

Package 'rclusterstudy'

January 2, 2015

rclusterstudy

A package for studying clustering algorithms for spatial data

Description

SPDB project studying clustering algorithms for spatial data.

More: Comparison of the quality and speed of spatial data clustering using a clustering algorithm PAM in two versions:

- Spatial data are treated the same as other data - is the one calculated the distance (similarity) on the basis of all available attributes.

- The distance (similarity) between objects is calculated based on two components: spatial distance and distance (similarity) calculated on the basis of descriptive attributes.

Main function: spatialCluster

Details

Package: rclusterstudy

Version: 0.9

Date: 2015-01-02

Depends: R (>= 3.1.1), sp, stringdist, cluster, rgeos, geosphere

Author(s)

Mateusz Zawislak <mateusz.andrzej.zawislak@gmail.com>

Examples

```
# Mississippi Police Crime Reports data description
```

```
police.data.description <- list(  
  "numerical.columns" = list(list(col = "TAX"), list(col = "TRANSFER"), list(col = "INC"),  
                             list(col = "CRIME"), list(col = "UNEMP"), list(col = "OWN"),  
                             list(col = "COLLEGE"), list(col = "WHITE"), list(col = "COMMUTE")),  
  "polygons" = list(list(col = "V22"))  
)
```

```
police.clustering.description <- list (
```

```

"data.description" = police.data.description,

"params" = list(
  "clusters.number" = 6,
  "weight.vector" = list("non.spatial" = 1, "spatial" = 14),
  "spatial" = TRUE
)
)

# cluster data
uciData <- read.csv(file="../data/police.csv", head=TRUE, sep=",", skip=0, dec=".")
spatialCluster(uciData, police.clustering.description)

```

calculateDistanceMatrix

Function that calculates objects distance matrix.

Description

It calculates only bottom-left part of matrix, because it is symmetric.

Usage

```
calculateDistanceMatrix(objects, clustering.description)
```

Arguments

objects List of objects.
clustering.description Data description and clustering params.

Value

Distance matrix between given objects.

calculateDistancesFromObject

Function that calculates distances between given object and the list of objects.

Description

Function that calculates distances between given object and the list of objects.

Usage

```
calculateDistancesFromObject(objects, object, clustering.description)
```

Arguments

objects	List of objects.
object	Core object.
clustering.description	Data description and clustering params.

Value

The vector of distances from object to each one from objects. Vector length is equal to objects number.

distanceBrayCurtis	<i>Function that returns Bray Curtis distance between given attribute's values.</i>
--------------------	---

Description

Function that returns Bray Curtis distance between given attribute's values.

Usage

```
distanceBrayCurtis(val1, val2)
```

Arguments

val1	Attribute value.
val2	Another attribute value.

Value

Bray Curtis distance between given attribute's values.

Examples

```
distanceBrayCurtis(3, 5)
```

distanceEuclidean	<i>Function that calculates Euclidian distance between two given points.</i>
-------------------	--

Description

Function that calculates Euclidian distance between two given points.

Usage

```
distanceEuclidean(x1, y1, x2, y2)
```

Arguments

x1	First point's x value.
y1	First point's y value.
x2	Second point's x value.
y2	Second point's y value.

Value

Euclidian distance between two given points.

Examples

```
distanceEuclidean(1.2, 3, 5, 6)
```

distanceHaversine	<i>Function that returns distance between two given geographic coordinates</i>
-------------------	--

Description

Function that returns distance between two given geographic coordinates

Usage

```
distanceHaversine(long1, lat1, long2, lat2)
```

Arguments

long1:	First point's longitude.
lat1:	First point's latitude.
long2:	Second point's longitude.
lat2:	Second point's latitude.

Value

Haversine distance between two given geographic coordinates.

Examples

```
distanceHaversine(1, 2, 1.1, 2.2)
```

distanceLevensthein	<i>Function that calculates Levensthein distance between two given strings.</i>
---------------------	---

Description

Function that calculates Levensthein distance between two given strings.

Usage

```
distanceLevensthein(x, y)
```

Arguments

x	First string.
y	Second string.

Value

Levensthein distance between two given strings.

Examples

```
distanceLevensthein("mateusz", "mateuss") # result: 1
```

distancePolygons	<i>Function that returns distance between two given polygons.</i>
------------------	---

Description

Function that returns distance between two given polygons.

Usage

```
distancePolygons(polygon1, polygon2)
```

Arguments

polygon1	Polygon description accepted by rgeos package.
polygon2	Second polygon.

Details

Polygons should be able to read by function readWKT from rgeos package.

Value

Length of the shortest path between the given polygons.

