Package 'dspace'

January 5, 2020

Type Package

Title Dividing space - Data driven segementation of spatial data

Version 1.0.0

Author Adam Dabrowski

Maintainer Adam Dabrowski <`adam.dabrowski@amu.edu.pl`>

Description Divides geographical polygon and point data into spatially cohesive regions based on fast greedy community finding algorithm.

License What license is it under?

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

Imports spgwr,dplyr,raster,sf,igraph,spdep,rgdal,reshape2,geosphere,tmap,ggplot2,caret

Suggests testthat

R topics documented:

`accuracy_ds` *accuracy_ds*

Description

Calculates the accuracy measure based on caret's ranger model for the regionalization done by polygon_ds or points_ds

Usage

```
accuracy_ds(x)
```

Arguments

`x`	data frame with class atribute and the data that has been taken into regionalization

Value

the accuracy measure of random forest classification

`build_graph` *build_graph*

Description

Creates fastgreedy community graph for extracting regions

Usage

```
build_graph(x, data, x.nb, method, style)
```

Arguments

`x`	polygon or point data from wich neighbourhood object will be created
`data`	data frame to build weights between polygons/points
`x.nb`	neighbourhood object created by prepare_points() or prepare_polygons()
`method`	method to calculate similarity/distance betweem neighbouring points or polygons
`style`	style of neighbourhood

Value

a list containing community object ("fg") and graph object ("graph")

`find_no_clusters` *find_no_clusters*

Description

Helps identify how many regions should be divided by analyzing the changes in modularity value

Usage

```
find_no_clusters(
  x,
  polygon = T,
  queen = T,
  method = "euclidean",
  data = 2:ncol(x),
  style = "B",
  n.neigh = 8,
  disjoint = F,
  range = 2:30
)
```

Arguments

`x`	spatial data for regionalization
`polygon`	logical if the data is polygon or point
`queen`	if data is polygon and without disjoint polygons, should the neighbourhood be treated by queen topology or rook topology
`method`	Character or function to declare distance method. If method is character, method must be "mahalanobis" or "euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowisk". If method is one of "euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowisk", see dist for details, because this function as used to compute the distance. If method="mahalanobis", the mahalanobis distance is computed between neighbour areas. If method is a function, this function is used to compute the distance.
`data`	data to analyze similarity between regions
`style`	style can take values “W”, “B”, “C”, “U”, “minmax” and “S”
`n.neigh`	number of neighbours considered in the k-nearest neighbour algorithm that builds topology
`disjoint`	if default settings generate error occuring to disjoint subgraphs it means, that in some places points or polygons are to disjoint to generate one connected graph. Use disjoint = T to enforce that one graph will be created. This is a slower option.
`range`	number of divisions to test the modularity. THe biger the numbers, the longer it will take to calculate plot

Value

A vector of modularity measures for given range of divisions

part_communities *part_communities*

Description

Divides graph into defined number of regions

Usage

```
part_communities(k, fg)
```

Arguments

k	number of clusters to create regionalization
fg	hierarchical community object created from build_graph()

Value

a vector of classes - numbers of regions that particular polygon or point are classified to

plot_modularity *plot_modularity*

Description

plots the modularity statistic against the number of clusters.

Usage

```
plot_modularity(mod)
```

Arguments

mod	modularity object

Value

a plot of modularity values calculated for a given range of clusters

points_ds *points_ds*

Description

Creates a vector of community assignment based on neighbouring points. It creates a topological structure in which nodes represent points and the edge is the similarity between nodes. Communities are created using fast greedy algorithm that maximizes their modularity.

