

Package ‘klcluster’

November 9, 2018

Type Package

Title (k,l)-clustering of trajectories

Version 0.1.0

Description Clustering of trajectories with k-center, k-median or k-means algorithms. The Frechet distance and dynamic time warping can be used for distance measures. In addition, there is a simplification algorithm implemented.

License What license is it under?

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

Imports Rcpp

LinkingTo Rcpp

Suggests knitr,
rmarkdown,
ggplot2,
data.table,
colorspace,
mapproj,
ggmap,
cowplot

VignetteBuilder knitr

R topics documented:

a55	2
bin_simplify	3
brc	3
calc_dot	4
center_disk	4
coord_wise_mean	4
count_sed	5
count_sed_random	5
df_to_mat	5
discrete_frechet	6
dist_c	6

dtw_dist	7
frechet_distance	7
fsp_centers	8
get_clusters	8
get_cluster_c	9
get_cluster_d	9
get_free_interval_c	9
hello	10
index_shuffle	10
kl_means	10
kl_median	11
kNN_trajectories	12
k_l_cluster	13
mat_to_df	14
p29	14
p94	14
plot_all_clust	15
plot_all_clust_grid	15
plot_all_clust_map	16
plot_centers_map	16
plot_grid_no_points	17
sample_center	17
sed	17
sed_randomized	18
simplify_all	18
simplify_curve	19
single_link	19
squared_dist	20
transform_move	20
Index	21

a55

*Data from Richard Mann***Description**

To add: URL

Usage

a55

Format

An object of class list of length 24.

bin_simplify	<i>simplify curve to given nr data points</i>
--------------	---

Description

The function simplifies a polygonal curve P by an approximation P' . Using a binary search on epsilon under the frechet distance, the desired number of data points is returned.

Usage

```
bin_simplify(path, size, precision)
```

Arguments

path	This is a data.frame containing two columns with information about the original trajectory/path. The columns are named x and y and contain the x and y coordinates of the trajectory.
size	This is the number of data points the simplified path should have
precision	This is the error allowed on the number of data points. This needs to be at least 1

brc	<i>Data from Richard Mann</i>
-----	-------------------------------

Description

To add: URL

Usage

```
brc
```

Format

An object of class list of length 24.

calc_dot	<i>calculate the dot product between two points (a,b) on a plane</i>
----------	--

Description

calculate the dot product between two points (a,b) on a plane

Usage

```
calc_dot(ax, ay, bx, by)
```

Arguments

ax	the x coordinate of point a
ay	the y coordinate of point a
bx	the x coordinate of point b
by	the y coordinate of point b

center_disk	<i>calculate center of disk</i>
-------------	---------------------------------

Description

calculate center of disk

Usage

```
center_disk(p1, p2, p3)
```

Arguments

points	a numeric matrix with an x and y column, filled with points
--------	---

coord_wise_mean	<i>calculate the coordinate wise mean between a set of points on the plane</i>
-----------------	--

Description

calculate the coordinate wise mean between a set of points on the plane

Usage

```
coord_wise_mean(coords, nr_coords)
```

Arguments

coords	a matrix, consisting of 2 columns, one with x coordinates, the second with the y coordinates
nr_coords,	specify the number of coordinates to calculate the mean over. Usefull in case this does not equal the number of points in coords.

count_sed	<i>counter debug</i>
-----------	----------------------

Description

counter debug

Usage

count_sed(points, times)

Arguments

points a numeric matrix with an x and y column, filled with points

count_sed_random	<i>counter debug random</i>
------------------	-----------------------------

Description

counter debug random

Usage

count_sed_random(points, times)

Arguments

points a numeric matrix with an x and y column, filled with points

df_to_mat	<i>transform dataframe to matrix</i>
-----------	--------------------------------------

Description

transform dataframe to matrix

Usage

df_to_mat(x)

discrete_frechet	<i>compute discrete frechet distance</i>
------------------	--

Description

from Thomas Eiter and Heikki Mannila

Usage

```
discrete_frechet(p1, p2)
```

Arguments

p1	data.frame containing x and y coordinates of trajectory
p2	data.frame containing x and y coordinates of trajectory

References

<http://www.kr.tuwien.ac.at/staff/eiter/et-archive/cdtr9464.pdf>

dist_c	<i>calculate euclidean distance between two points (a,b) on a plane</i>
--------	---