Examples

```
distancePolygons("POLYGON((1 1,5 1,5 5,1 5,1 1))","POLYGON((12 12,12 13,13 13,13 12,12 12))")
```

distanceReal	<i>Function that returns distance between two given numbers. It can be scaled to range [min.val, max.val]</i>
--------------	---

Description

Function that returns distance between two given numbers. It can be scaled to range [min.val, max.val]

Usage

```
distanceReal(val1, val2, min.val = 0, max.val = 1)
```

Arguments

val1	Attribute value.
val2	Another attribute value.
min.val	(Optional) Minimum value of the feature in the collection.
max.val	(Optional) Maximum value of the feature in the collection.

Value

Scaled distance between given attribute's values, which type is numerical.

Examples

```
distanceReal(3, 4)
distanceReal(5, 12.4, 3, 18.9)
```

getColumnIndex	<i>Function that returns column index for the given column name in data.frame</i>
----------------	---

Description

Function that returns column index for the given column name in data.frame

Usage

```
getColumnIndex(data, column.name)
```

Arguments

data	data
column.name	column name

Value

column index for the given column name

getColumnName	<i>Function that gives column name for the give column index in data.frame</i>
---------------	--

Description

Function that gives column name for the give column index in data.frame

Usage

```
getColumnName(data, index)
```

Arguments

data	Data.
index	Column index.

Value

Column name for the give column index.

objectsDistance	<i>Function that calculates distance between two given objects.</i>
-----------------	---

Description

It sums distances between all attributes values.

Usage

```
objectsDistance(d1, d2, clustering.description)
```

Arguments

d1	First object.
d2	Second object.
clustering.description	Data description and clustering params.

Value

Distance between two given objects as double.

spatialCluster

Function that clusters data with PAM algorithm.

Description

Clustering data description must have two parts: data.description which contains data description and params which contains clustering algorithm parameters.

Data must have data.frame type and can be (for example) read from file with read.csv function.

Usage

```
spatialCluster(data, clustering.description)
```

Arguments

data data to cluster (data.frame type, for example read by read.csv function)
clustering.description description and clustering params

Details

Possible data description fields:

string.columns - string features

numerical.columns - numerical features

points - points features

geographic.coordinates - geographic coordinates features

polygons - polygons features

Clustering params description (all fields must be filled):

clusters.number - weight of spatial components' distance (spatial) of the normal components'
distance (non.spatial)

weight.vector - numerical features

spatial - decides whether spatial data considered as spatial data

Example data description is shown below (in the examples)!

Value

cluster model

Examples

```
# structure of data description
example.clustering.description <- list (
  "data.description" = list(
    "class.column" = c("V1"),
    "string.columns" = list(list(col = "V1")),
    "numerical.columns" = list(list(col = "V6"), list(col = "V7"), list(col = "V8"), list(col = "V9")),
    "points" = list(list("x" = list(col = "V2"), "y" = list(col = "V5"))),
    "geographic.coordinates" = list(list("long" = list(col = "V3"), "lat" = list(col = "V4")))
  ),
```



```

"params" = list(
  "clusters.number" = 6,
  "weight.vector" = list("non.spatial" = 1, "spatial" = 14),
  "spatial" = TRUE
)
)

# Mississippi Police Crime Reports data description

police.data.description <- list(
  "numerical.columns" = list(list(col = "TAX"), list(col = "TRANSFER"), list(col = "INC"),
                             list(col = "CRIME"), list(col = "UNEMP"), list(col = "OWN"),
                             list(col = "COLLEGE"), list(col = "WHITE"), list(col = "COMMUTE")),
  "polygons" = list(list(col = "V22"))
)

police.clustering.description <- list (
  "data.description" = police.data.description,

  "params" = list(
    "clusters.number" = 6,
    "weight.vector" = list("non.spatial" = 1, "spatial" = 14),
    "spatial" = TRUE
  )
)

# cluster data
uciData <- read.csv(file="../data/police.csv", head=TRUE, sep=",", skip=0, dec=".")
spatialCluster(uciData, police.clustering.description)

```

stringDistance

Function that calculates scaled distance between two strings.

Description

Function that calculates scaled distance between two strings.

Usage

```
stringDistance(x, y)
```

Arguments

x	First string.
y	Second string.

Value

Scaled distance between two given strings.

Examples

```
stringDistance("mateusz", "mateuss") # result: 0.1428571
```

Index

*Topic **cluster**

spatialCluster, 8

*Topic **distance**

calculateDistanceMatrix, 2

distanceBrayCurtis, 3

distanceEuclidean, 3

distanceHaversine, 4

distanceLevensthein, 5

distancePolygons, 5

distanceReal, 6

objectsDistance, 7

stringDistance, 9

*Topic **spatial**

distanceEuclidean, 3

distanceHaversine, 4

distancePolygons, 5

calculateDistanceMatrix, 2

calculateDistancesFromObject, 2

distanceBrayCurtis, 3

distanceEuclidean, 3

distanceHaversine, 4

distanceLevensthein, 5

distancePolygons, 5

distanceReal, 6

getColumnIndex, 6

getColumnName, 7

objectsDistance, 7

rclusterstudy, 1

rclusterstudy-package (rclusterstudy), 1

spatialCluster, 8

stringDistance, 9