Package 'mstclustering'

May 9, 2026

Title ``MST-Based Clustering"

Version 1.0.0.0

Author Kevin Michael Frick <kmfrick98@gmail.com>

Maintainer Kevin Michael Frick <kmfrick98@gmail.com>

Description
Implements a minimum-spanning-tree-based heuristic for k-means clustering using a union-find disjoint set and the algorithm in Kruskal (1956) <doi:10.1090/S0002-9939-1956-0078686-7>.

License AGPL (>= 3)

Encoding UTF-8

RoxygenNote 7.1.2

Depends R (>= 4.1.0)

Imports reshape2, data.table

NeedsCompilation no

Repository CRAN

Date/Publication 2022-02-08 08:40:02 UTC

Contents

find.set *find.set*

Description

Find the set an element belongs to.

Usage

```
find.set(i, ufds)
```

Arguments

i	The element to check.
ufds	A data.table representing a UFDS.

Value

An integer: the root node of the set the element belongs to.

gen.child.list.mst *gen.child.list.mst*

Description

Generate an adjacency list

Usage

```
gen.child.list.mst(clust.edge.list, m)
```

Arguments

clust.edge.list	The return value of the kruskal() function.
m	Number of nodes.

Value

An adjacency list in the form of a list of vectors.

`gen.edge.list` *gen.edge.list*

Description

Generate edge list from a distance matrix Duplicates are not deleted, because they will not be counted by Kruskal's algorithm If a check is O(1), this only adds an O(E) overhead, which is negligible

Usage

```
gen.edge.list(mat)
```

Arguments

`mat`	The distance matrix.

Value

A data frame with three columns: 'from', 'to', 'dist'.

`is.same.set` *is.same.set*

Description

Check if two elements are in the same set

Usage

```
is.same.set(i, j, ufds)
```

Arguments

`i`	The first element in the tuple.
`j`	The second element in the tuple.
`ufds`	A data.table representing a UFDS.

Value

TRUE if the elements are in the same set, FALSE otherwise.

kruskal *kruskal*

Description

Kruskal's algorithm for MST computation.

Usage

```
kruskal(edge.list, m)
```

Arguments

edge.list	A data frame with columnns 'from', 'to', 'dist'.
m	Number of nodes.

Value

A list of edges in the MST, in the same format as the input argument edge.list.

mst.cluster *mst.cluster*

Description

Run clustering using MST. Before calling this function, remove some edges from the MST, for example the k-1 heaviest.

Usage

```
mst.cluster(child.list.mst, m, k)
```

Arguments

child.list.mst	The return value of the gen.child.list.mst() function with k-1 edges removed.
m	Number of nodes.
k	The number of clusters.

Value

A vector whose k-th element is the cluster the k-th point belongs to.

Examples

```
iris.clean <- iris[,-5]
iris.dist <- as.matrix(dist(iris.clean))
iris.edge.list <- gen.edge.list(iris.dist)
m <- nrow(iris.dist)
iris.mst.edge.list <- kruskal(iris.edge.list, m)
k <- 3
n.edges <- nrow(iris.mst.edge.list)
iris.mst.edge.list <- iris.mst.edge.list[1:(n.edges - (k - 1)),]
iris.child.list.mst <- gen.child.list.mst(iris.mst.edge.list, m)
iris.clust.mst <- mst.cluster(iris.child.list.mst, m, k)
```

---

reset.ufds *reset.ufds*

Description

Initialize UFDS

Usage

```
reset.ufds(m)
```

Arguments

m	Number of elements.

Value

A data table containing a 'rank' column and a 'parent' column.

---

union.set *union.set*

Description

Join the sets the two elements passed as arguments belong to.

Usage

```
union.set(i, j, ufds)
```

Arguments

i	The first element in the tuple.
j	The second element in the tuple.
ufds	A data.table representing a UFDS.

Value

No return value, called for side effects on rank and p.

Index