Package 'SGCS'

March 25, 2019

Type Package

Title Spatial Graph Based Clustering Summaries for Spatial Point Patterns

Version 2.7

Date 2019-03-25

Author Tuomas Rajala

Maintainer Tuomas Rajala <tuomas.rajala@iki.fi>

Depends spatstat

Description Graph based clustering summaries for spatial point patterns. Includes Connectivity function, Cumulative connectivity function and clustering function, plus the triangle/triplet intensity function T.

License GPL (>= 2)

RoxygenNote 6.1.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-03-25 18:53:32 UTC

R topics documented:

`arcs` *Compute the boundary of disks*

Description

Compute the boundary of disks

Usage

```
arcs(x, r)
```

Arguments

`x`	point pattern
`r`	radius of the disks

`bounding_box_xy` *Compute the bounding window from coordinates*

Description

Compute the bounding window from coordinates

Usage

```
bounding_box_xy(x)
```

Arguments

`x`	Columnwise matrix of coordinates, n-dimensional.

Details

Rectangular or cuboidal bounding box around coordinates given as matrix 'x'. Enlarged with the Ripley and Rasson 1977 method.

clustfun *Clustering function*

Description

Summarise the amount of local connectivity in a stationary and isotropic point pattern (2/3d).

Usage

```
clustfun(x, r, correction = "border", scaled = FALSE, ...)
```

Arguments

x	Point pattern
r	Vector of distances
correction	Border correction. "none" or "border" (reduced window) supported.
scaled	Scale with theoretical value? Should be done afterwards.
...	ignored.

Details

This is a generalisation of the clustering coefficient by Watts and Strogatz 1998.

Reduced border correction available.

Value

fv-object, see spatstat for more.

Examples

```
## Not run:
en <- envelope(rcell(nx=15), fun=clustfun)
plot(en)

## End(Not run)
```

clustfun_denominator *Estimate the denominator of clustering function*

Description

Estimate the denominator of clustering function

Usage

```
clustfun_denominator(x, r, correction = "best", ...)
```

Arguments

x	Point pattern
r	Vector of distances to estimate the function
correction	Border correction. Either "border", "none" or "best"(="border).
...	Ignored.

Details

Support in 3D therefore only for cuboidal windows. (inc. rotated)

Value

fv-object.

confun *Connectivity function and cumulative connectivity function.*

Description

Amount of network connected pairs as a function of distance.

Usage

```
confun(x, r, R, h, adjust = 1, preGraph = NULL, ...)
```

Arguments

x	Point pattern
r	Vector of distances to estimate the function
R	The radius for generating the network, as geometric graph.
h	Smoothing parameter. h=0 and h>0 mean different things, see Details.
adjust	Adjust h by this factor (default=1).
preGraph	Precomputed network/graph, as a spatgraph-object. Alternative to R.
...	ignored.

Details

If h=0 we compute the cumulative version of the connecitivity function, corresponding to Ripley's K-function under the condition that the points in each pair must belong to the same component in an underlying network.

The underlying network can be given, or it will be computed as a geometric graph with parameter 'R'. If given as 'preGraph', it must be a **spatgraphs**-object, with same dimensions as the point pattern.

If h>0: Compute the probability of a pair being in the same component given their distance is ~ r. Uses kernel smoothing with bandwidth h.

Sensible defaults are computed for h and R if not given.

Border correction is done via translation correction. The bias is unknown as the network censoring is quite complex.

Theoretical values are unknown due to the graph conditioning.

Value

fv-object, see spatstat for more. Theoretical values unknown.

Examples

```
## Not run:
x <- rMatClust(10, 0.1, 10)
plot(Cx<-confun(x,h=0, R=0.1))

# fit wrong model
ftho <- thomas.estpcf(x)
yf <- function()rThomas(ftho$par[1], ftho$par[2], x$n/ftho$par[1])
CC <- envelope(x, fun=confun, h=0, sim=yf, R=0.1)
C <- envelope(x, fun=confun, sim=yf, R=0.1)

plot(CC)
plot(C)

## End(Not run)
```

---

default_r *Default range vector*

Description

Default range vector

Usage

```
default_r(x, r)
```

Arguments

x	point pattern, internal format
r	most likely missing. if not , returned. Compute the range vector

---

edge_distance *Distances to observation window edge*

Description

Compute the distances from points to observation window edge.

Usage

```
edge_distance(x)
```

Arguments

x	Point pattern Uses bdist.points for ppp objects, otherwise relies on a cuboidal/rectangular window.