Usage

```
points_ds(
  x,
  k = 2,
  queen = T,
  data = 2:ncol(x),
  method = "euclidean",
  style = "B",
  disjoint = F,
  n.neigh = 8,
  plot = T,
  accuracy = T
)
```

Arguments

x	point or polygon shapefile data;
k	number of clusters;
data	atributes of the spatial data frame to calculate similarity or distance measure;
method	Character or function to declare distance method. If method is character, method must be "mahalanobis" or "euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowisk". If method is one of "euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowisk", see dist for details, because this function as used to compute the distance. If method="mahalanobis", the mahalanobis distance is computed between neighbour areas. If method is a function, this function is used to compute the distance.
style	style can take values "W", "B", "C", "U", "minmax" and "S"
disjoint	if default settings generate error occuring to disjoint subgraphs it means, that in some places points or polygons are to disjoint to generate one connected graph. Use disjoint = T to enforce that one graph will be created. This is a slower option.
n.neigh	number of neighbours considered in the k-nearest neighbour algorithm that builds topology
plot	should the neighbourhood be plotted
accuracy	logical should accuracy be calculated based on randomForest algorithm

Value

vector of numbers representing regions to whicheach element

`polygon_ds` *polygon_ds*

Description

Creates a vector of community assignment based on neighbouring polygons. It creates a topological structure in which nodes represent polygons and the edge is the similarity between nodes. Communities are created using fast greedy algorithm that maximizes their modularity.

Usage

```
polygon_ds(
  x,
  k = 2,
  queen = T,
  data = 2:ncol(x),
  method = "euclidean",
  style = "B",
  disjoint = F,
  n.neigh = 8,
  plot = T,
  accuracy = T
)
```

Arguments

`x`	point or polygon shapefile data;
`k`	number of clusters;
`data`	atributes of the spatial data frame to calculate similarity or distance measure;
`method`	Character or function to declare distance method. If method is character, method must be "mahalanobis" or "euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowisk". If method is one of "euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowisk", see dist for details, because this function as used to compute the distance. If method="mahalanobis", the mahalanobis distance is computed between neighbour areas. If method is a function, this function is used to compute the distance.
`style`	style can take values "W", "B", "C", "U", "minmax" and "S"
`disjoint`	if default settings generate error occuring to disjoint subgraphs it means, that in some places points or polygons are to disjoint to generate one connected graph. Use disjoint = T to enforce that one graph will be created. This is a slower option.
`n.neigh`	number of neighbours considered in the k-nearest neighbour algorithm that builds topology
`plot`	should the neighbourhood be plotted
`accuracy`	logical should accuracy be calculated based on randomForest algorithm

Value

vector of numbers representing regions to whicheach element

Examples

```
 data("World",package = "tmap")
 world<- filter(World,!is.na(World$lifeExp))
 modularity<-find_no_clusters(world,data=c(9,10),disjoint = T,plot=T,n.neigh = 6)
 plot_modularity(modularity)
world$class<-polygon_ds(world,data=c(9,10),k=7,style="B",disjoint = T,plot=T,n.neigh = 6)
 qtm(world,"class")
```

`prepare_points` *prepare_points*

Description

Prepares points for regionalization - changes simplea features to SpatailPolygonsDataFrame an calculates neighbourhood objects

Usage

```
prepare_points(x, method = "euclidean", n.neigh = 8, plot = T)
```

Arguments

`x`	point object
`method`	the distance/similarity to calculate
`n.neigh`	at least how many neighbours should be taken into consideration
`plot`	logical if TRUE a plot showing neighbourhoods is beeing presented
`k`	number of clusters

Value

neighbourhoods for coummunity finding

`prepare_polygons` *prepare_polygon*

Description

Prepares polygons for regionalization - changes simplea features to SpatailPolygonsDataFrame an calculates neighbourhood objects

Usage

```
prepare_polygons(x, queen, method, disjoint, n.neigh, plot)
```

Arguments

`x`	point object
`queen`	logical should the wueen or the rook neighbourhood be calculated
`method`	the distance/similarity to calculate
`disjoint`	logical if polygons are not continuous
`n.neigh`	at least how many neighbours should be taken into regionalization if disjoint==T
`plot`	logical if TRUE a plot showing neighbourhoods is beeing presented
`k`	number of clusters

Value

neighbourhoods for coummunity finding

Index