Description

calculate euclidean distance between two points (a,b) on a plane

Usage

```
dist_c(ax, ay, bx, by)
```

Arguments

ax	the x coordinate of point a
ay	the y coordinate of point a
bx	the x coordinate of point b
by	the y coordinate of point b

dtw_dist	<i>compute DTW distance between two paths</i>
----------	---

Description

compute DTW distance between two paths

Usage

```
dtw_dist(p1, p2, sqr_dist, normalize_dist)
```

Arguments

p1	This is a data.frame containing two columns with information about the original trajectory/path. The columns are named x and y and contain the x and y coordinates of the trajectory.
p2	This is a data.frame containing two columns with information about the original trajectory/path. The columns are named x and y and contain the x and y coordinates of the trajectory.
sqr_dist	default = FALSE. If dist_measure = 3, then: square_dist = TRUE ensures the dtw uses squared distances square_dist = FALSE ensures the dtw uses normal distances
normalize_dist	default = TRUE. If dist_measure = 3, then: norm_dist = TRUE ensures dtw distances returns a normalized distance i.e. $dtw_dist = dtw_dist / \max(lp1l, lp2l)$ if square_dist = TRUE then $dtw_dist = \sqrt{dtw_dist / \max(lp1l, lp2l)}$ norm_dist = FALSE outputs actual dtw distance

frechet_distance	<i>calculate continuous frechet distance</i>
------------------	--

Description

The function calculates the frechet distance between two paths. The distance has a precision variable that indicates the precision of the outcome.

Usage

```
frechet_distance(p1, p2, precision)
```

Arguments

precision	This is a number containing the precision with which to calculate the frechet distance
path1	This is a data.frame containing two columns with information about the original trajectory/path. The columns are named x and y and contain the x and y coordinates of the trajectory.
path2	This is a data.frame containing two columns with information about the original trajectory/path. The columns are named x and y and contain the x and y coordinates of the trajectory.

Value

A number `apx_eps` which equals the frechet distance. Note that the outcome is not exact. The true epsilon lies in the following range: $\text{apx_eps} - \text{precision} \leq \text{epsilon} \leq \text{apx_eps}$

<code>fsp_centers</code>	<i>fsp centers</i>
--------------------------	--------------------

Description

fsp centers

Usage

```
fsp_centers(paths, clust_out, dist_measure = 3, dtw_settings = c(1,
  TRUE, TRUE))
```

Arguments

<code>paths</code>	This is a list of data.frames. Each data.frame represents a path. Each data.frame contains two columns. The columns are named x and y and contain the x and y coordinates of the trajectory
<code>clust_out</code>	the output of one of the clustering algorithms
<code>dist_measure</code>	input either "cont_frechet", "disc_frechet", "dtw" or 1,2,3 respectively. Specifies the distance measure to be used.
<code>dtw_settings</code>	a vector of 3 variables with settings when using dtw distance <code>c('method_dtw', 'squared_dist', 'normalize_dist')</code> <code>method_dtw</code> : integer 1,2,3,4; specified what center type to use <code>squared_dist</code> : bool; specifies wheter to use squared distances <code>normalize_dist</code> : bool; specifies wheteher to normalize distances to output in cost in the same range as frechet distance

Value

Clustering output containing clusters, centers and result

<code>get_clusters</code>	<i>cluster from distance matrix</i>
---------------------------	-------------------------------------

Description

cluster from distance matrix

Usage

```
get_clusters(dist_m)
```

get_cluster_c	<i>cluster continuous</i>
---------------	---------------------------

Description

cluster continuous

Usage

```
get_cluster_c(centers, paths, prec = 1)
```

get_cluster_d	<i>cluster discrete</i>
---------------	-------------------------

Description

cluster discrete

Usage

```
get_cluster_d(centers, paths)
```

get_free_interval_c	<i>subroutine computing free space diagram cont. frechet</i>
---------------------	--

Description

subroutine computing free space diagram cont. frechet

Usage

```
get_free_interval_c(avec, uvec, vvec, eps)
```

Arguments

avec	numericvector corresponding to one point on the plane avec[0] is the x-coordinate avec[1] is the y-coordinate
uvec	- see avec
vvec	- see avec

hello	<i>Hello, World!</i>
-------	----------------------

Description

Prints 'Hello, world!'.

Usage

```
hello()
```

Examples

```
hello()
```

index_shuffle	<i>Create index permutation</i>
---------------	---------------------------------

Description

Create index permutation

Usage

```
index_shuffle(start, n)
```

Arguments

integer x - number of elements in vectors

kl_means	<i>kl-means</i>
----------	-----------------

Description

The function computes k centers of the given input paths and initial centers, where the centers are simplified to l data points.