---

internalise_pp *Convert input to SGCS internal point pattern data*

Description

Convert input to SGCS internal point pattern data

Usage

```
internalise_pp(x)
```

Arguments

x	some form of point pattern data.

Details

Understands 'ppp', 'pp3', 'matrix' and list(coordinate_matrix, bbox).

Data.frame marks are not well understood, so convert the mark data to suit the analysis before use.

Mainly for internal use.

internal_to_ppp *Data to spatstat format*

Description

Data to spatstat format

Usage

```
internal_to_ppp(x)
```

Arguments

x	Pattern in internal format

Kfun *Ripley's K-function*

Description

See Kest for more rigorous implementation in 2D.

Usage

```
Kfun(x, r, ...)
```

Arguments

x	Point pattern
r	Vector of distances to estimate the function
...	ignored.

Details

Border correction is translation correction. Support in 3D therefore only for cuboidal windows.

Value

fv-object.

morphoArea *Morphologicals: Relative area fraction of diluted pattern*

Description

Dilute pattern with b(o,r) and compute area.

Usage

```
morphoArea(x, r, ...)
```

Arguments

x	Point pattern
r	Vector of distances to estimate the function
...	Ignored.

Value

The default plotted curve, "rAF", is relative to lambda * pi * r^2. The component "AF" holds the area fraction.

morphoEuler *Morphologicals: Euler number of dilated pattern*

Description

Dilate pattern with b(o,r) and compute the Euler number.

Usage

```
morphoEuler(x, r, ...)
```

Arguments

x	Point pattern
r	Vector of distances to estimate the function
...	Ignored.

Value

Only for 2D.

Reduced sample border correction.

morphoLength *Morphologicals: Relative boundary length of diluted pattern*

Description

Dilute pattern with b(o,r) and compute boundary length.

Usage

```
morphoLength(x, r, ...)
```

Arguments

x	Point pattern
r	Vector of distances to estimate the function
...	Ignored.

Value

Only for 2D.

The default plotted curve, "u(r)" is U(r)/(2 * r * lambda * pi) with U(r) the boundary length fraction.

Reduced sample border correction.

pairwise_distances *pairwise distances*

Description

pairwise distances

Usage

```
pairwise_distances(x, toroidal = FALSE)
```

Arguments

x	Point pattern data in some format.
toroidal	For toroidal distances.

Value

Value is the lower triangle of the distasnce matrix, returned as a vector.

Rfun *Clustering function versio 2*

Description

Ratio of triplets to pairs^2, motivated by T/K^2.

Usage

```
Rfun(x, r, correction = "border", scaled = FALSE, ...)
```

Arguments

x	Point pattern
r	Vector of distances
correction	Border correction. "none" or "border" (reduced window) supported.
scaled	Scale with theoretical value?
...	ignored.

Details

Reduced border correction available.

Value

fv-object, see spatstat for more.

Examples

```
## Not run:
en <- envelope(rcell(nx=15), fun=Rfun)
plot(en)

## End(Not run)
```

SGCS *SGCS: Spatial Graph based Clustering Summaries*

Description

Some functions for analysing clusters and clustering of spatial point patterns.

Details

See package **spatstat** for more details on point pattern analysis, especially Section II: Exploratory Data Analysis.

This package provides functional summaries for analysing stationary and isotropic point patterns in 2D and 3D.

`confun` Connectivity function, smoothed and cumulative

`clustfun` Clustering function, generalisation of clustering coefficient

`Tfun` Triangle/triplet intensity function

The Ripley's K-function is there as well as spatstat does not support 3D.

The main source for the definitions of connectivity and clustering functions is **Rajala: Spatial clustering and graph based statistical features, JYU preprints, 2010**

`Tfun` *Triplet intensity function*

Description

Summarise the number of r-close triangles in a stationary and isotropic point pattern (2d,3d).

Usage

```
Tfun(x, r, ...)
```

Arguments

`x`	Point pattern
`r`	Vector of distances to estimate the function
`...`	ignored.

Details

Border correction is done via minus sampling.

See

Schladitz, Baddeley: A Third order point process characteristic, SJS, vol 27, 657-671, 2000.

for details.

Value

`fv`-object.

translation_weights *Translation weights Compute the translation edge correction weights*

Description

Translation weights Compute the translation edge correction weights

Usage

```
translation_weights(x)
```

Arguments

x	Some form of point pattern data.

Index