Package 'tbart'

January 27, 2015

Type Package

Title Teitz and Bart p-median algorithm

Version 0.9.1

Date 2014-08-15

Author Chris Brunsdon

Maintainer Chris Brunsdon <christopher.brunsdon@nuim.ie>

Description Solves Teitz and Bart's p-median problem - given a set of points attempts to find subset of size p such that summed distances of any point in the set to the nearest point in p is minimised. Although generally effective, this algorithm does not guarantee that a globally optimal subset is found.

License GPL (>= 2)

Depends Rcpp (>= 0.10.3), sp

Suggests GISTools, RColorBrewer

LinkingTo Rcpp

Collate 'RcppExports.R' 'tbmain.R' 'tbart-describes.R'

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-08-16 17:31:10

R topics documented:

tbart-package 2
allocate 2
euc.dists 3
star.diagram 4
tb 5
tb.raw 5

Index **7**

`tbart-package` *Teitz and Bart's p-median problem with Spatial* and Spatial*DataFrame objects*

Package: & tbart
Type: & Package
Version: & 0.9
Date: & 2013-06-13
License: & GPL (>= 2)
Maintainer: & Chris Brunsdon mailto:brunsdon@liverpool.ac.uk

Description

Solves Teitz and Bart's p-median problem - given a set of points attempts to find subset of size p such that summed distances of any point in the set to the nearest point in p is minimised. Although generally effective, this algorithm does not guarantee that a globally optimal subset is found.

Author(s)

Chris Brunsdon

References

Teitz, M. B., and P. Bart (1968), Heuristic methods for estimating generalized vertex median of a weighted graph, Operations Research, 16, 955-961.

`allocate` *Teitz-Bart algorithm applied to Spatial* and Spatial*DataFrame objects - report allocations*

Description

Teitz-Bart algorithm applied to Spatial* and Spatial*DataFrame objects - report allocations

Usage

```
allocate(swdf1, swdf2, p, metric, verbose = FALSE)
```

Arguments

swdf1	- first Spatial* or Spatial*DataFrame objects
swdf2	- second Spatial* or Spatial*DataFrame objects (if omitted, defaults to the same value as swdf1)
p	- either a guess at the initial p-median set of a single integer indicating the size of the set (which is then chosen randomly)
metric	- the distance matrix (defaults to Euclidean computed via euc.dists(swdf1,swdf2) if not supplied)
verbose	- if TRUE print out each swap in the algorithm (default is FALSE)

Value

List of nearest neigbour indices for each element from the p-median set

Examples

```
data(meuse)
coordinates(meuse) <- ~x+y
allocate(meuse,p=5)



require(RColorBrewer)
require(GISTools)
data(georgia)
allocations.list <- allocate(georgia2,p=5)
zones <- gUnaryUnion(georgia2,allocations.list)
plot(zones,col=brewer.pal(5,"Accent"))
plot(georgia2,border=rgb(0,0,0,0.1),add=TRUE)
points(coordinates(georgia2)[allocations.list,],pch=16,cex=2,col=rgb(1,0.5,0.5,0.1))
```

---

euc.dists *Euclidean distances from a Spatial* or Spatial*DataFrame object*

Description

Euclidean distances from a Spatial* or Spatial*DataFrame object

Usage

```
euc.dists(swdf1, swdf2, scale)
```

Arguments

swdf1	- First Spatial*DataFrame object
swdf2	- Second Spatial*DataFrame object (if omitted, defaults to the same value as swdf1)
scale	- allows re-scaling eg: value of 1000 means distances in km if coordinates of swdf1/swdf2 in meters.

Value

Distance matrix (if swdf1 or swdf2 not SpatialPoints*, distances are based on points obtained from coordinates function)

Examples

```
data(meuse)
coordinates(meuse) <- ~x+y
euc.dists(meuse,scale=1000)
```

---

star.diagram *Creates the lines for a 'star diagram'*

Description

Creates the lines for a 'star diagram'

Usage

```
star.diagram(swdf1, swdf2, alloc)
```

Arguments

swdf1	- first Spatial* or Spatial*DataFrame objects
swdf2	- second Spatial* or Spatial*DataFrame objects (if omitted, defaults to the same value as swdf1)
alloc	- a list saying which coordinate in swdf2 is allocated to each point in swdf1

Examples

```
data(meuse)
coordinates(meuse) <- ~x+y
allocations.list <- allocate(meuse,p=5)
star.lines <- star.diagram(meuse,alloc=allocations.list)
plot(star.lines)
```

tb — *Teitz-Bart algorithm applied to Spatial* and Spatial*DataFrame objects - report p-median set*

Description

Teitz-Bart algorithm applied to Spatial* and Spatial*DataFrame objects - report p-median set

Usage

```
tb(swdf1, swdf2, p, metric, verbose = FALSE)
```

Arguments

swdf1	- first Spatial* or Spatial*DataFrame objects
swdf2	- second Spatial* or Spatial*DataFrame objects (if omitted, defaults to the same value as swdf1)
p	- either a guess at the initial p-median set of a single integer indicating the size of the set (which is then chosen randomly)
metric	- the distance matrix (defaults to Euclidean computed via euc.dists(swdf1,swdf2) if not supplied)
verbose	- if TRUE print out each swap in the algorithm (default is FALSE)

Value

Set of point indices for p-median (may be local optimum)

Examples

```
data(meuse)
coordinates(meuse) <- ~x+y
tb(meuse,p=5)
```

tb.raw — *Teitz-Bart algorithm applied to a 'raw' distance matrix*

Description

Teitz-Bart algorithm applied to a 'raw' distance matrix

Usage

```
tb.raw(d, guess, verbose = FALSE)
```

Arguments

d	- A distance matrix (not necessarily Euclidean)
guess	- a guess at the set of p points constituting the p-median
verbose	- if TRUE print out each swap in the algorithm (default is FALSE)

Value

Set of point indices for p-median (may be local optimum)

Examples

```
x1 <- rnorm(100)
y1 <- rnorm(100)
d <- as.matrix(dist(cbind(x1,y1)))
tb.raw(d,c(1,2))
```

Index

allocate, 2

euc.dists, 3

star.diagram, 4

tb, 5
tb.raw, 5
tbart-package, 2