Usage

```
kl_means(paths, clust_out, max_iter = 1, dist_measure = 2,
         clusts_to_use = NA, prec_eps = 1e-04, square_dist = FALSE,
         norm_dist = TRUE, update = TRUE)
```

Arguments

paths	This is a list of data.frames. Each data.frame represents a path. Each data.frame contains two columns. The columns are named x and y and contain the x and y coordinates of the trajectory
clust_out	the output of one of the clustering algorithms
max_iter	integer with maximum number of iterations to perform
dist_measure	either input 1, 2 or 3 correspondig to "continuous frechet", "discrete frechet" or "dtw"
clusts_to_use	(optional)vector of integers with indexes of clusters to use in the algorithm#
prec_eps	This is a number containing the precision with which to calculate the frechet distance
square_dist	default = FALSE. If dist_measure = 3, then: square_dist = TRUE ensures the dtw uses squared distances square_dist = FALSE ensures the dtw uses normal distances
norm_dist	default = TRUE. If dist_measure = 3, then: norm_dist = TRUE ensures dtw distances returns a normalized distance i.e. $dtw_dist = dtw_dist / \max(lp1l, lp2l)$ if square_dist = TRUE then $dtw_dist = \sqrt{dtw_dist / \max(lp1l, lp2l)}$ norm_dist = FALSE outputs actual dtw distance
update	if TRUE then all output will be updated

Value

Clustering output containing clusters, centers and result

kl_median	<i>kl-median</i>
-----------	------------------

Description

The function computes k centers of the given input paths and initial centers, where the centers are simplified to l data points.

Usage

```
kl_median(paths, clust_out, max_iter = 1, dist_measure = 2,
  clusts_to_use = NA, prec_eps = 1e-04, square_dist = FALSE,
  norm_dist = TRUE, update = TRUE)
```

Arguments

paths	This is a list of data.frames. Each data.frame represents a path. Each data.frame contains two columns. The columns are named x and y and contain the x and y coordinates of the trajectory
clust_out	the output of one of the clustering algorithms
max_iter	integer with maximum number of iterations to perform
dist_measure	either input 1, 2 or 3 correspondig to "continuous frechet", "discrete frechet" or "dtw"

clusts_to_use	(optional)vector of integers with indexes of clusters to use in the algorithm#'
prec_eps	This is a number containing the precision with which to calculate the frechet distance
square_dist	default = FALSE. If dist_measure = 3, then: square_dist = TRUE ensures the dtw uses squared distances square_dist = FALSE ensures the dtw uses normal distances
norm_dist	default = TRUE. If dist_measure = 3, then: norm_dist = TRUE ensures dtw distances returns a normalized distance i.e. $dtw_dist = dtw_dist / \max(lp1l, lp2l)$ if square_dist = TRUE then $dtw_dist = \sqrt{dtw_dist / \max(lp1l, lp2l)}$ norm_dist = FALSE outputs actual dtw distance
update	if TRUE then all output will be updated

Value

Clustering output containing clusters, centers and result

kNN_trajectories	<i>compute 1-NN clusters</i>
------------------	------------------------------

Description

compute 1-NN clusters

Usage

```
kNN_trajectories(dt, l, prec_l = 1, dist_measure = 2, prec_eps = NA,
  square_dist = FALSE, norm_dist = TRUE)
```

Arguments

dt	list of data frames, each data frame representing a path
l	number of datapoints to return for each center
prec_l	precision on l, i.e. $l - prec_l \leq center \leq l + prec_l$
dist_measure	either input 1, 2 or 3 correspondig to "continuous frechet", "discrete frechet" or "dtw"
prec_eps	if dist_measure = 1 (continuous frechet) then this is the precision of the epsilon approximation default is NA
square_dist	default = FALSE. If dist_measure = 3, then: square_dist = TRUE ensures the dtw uses squared distances square_dist = FALSE ensures the dtw uses normal distances
norm_dist	default = TRUE. If dist_measure = 3, then: norm_dist = TRUE ensures dtw distances returns a normalized distance i.e. $dtw_dist = dtw_dist / \max(lp1l, lp2l)$ if square_dist = TRUE then $dtw_dist = \sqrt{dtw_dist / \max(lp1l, lp2l)}$ norm_dist = FALSE outputs actual dtw distance

k_l_cluster	<i>(k,l) clustering</i>
-------------	-------------------------

Description

The function computes an number of centers of the given input paths, where the centers are simplified to l data points. The number of centers can be predefined when input k is given. However, the algorithm can also determine the centers based on a given radius. When both k and radius are specified, the algorithm returns a number of centers depending on which condition is satisfied first

