Package 'akmedoids'

March 5, 2019

Type Package

Title Akmedoids: 'Anchored kmedoids' for longitudinal data clustering

Version 0.1.0

Date 2019-02-06

Author Monsuru Adepeju [cre, aut], Samuel Langton [aut], Jon Bannister [aut]

Maintainer Monsuru Adepeju <monsuurg2010@gmail.com>

Description Advances a longitudinal clustering technique, 'akmedoids' for grouping trajectories based on the similarities of their long-term trends and determines the optimal solution based on the Calinski and Harabatz criterion. The 'Statsplot' function included helps to extract the descriptive statistics and generate a visualisation of the resulting groups. The package also include a list of other useful functions for exploring and manipulating longitudinal data prior to the clustering process.

Depends R (>= 3.5.0)

License GPL-2

Encoding UTF-8

LazyData true

Imports kml, devtools, Hmisc, ggplot2, rgdal, base, utils, reshape2, later, Rdpack, longitudinalData

RoxygenNote 6.1.1

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

`akmedoids.clust` *Anchored k-medoids clustering*

Description

Given a list of trajectories and a functional method, this function clusters the trajectories into a k number of groups. If a vector of two numbers is given, the function determines the best solution based on the Calinski-Harabatz criterion.

Usage

```
akmedoids.clust(traj, id_field = FALSE, method = "linear", k = c(3,6))
```

Arguments

`traj`	[matrix (numeric)]: longitudinal data. Each row represents an individual trajectory (of observations). The columns show the observations at consecutive time steps.
`id_field`	[numeric or character] Whether the first column of the `traj` is a unique (`id`) field. Default: `FALSE`. If `TRUE` the function recognises the second column as the first time step.
`method`	[character] The parametric Initialisation strategy. Currently, the only available method is `linear` method, set as `"linear"`. This uses the time-dependent linear regresion lines and the resulting groups are order in the order on increasing slopes.
`k`	[integer or vector (numeric)] either an exact integer number of clusters, or a vector of length two specifying the minimum and maximum numbers of clusters to be examined from which the best solution will be determined. In either case, the minimum number of clusters is `3`. The default is `c(3,6)`. The best solution is determined using the Calinski-Harabatz criterion (`Calinski T. & Harabasz J. 1974`).

Details

This function works by first approximating the trajectories based on the chosen parametric forms (e.g. linear, quadratic etc.), and then partition the original trajectories based on the form groupings, in similar fashion k-means clustering approach (`Genolini et al. 2015`). The key distintion of the `akmedoids` as compared with the existing longitudinal approaches is that both the initial centroids as well as the subsequent cluster means (as the iteration progresses) are based the selection of medoids observations.

Value

The key output is a vector of cluster labels of length equal to the number of trajectories. Each label indicate the group membership of the corresponding trajectory of the `traj`. In addition, a plot of the Calinski-Harabatz scores is shown is shown if a vector of `k` is provided.

References

1. Genolini, C. et al. (2015). kml and kml3d: R Packages to Cluster Longitudinal Data. Journal of Statistical Software, 65(4), 1-34. URL http://www.jstatsoft.org/v65/i04/. 2. Calinski T, Harabasz J (1974) A dendrite method for cluster analysis. Commun Stat 3:1-27. 3. Genolini, C.et al. (2016) Package 'longitudinalData'

Examples

```
traj <- gm.crime.sample1
print(traj)
traj <- dataImputation(traj, id_field = TRUE, method = 2, replace_with = 1, fill_zeros = FALSE)
traj <- props(traj, id_field = TRUE)
print(traj)
output <- akmedoids.clust(traj, id_field = TRUE, method = "linear", k = c(3,6))
print(output)  #type 'as.vector(output$optimSolution)'
```

---

alphaLabel *Numerics ids to alphabetical ids*

Description

Function to transform a list of numeric ids to alphabetic ids

Usage

```
alphaLabel(x)
```

Arguments

x	A vector of numeric ids

---

dataImputation *Data imputation for longitudinal data*

Description

This function fills any missing entries (NA, Inf, null) in a matrix or dataframe, according to a specified method. By default, '0' is considered a value.

