with standardized columns
K	number of clusters
nrep	number of repetitions for the call to kmeans (default is 100).
safety	As described in the help to kmeans, the algorithm may not converge in the quick-transfer stage. If safety is set to TRUE, each run where this happens is discarded and repeated from a different random starting point.
...	further arguments to kmeans.

Details

This function calculates starting values for wrapFKM as the solution of k-means clustering.

Value

A matrix with K columns and nrow(d) rows.

Author(s)

Holger Sennhenn-Reulen

Examples

```
## Not run: getStart(d, K = 10, nrep = 100, safety = TRUE, ...)
```

---

plotCS *Plot Cluster Segregation*

Description

Pairwise comparisons of cluster segregations.

Usage

```
plotCS(fkm, which_clusters = NULL, colors = NULL, main = "")
```

Arguments

fkm	Result for one single fuzziness parameter m as calculated by wrapFKM.
which_clusters	Which clusters should be plotted? (Default is NULL, and all pair-wise cluster combinations are plotted).
colors	colors to be used (default is NULL, and colors are automatically provided)
main	main title (default is no title)

Value

A plot with pairwise comparisons of cluster segregations.

Author(s)

Holger Sennhenn-Reulen

Examples

```
## Not run: plotCS(fkm, which_clusters = NULL, colors = NULL, main = "")
```

`plotNcluster` *Plot the Cluster Solution Across Varying Fuzziness Parameter*

Description

Plot the cluster solution, ie. the number of clusters, as a step function across varying fuzziness parameter `m`.

Usage

```
plotNcluster(fkm, ...)
```

Arguments

`fkm`	Result for one single fuzziness parameter `m` as calculated by `wrapFKM`.
`...`	Additional attributes to `plot`.

Value

A step plot of the numbers of clusters (y axis) across different values for the fuzziness parameter m (x axis). The largest values of m conditional on a fixed numbers of clusters are highlighted using grey, dashed lines and bullet points.

Author(s)

Holger Sennhenn-Reulen

Examples

```
## Not run: plotNcluster(fkm, ...)
```

plotTC *Plot Typicality Coefficients*

Description

Plot Typicality Coefficients as stapled Histograms

Usage

```
plotTC(fkm, main = NULL)
```

Arguments

fkm	Result from wrapFKM.
main	Main title (default is NULL, resulting in a main title with the fuzziness parameter m).

Details

Relies on ggplot2 and plyr.

Value

A plot with frequencies of typicality coefficients.

Author(s)

Holger Sennhenn-Reulen

Examples

```
## Not run: plotTC(fkm, main = NULL)
```

wrapFKM *Wrapper for* FKM

Description

Wrapper function for a call to fuzzy k-means function FKM.

Usage

```
wrapFKM(d, m, start, maxit = 1e4, threshold = 0.1)
```

Arguments

d	data-set with standardized columns
m	Fuzziness parameter
start	Starting values as provided by getStart.
maxit	Maximum number of iterations (default is 10000).
threshold	Upper limit below which a cluster distance (as defined by matrix H in FKM) is set to be 0.

Details

The function is a specific wrapper function to a function which gives the same results as FKM from the R package fclust.

The below example gives an exemplary complete run for an analysis as implemented by this package DoTC.

The selection of the fuzziness parameter m is crucial for the result of the wrapFKM function. We have good experiences with following strategy: first use a coarse grid of proposal values for m, look on aggregation of clusters across this proposal vector – as for example implemented by getStart –, and then refine this grid for a certain sub-interval of interest.

Value

A list with the results from the call to FKM:

U (matrix containing case-wise (rows) cluster (columns) affiliation values), H (pair-wise cluster distance matrix), value (terminal value of the fuzzy-clustering algorithm), iter (number of iterations needed to get to value), k (number od proposed clusters to the start solution start), call (call to the interior FKm function), combined (which clusters are members of any combinations), remaining (which clusters stay remaining), combinations (which are the combinations that lead to the reduction), and n_cluster (number of reduced clusters),

and further attributes of the solution across potential different proposal fuzziness values:

m (all proposed fuzziness values), n_cluster (the number of reduced clusters), m_before_step (the maximum fuzziness parameter before a reduction in n_cluster), and which_list_indexes_m_before_step (where are the respective results to m_before_step).

Author(s)

Holger Sennhenn-Reulen

References

Paolo Giordani, Maria Brigida Ferraro (2015). *fclust: Fuzzy Clustering*, on CRAN.

Examples

```
## Not run:
## Load and standardize (by column) data:
d <- read.csv("data_file.csv")
```

```
d <- apply(d, MAR = 2, FUN = scale)
## Set maximal number of clusters:
K <- 10
## Set random seed:
set.seed(1604)
## Get k-means-clustering solutions as starting values:
start <- getStart(d = d, K = K)
## Proposal vector for fuzziness parameter m:
m_proposal <- seq(1.1, 2.5, by = 0.1)
## Calculate results of fuzzy clustering:
fkm_result <- wrapFKM(d = d, m = m_proposal, start = start)
## Plot cluster solution across varying m:
plotNcluster(fkm = fkm_result)
## Plot distribution of typicality coefficients:
plotTC(fkm_result[[1]])
## Plot pairwise cluster segregation comparisons:
plotCS(fkm_result[[1]])
## End(Not run)
```

Index

extract, 2

getStart, 2, *6*

plotCS, 3
plotNcluster, 4
plotTC, 5

wrapFKM, *2–5*, 5