Usage

```
k_l_cluster(paths, radius = NA, k = NA, l = 20, prec_l = 1,
  dist_measure = 2, prec_eps = 1e-04, method = "minmax",
  square_dist = FALSE, norm_dist = TRUE)
```

Arguments

paths	This is a list of data.frames. Each data.frame represents a path. Each data.frame contains two columns. The columns are named x and y and contain the x and y coordinates of the trajectory
radius	The algorithm computes the number of centers such that each path's distance to one of the centers is smaller than or equal to radius
k	The algorithm computes k centers
l	The number of data points to simplify each center to.
prec_l	This is a number containing the precision of the number of data points to return.
dist_measure	either input 1, 2 or 3 correspondig to "continuous frechet", "discrete frechet" or "dtw"
prec_eps	This is a number containing the precision with which to calculate the frechet distance
method	Either "minmax" or "prob" for choosing centers
square_dist	default = FALSE. If dist_measure = 3, then: square_dist = TRUE ensures the dtw uses squared distances square_dist = FALSE ensures the dtw uses normal distances
norm_dist	default = TRUE. If dist_measure = 3, then: norm_dist = TRUE ensures dtw distances returns a normalized distance i.e. dtw_dist = dtw_dist/max(lp1l, lp2l) if square_dist = TRUE then dtw_dist = sqrt(dtw_dist/max(lp1l, lp2l)) norm_dist = FALSE outputs actual dtw distance

Value

Clustering output containing clusters, centers and result

mat_to_df	<i>transform matrix to dataframe</i>
-----------	--------------------------------------

Description

transform matrix to dataframe

Usage

mat_to_df(mat, path)

p29	<i>Data from Richard Mann</i>
-----	-------------------------------

Description

To add: URL

Usage

p29

Format

An object of class list of length 24.

p94	<i>Data from Richard Mann</i>
-----	-------------------------------

Description

To add: URL

Usage

p94

Format

An object of class list of length 24.

plot_all_clust	<i>plot all clusters without grid</i>
----------------	---------------------------------------

Description

plot all clusters without grid

Usage

```
plot_all_clust(all_paths, out_dt, name_plot = "")
```

Arguments

all_paths	is the list of data.frames with the trajectories
out_dt	contains the clustering output
name_plot	is a string containing the title of the plot

plot_all_clust_grid	<i>plot all clusters with points and grid</i>
---------------------	---

Description

plot all clusters with points and grid

Usage

```
plot_all_clust_grid(all_paths, out_dt, name_plot = "")
```

Arguments

all_paths	is the list of data.frames with the trajectories
out_dt	contains the clustering output
name_plot	is a string containing the title of the plot

plot_all_clust_map	<i>plot all clusters on a map using ggmap and stamenmaps</i>
--------------------	--

Description

plot all clusters on a map using ggmap and stamenmaps

Usage

```
plot_all_clust_map(all_paths, out_dt, name_plot = "", map_type,
  zoom_ggmap = 13)
```

Arguments

all_paths	is the list of data.frames with the trajectories
out_dt	contains the clustering output
name_plot	is a string containing the title of the plot
map_type	contains the map type you want to use, see [stamenmaps](maps.stamen.com)
zoom_ggmap	defines the quality of the map

plot_centers_map	<i>plot all centers on maps</i>
------------------	---------------------------------

Description

plot all centers on maps

Usage

```
plot_centers_map(all_paths, out_dt, name_plot = "",
  map_type = "terrain", zoom_ggmap = 13)
```

Arguments

all_paths	is the list of data.frames with the trajectories
out_dt	contains the clustering output
name_plot	is a string containing the title of the plot
map_type	contains the map type you want to use, see [stamenmaps](maps.stamen.com)
zoom_ggmap	defines the quality of the map

plot_grid_no_points	<i>plot all clusters without dots</i>
---------------------	---------------------------------------

Description

plot all clusters without dots

Usage

```
plot_grid_no_points(all_paths, out_dt, name_plot = "")
```

Arguments

all_paths	is the list of data.frames with the trajectories
out_dt	contains the clustering output
name_plot	is a string containing the title of the plot

sample_center	<i>sample centers</i>
---------------	-----------------------

Description

sample centers

Usage

```
sample_center(path, size)
```

sed	<i>Smallest enclosing disc</i>
-----	--------------------------------

Description

Smallest enclosing disc

Usage

```
sed(points)
```

Arguments

points	a numeric matrix with an x and y column, filled with points
--------	---

sed_randomized	<i>Ramdomized smallest enclosing disk</i>
----------------	---

Description

Randomized smallest enclosing disk

Usage

```
sed_randomized(points)
```

Arguments

points	a numeric matrix with an x and y column, filled with points
--------	---

simplify_all	<i>simplify curves to given nr data points</i>
--------------	--

Description

The function simplifies all polygonal curve P, stored in a list by an approximation P'. Using a binary search on epsilon under the frechet distance, the desired number of data points is returned for each path.