Usage

```
dataImputation(traj, id_field = FALSE, method = 2, replace_with = 1, fill_zeros = FALSE)
```

Arguments

traj	[matrix (numeric)]: longitudinal data. Each row represents an individual trajectory (of observations). The columns show the observations at consecutive time steps.
id_field	[numeric or character] Whether the first column of the traj is a unique (id) field. Default: FALSE. If TRUE the function recognises the second column as the first time step.
method	[an integer] indicating a method for calculating the missing values. Options are: '1': arithmetic method, and '2': regression method. The default is '1': arithmetic method

replace_with	[an integer from 1 to 6] indicating the technique, based on a specified method, for calculating the missing entries. '1': arithmetic method, replace_with options are: '1': Mean value of the corresp column; '2': Minimum value of corresp column; '3': Maximum value of corresp column; '4': Mean value of corresp row; '5': Minimum value of corresp row, or '6': Maximum value of corresp row. For '2': regression method: the available option for the replace_with is: '1': linear. The regression method fits a linear regression line to a trajectory with missing entry(s) and estimate the missing data values from the regression line. Note: only the missing data points derive their new values from the regression line while the rest of the data points retain their original values. The function terminates if there are trajectories with only one observation. The default is '1': Mean value of the corresp column
fill_zeros	[TRUE or FALSE] whether to consider zeros 0 as missing values when 2: regression method is used. The default is FALSE.

Details

Given a matrix or data.frame with some missing values indicated by (NA, Inf, null), this function impute the missing value by using either an estimation from the corresponding rows or columns, or to use a regression method to estimate the missing values.

Value

A data.frame with missing values (NA, Inf, null) imputed according to the a specified technique.

Examples

```
traj <- gm.crime.sample1
 print(traj)
 dataImputation(traj, id_field = TRUE, method = 1, replace_with = 1, fill_zeros = FALSE)
```

---

gm.crime.sample1 *Longitudinal dataset*

Description

Simulated longitudinal datasets with missing values (NA, Inf, null)

Usage

```
gm.crime.sample1
```

Format

A matrix

`outlierDetect` *Outlier detection and replacement*

Description

This function identifies outlier observations in the trajectories, and allows a user to replace the observations or remove trajectories entirely.

Usage

```
outlierDetect(traj, id_field = FALSE, method = 1, threshold = 0.95, count = 1, replace_with = 1)
```

Arguments

`traj`	[matrix (numeric)]: longitudinal data. Each row represents an individual trajectory (of observations). The columns show the observations at consecutive time steps.
`id_field`	[numeric or character] Whether the first column of the `traj` is a unique (`id`) field. Default: `FALSE`. If `TRUE` the function recognises the second column as the first time step.
`method`	[integer (numeric)] indicating the method for identifying the outlier. Options are: `'1': quantile` method (`default`), and `'2': manual` method. The `manual` method requires a user-defined value.
`threshold`	[numeric] A cut-off value for outliers. If the `method` parameter is set as `'1': quantile`, the `threshold` should be a numeric vector of probability between `[0,1]`, whilst if the `method` is set as `'2': manual`, the `threshold` could be any numeric vector.
`count`	[integer (numeric)] indicating the number of observations (in a trajectory) that must exceed the `threshold` in order for the trajectory to be considered an `outlier`. Default is `1`.
`replace_with`	[integer (numeric)] indicating the technique to use for calculating a replacement for an outlier observation. The remaining observations on the row or the column in which the outlier observation is located are used to calculate the replacement. The replacement options are: `'1'`: Mean value of the column, `'2'`: Mean value of the row and `'3'`: remove the row (trajectory) completely from the data. Default value is the `'1'` option.

Details

Given a matrix, this function identifies outliers that exceed the threshold and replaces the outliers with an estimate calculated using the other observations either the rows or the columns in which the outlier observation is located. Option is also provided to remove the trajectories (containing the outlier) from the data.

Value

A dataframe with outlier observations replaced or removed.

Examples

```
traj <- gm.crime.sample1
traj <- dataImputation(traj, id_field=TRUE, method = 1, replace_with = 1)
traj <- props(traj, id_field=TRUE)#remove this later
outlierDetect(traj, id_field = TRUE, method = 1, threshold = 0.95, count = 1, replace_with = 1)
outlierDetect(traj, id_field = TRUE, method = 2, threshold = 15, count = 4, replace_with = 3) #for method 2
```

---

population *sample population (denominator) data*

Description

Simulated denominator data with missing observation

Usage

```
population
```

Format

A matrix

---

props *Convert counts or rates to proportion*

Description

This function converts counts or rates to proportion.

Usage

```
props(traj, id_field = FALSE)
```

Arguments

traj	[matrix (numeric)]: longitudinal data. Each row represents an individual trajectory (of observations). The columns show the observations at consecutive time steps.
id_field	[numeric or character] Whether the first column of the traj is a unique (id) field. Default: FALSE. If TRUE the function recognises the second column as the first time step.

