Package ‘TDAmapper’

August 29, 2016

Title Analyze High-Dimensional Data Using Discrete Morse Theory

Version 1.0

Date 2015-05-26

Author Paul Pearson [aut, cre, trl],
Daniel Muellner [aut, ctb],
Gurjeet Singh [aut, ctb]

Maintainer Paul Pearson <pearsonp@hope.edu>

Description Topological Data Analysis using Mapper (discrete Morse theory). Generate a 1-dimensional simplicial complex from a filter function defined on the data: 1. Define a filter function (lens) on the data. 2. Perform clustering within within each level set and generate one node (vertex) for each cluster. 3. For each pair of clusters in adjacent level sets with a nonempty intersection, generate one edge between vertices. The function mapper1D uses a filter function with codomain R, while the the function mapper2D uses a filter function with codomain R^2.

Depends R (>= 3.1.2)

Suggests fastcluster, igraph

License GPL-3

LazyData true

URL https://github.com/paultpearson/TDAmapper/

BugReports https://github.com/paultpearson/TDAmapper/issues

NeedsCompilation no

Repository CRAN

Date/Publication 2015-05-31 09:23:14

R topics documented:

cluster_cutoff_at_first_empty_bin

cluster_cutoff_at_first_empty_bin function

Description

This function decides where to cut the hierarchical clustering tree to define clusters within a level set.

Usage

```
cluster_cutoff_at_first_empty_bin(heights, diam, num_bins_when_clustering)
```

Arguments

`heights` Height values in hierarchical clustering.

`diam` Maximum distance between points in a level set.

`num_bins_when_clustering`
Controls how many bins there are in the histogram used to determine cutoff. values

Value

Numerical value for cutoff point of hierarchical cluster diagram.

Author(s)

Paul Pearson, `<pearsonp@hope.edu>`

References

`https://github.com/paultpearson/TDAmapper`

See Also

`mapper1D`, `mapper2D`

mapper1D *mapper1D function*

Description

This function uses a filter function f: X -> R on a data set X that has n rows (observations) and k columns (variables).

Usage

```
mapper1D(distance_matrix = dist(data.frame(x = 2 * cos(0.5 * (1:100)), y =
  sin(1:100))), filter_values = 2 * cos(0.5 * (1:100)), num_intervals = 10,
  percent_overlap = 50, num_bins_when_clustering = 10)
```

Arguments

`distance_matrix`
An n x n matrix of pairwise dissimilarities.

`filter_values` A length n vector of real numbers.

`num_intervals` A positive integer.

`percent_overlap`
A number between 0 and 100 specifying how much adjacent intervals should overlap.

`num_bins_when_clustering`
A positive integer that controls whether points in the same level set end up in the same cluster.

Value

An object of class `TDAmapper` which is a list of items named `adjacency` (adjacency matrix for the edges), `num_vertices` (integer number of vertices), `level_of_vertex` (vector with `level_of_vertex[i]` = index of the level set for vertex i), `points_in_vertex` (list with `points_in_vertex[[i]]` = vector of indices of points in vertex i), `points_in_level` (list with `points_in_level[[i]]` = vector of indices of points in level set i, and `vertices_in_level` (list with `vertices_in_level[[i]]` = vector of indices of vertices in level set i.

Author(s)

Paul Pearson, <`pearsonp@hope.edu`>

References

https://github.com/paultpearson/TDAmapper

See Also

`mapper2D`

Examples

```
m1 <- mapper1D(
       distance_matrix = dist(data.frame( x=2*cos(0.5*(1:100)), y=sin(1:100) )),
       filter_values = 2*cos(0.5*(1:100)),
       num_intervals = 10,
       percent_overlap = 50,
       num_bins_when_clustering = 10)
## Not run:
#install.packages("igraph")
library(igraph)
g1 <- graph.adjacency(m1$adjacency, mode="undirected")
plot(g1, layout = layout.auto(g1) )

## End(Not run)
```

mapper2D *mapper2D function*

Description

This function uses a filter function f: X -> R^2 on a data set X that has n rows (observations) and k columns (variables).

Usage

```
mapper2D(distance_matrix = dist(data.frame(x = 2 * cos(1:100), y =
  sin(1:100))), filter_values = list(2 * cos(1:100), sin(1:100)),
  num_intervals = c(5, 5), percent_overlap = 50,
  num_bins_when_clustering = 10)
```

Arguments

`distance_matrix`
an n x n matrix of pairwise dissimilarities

`filter_values`
a list of two length n vector of real numbers

`num_intervals`
a vector of two positive integers

`percent_overlap`
a number between 0 and 100 specifying how much adjacent intervals should overlap

`num_bins_when_clustering`
a positive integer that controls whether points in the same level set end up in the same cluster

Value

An object of class TDAmapper which is a list of items named adjacency (adjacency matrix for the edges), num_vertices (integer number of vertices), level_of_vertex (vector with level_of_vertex[i] = index of the level set for vertex i), points_in_vertex (list with points_in_vertex[[i]] = vector of indices of points in vertex i), points_in_level (list with points_in_level[[i]] = vector of indices of points in level set i, and vertices_in_level (list with vertices_in_level[[i]] = vector of indices of vertices in level set i.

Author(s)

Paul Pearson, <pearsonp@hope.edu>

References

https://github.com/paultpearson/TDAmapper

See Also

mapper1D

Examples

```
m2 <- mapper2D(
       distance_matrix = dist(data.frame( x=2*cos(1:100), y=sin(1:100) )),
       filter_values = list( 2*cos(1:100), sin(1:100) ),
       num_intervals = c(5,5),
       percent_overlap = 50,
       num_bins_when_clustering = 10)
## Not run:
library(igraph)
g2 <- graph.adjacency(m2$adjacency, mode="undirected")
plot(g2, layout = layout.auto(g2) )

## End(Not run)
```

Index