Usage

```
simplify_all(paths, size, precision)
```

Arguments

paths	This is a list of data.frames. Each data.frame contains two columns with information about the original trajectory/path. The columns are named x and y and contain the x and y coordinates of the trajectory.
size	This is the number of data points the simplified path should have
precision	This is the error allowed on the number of data points. This needs to be at least 1

simplify_curve	<i>simplify under frechet</i>
----------------	-------------------------------

Description

The function simplifies a polygonal curve P by an approximation P' subject to an error epsilon under the Frechet distance.

Usage

```
simplify_curve(path, eps)
```

Arguments

path	This is a data.frame containing two columns with information about the original trajectory/path. The columns are named x and y and contain the x and y coordinates of the trajectory.
eps	This is a number representing epsilon which will be input for the Frechet distance.

single_link	<i>single linkage clustering</i>
-------------	----------------------------------

Description

This algorithm builds a hierarchy of clusterings and returns a number of centers given either radius or k

Usage

```
single_link(paths, radius = NA, k = NA, l = 20, prec_l = 1,
  dist_measure = 2, prec_eps = 1e-04, square_dist = FALSE,
  norm_dist = TRUE)
```

Arguments

paths	This is a list of data.frames. Each data.frame represents a path. Each data.frame contains two columns. The columns are named x and y and contain the x and y coordinates of the trajectory
radius	The algorithm computes the number of centers such that each path's distance to one of the centers is smaller than or equal to radius
k	The algorithm computes k centers
l	The number of data points to simplify each center to.
prec_l	This is a number containing the precision of the number of data points to return.
dist_measure	either input 1, 2 or 3 correspondig to "continuous frechet", "discrete frechet" or "dtw"
prec_eps	This is a number containing the precision with which to calculate the frechet distance

square_dist	default = FALSE. If dist_measure = 3, then: square_dist = TRUE ensures the dtw uses squared distances square_dist = FALSE ensures the dtw uses normal distances
norm_dist	default = TRUE. If dist_measure = 3, then: norm_dist = TRUE ensures dtw distances returns a normalized distance i.e. $dtw_dist = dtw_dist / \max(lp1l, lp2l)$ if square_dist = TRUE then $dtw_dist = \sqrt{dtw_dist / \max(lp1l, lp2l)}$ norm_dist = FALSE outputs actual dtw distance

squared_dist	<i>calculate the squared distance between two points (a,b) on a plane</i>
--------------	---

Description

calculate the squared distance between two points (a,b) on a plane

Usage

```
squared_dist(ax, ay, bx, by)
```

Arguments

ax	the x coordinate of point a
ay	the y coordinate of point a
bx	the x coordinate of point b
by	the y coordinate of point b

transform_move	<i>Transform movebank data to input format</i>
----------------	--

Description

this function transforms movebank data format to the right input format for the clustering algorithms

Usage

```
transform_move(move_dt)
```

Arguments

move_dt	move data structure, e.g. "Move", "MoveStack", "MoveBurst"
---------	--

Index

*Topic **datasets**

a55, [2](#)
brc, [3](#)
p29, [14](#)
p94, [14](#)

a55, [2](#)

bin_simplify, [3](#)
brc, [3](#)

calc_dot, [4](#)
center_disk, [4](#)
coord_wise_mean, [4](#)
count_sed, [5](#)
count_sed_random, [5](#)

df_to_mat, [5](#)
discrete_frechet, [6](#)
dist_c, [6](#)
dtw_dist, [7](#)

frechet_distance, [7](#)
fsp_centers, [8](#)

get_cluster_c, [9](#)
get_cluster_d, [9](#)
get_clusters, [8](#)
get_free_interval_c, [9](#)

hello, [10](#)

index_shuffle, [10](#)

k_l_cluster, [13](#)
kl_means, [10](#)
kl_median, [11](#)
kNN_trajectories, [12](#)

mat_to_df, [14](#)

p29, [14](#)
p94, [14](#)
plot_all_clust, [15](#)
plot_all_clust_grid, [15](#)
plot_all_clust_map, [16](#)

plot_centers_map, [16](#)
plot_grid_no_points, [17](#)

sample_center, [17](#)
sed, [17](#)
sed_randomized, [18](#)
simplify_all, [18](#)
simplify_curve, [19](#)
single_link, [19](#)
squared_dist, [20](#)

transform_move, [20](#)