Details

Given a matrix of observations (counts or rates), this function convert each observation to a proportion equivalent of the sum of each column. In other words, each observation is dividing by the sum of the column where it is located. , i.e. prop = [a cell value] / sum[corresponding column]

Value

A matrix of proportion measures

Examples

```
traj <- gm.crime.sample1
head(traj) #
traj <- dataImputation(traj, id_field = TRUE, method = 2, replace_with = 1,
fill_zeros = FALSE) #filling the missing values
traj <- props(traj, id_field = TRUE)
print(traj)
```

Statsplot *Descriptive (Change) statistics and plots*

Description

This function perform two tasks: (i) it generate the descriptive and change statistics of groups, particularly suited for the outputs form the `akmedoids.clust` function, and (ii) generates the plots of the groups (performances).

Usage

```
Statsplot(clustr, traj, id_field = TRUE, bandw = 0.25,
  type = "lines", y.scaling = "fixed")
```

Arguments

`clustr`	[vector (charater)] A vector of cluster membership (labels). For instance, the result extracted from the `akmedoids.clust` function.
`traj`	[matrix (numeric)]: corresponding longitudinal data used to generate `clustr` (with rows corresponding to each label of `clustr`). For example, the first label of `clustr` is the group label of the first row of `traj` matrix, and so on.
`id_field`	[numeric or character] Whether the first column of the `traj` is a unique (`id`) field. Default: `FALSE`. If `TRUE` the function recognises the second column as the first time step.
`bandw`	[numeric] A small probability (quantile) value between `[0,1]` to partition the trajectories into three classes, i.e. `lower`, `central`, and the `uppper` classes. The middle of the `central` class is defined by the average slope of all trajectories. The upper and the lower limits of the `central` class is determined by the value of `bandw`. Default value is `0.25`, indicating that all slopes within 25th quantiles of the maximum slopes on both sides of the average slope are categorised as `central` class.
`type`	[character] plot type. Available options are: `"lines"` and `"stacked"`.
`y.scaling`	[character] works only if `type="lines"`. `y.scaling` set the vertical scales of the cluster panels. Options are: `"fixed"`: uses uniform scale for all panels, `"free"`: uses variable scales for panels.

Details

Generates the descriptive and change statistics of the trajectory groupings. Given a vector of group membership (labels) and the corresponding data matrix (or data.frame) indexed in the same order, this function generates all the descriptive and change statistics of all the groups. The function can generate a 'line' and a 'area stacked' plot drawing from the functionalities of the `ggplot` library. Therefore, for a more customised visualisation, we recommend that users employ `ggplot` directly (`Wickham H. (2016)`).

Value

A plot showing group membership or sizes (proportion) and statistics.

References

`Wickham H. (2016). Elegant graphics for Data Analysis. Spring-Verlag New York (2016)`

Examples

```
traj <- gm.crime.sample1
print(traj)
traj <- dataImputation(traj, id_field = TRUE, method = 1, replace_with = 1, fill_zeros = FALSE)
print(traj)
traj <- props(traj, id_field = TRUE)
clustr <- akmedoids.clust(traj, id_field = TRUE, method = "linear", k = c(3,6))
clustr <- as.vector(clustr$optimSolution)
print(Statsplot(clustr, traj, id_field=TRUE, type="lines", y.scaling="fixed"))
print(Statsplot(clustr, traj, id_field=TRUE, bandw = 0.60, type="stacked"))
```

---

wSpaces *Removing whitespaces*

Description

This function removes all the leading and the trailing whitespaces in a longitudinal data

Usage

```
wSpaces(traj)
```

Arguments

`traj`	[matrix (numeric)]: longitudinal data. Each row represents an individual trajectory (of observations). The columns show the observations at consecutive time steps.

Details

Given a matrix suspected to contain whitespaces, this function removes all the whitespaces and returns a cleaned data. 'Whitespaces' are white characters often introduced into data by typo errors or systematically by data recording devices. As a example, a character like "A" is not the same as " A" or "A ", containing whitespaces. , in a character such as " abcd" (or "abcd ") is not the same as "abcd". The former contains a leading (or trailing) white-characters.

Value

A matrix with all whitespaces (if any) removed.

References

https://en.wikipedia.org/wiki/Whitespace_character

Examples

```
traj <- gm.crime.sample1
wSpaces(traj)
```

Index