Package 'globalKinhom'

February 26, 2022

Type Package

Title Inhomogeneous K- And Pair Correlation Functions Using Global Estimators

Version 0.1.4

Date 2022-02-25

Encoding UTF-8

Author Thomas Shaw [aut, cre],
Ege Rubak [ctb],
Adrian Baddeley [ctb],
Rolf Turner [ctb]

Maintainer Thomas Shaw <shawtr@umich.edu>

Depends R (>= 3.5.0), spatstat.random (>= 2.1.0), spatstat.core (>= 2.4.0), spatstat.geom (>= 2.3.2)

Imports stats, utils, grDevices

Description Second-order summary statistics K- and pair-correlation functions describe interactions in point pattern data. This package provides computations to estimate those statistics on inhomogeneous point processes, using the methods of in T Shaw, J Møller, R Waagepetersen, 2020 <arXiv:2004.00527>.

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Date/Publication 2022-02-25 23:40:05 UTC

R topics documented:

`globalKinhom-package` *Inhomogeneous K- And Pair Correlation Functions Using Global Estimators*

Description

Second-order summary statistics K- and pair-correlation functions describe interactions in point pattern data. This package provides computations to estimate those statistics on inhomogeneous point processes, using the methods of in T Shaw, J Møller, R Waagepetersen, 2020 <arXiv:2004.00527>.

Details

The DESCRIPTION file:

Package:	globalKinhom
Type:	Package
Title:	Inhomogeneous K- And Pair Correlation Functions Using Global Estimators
Version:	0.1.4
Date:	2022-02-25
Encoding:	UTF-8
Authors@R:	c(person("Thomas", "Shaw", role=c("aut", "cre"), email="shawtr@umich.edu"), person("Ege", "Rubak", role=
Author:	Thomas Shaw [aut, cre], Ege Rubak [ctb], Adrian Baddeley [ctb], Rolf Turner [ctb]
Maintainer:	Thomas Shaw <shawtr@umich.edu>
Depends:	R (>= 3.5.0), spatstat.random (>= 2.1.0), spatstat.core (>= 2.4.0), spatstat.geom (>= 2.3.2)
Imports:	stats, utils, grDevices
Description:	Second-order summary statistics K- and pair-correlation functions describe interactions in point pattern data. T
License:	GPL (>= 2)

Index of help topics:

`Kglobal`	`(cross) K functions with a global intensity reweighting`
`expectedPairs`	`Expected pairs in an inhomogeneous poisson process`
`globalKinhom-package`	`Inhomogeneous K- And Pair Correlation Functions Using Global Estimators`
`pcfglobal`	`(cross) pair correlation functions with a global intensity reweighting`

This package accompanies Shaw et al (2020). It provides "global" estimators for the non-parametric K- and pair correlation functions, which summarize the second order interactions of second-order intensity-reweighted stationary (SOIRS) point processes. These estimators provide an alternative to those proposed by Baddeley et al (2000) for SOIRS point processes, which we refer to as "local" estimators. The local estimators are implemented in the `spatstat.core` package as `pcfinhom` and `Kinhom`, with `pcfcross.inhom` and `Kcross.inhom` for the corresponding cross-pcf and cross-K-function.

Where possible, the interfaces are made to match those used by the spatstat.core package.

Author(s)

Thomas Shaw [aut, cre], Ege Rubak [ctb], Adrian Baddeley [ctb], Rolf Turner [ctb]

Maintainer: Thomas Shaw <shawtr@umich.edu>

References

T Shaw, J Møller, R Waagepetersen. 2020. "Globally Intensity-Reweighted Estimators for K- and pair correlation functions". arXiv:2004.00527 [stat.ME].

A Baddeley, J Møller, R Waagepetersen. 2000. "Non- and Semi-Parametric Estimation of Interaction in Inhomogeneous Point Patterns". Statistica Neerlandica 54, 329-350.

See Also

spatstat.core, Kglobal, link{pcfglobal}

`expectedPairs` *Expected pairs in an inhomogeneous poisson process*

Description

Compute the expected number of pairs at a given displacement h in a poisson process with a given intensity function. This corresponds to the integrals γ of Shaw et al. 2020. The various functions correspond to the univariate and bivariate versions of the anisotropic or isotropic versions of γ. The final two options (`expectedPairs_kernloo` and `expectedPairs_iso_kernloo`), provide implementations of the leave-out kernel estimates of γ: $\bar{\gamma}(h)$ and $\bar{\gamma}^{\mathrm{iso}}(r)$. In those cases, the point pattern `X` itself is passed to the routine, rather than the (true or estimated) intensities `rho` etc. The estimators for $\bar{\gamma}(h)$ are only applicable to univariate processes. See Shaw et al, 2020 for details.

Usage

```
expectedPairs(rho, hx, hy=NULL, method=c("mc", "lattice"),
                tol=.005, dx=diff(as.owin(rho)$xrange)/200, maxeval=1e6,
                maxsamp=5e3)

expectedCrossPairs(rho1, rho2=NULL, hx, hy=NULL, method=c("mc", "lattice"),
                tol=.005, dx=diff(as.owin(rho1)$xrange)/200, maxeval=1e6,
                maxsamp=5e3)

expectedPairs_iso(rho, r, tol=.001, maxeval=1e6, maxsamp=5e3)

expectedCrossPairs_iso(rho1, rho2=NULL, r, tol=.001, maxeval=1e6, maxsamp=5e3)

expectedPairs_kernloo(X, hx,hy, sigma=bw.CvL, tol=.005, maxeval=1e6,
                            maxsamp=5e3, leaveoneout=TRUE)
```

```
expectedPairs_iso_kernloo(X, r, sigma=bw.CvL, tol=.001, maxeval=1e6,
                                    maxsamp=5e3, leaveoneout=TRUE)
```

Arguments

rho1, rho2, rho	Intensity functions, either of class im or funxy. This may be produced by density.ppp or densityfun.ppp, or provided by a fitted intensity model.
X	Point pattern of class ppp with the points of the pattern for which $\bar{\gamma}$ is to be estimated.
hx, hy	For expectedPairs and expectedCrossPairs (i.e. $\gamma(h)$), the displacements $h \in \mathrm{R}^2$ to evaluate γ at. These can be in any format supported by xy.coords.
r	For the isotropic versions $\gamma^{\mathrm{iso}}(r)$, the separations r at which γ^{iso} is to be evaluated.
method	Either mc (the default) or lattice. Compute integral using monte-carlo or on a lattice.
tol	A tolerance for how precise the integral should be. This is compared to a standard error for the mc estimate.
sigma	Smoothing bandwidth for direct kernel-based estimators $\bar{\gamma}$.
leaveoneout	Use leave-out estimators. This should generally be true except for the purpose of evaluating the bias of the standard estimators. See Shaw et al 2020 for details.
maxeval	Maximum number of evaluations of rho per iteration. Prevents memory-related crashes that can occur.
maxsamp	Maximum number of monte carlo samples per iteration. If this is too large, you may do more work than required to achieve tol.
dx	if method=="lattice", a lattice spacing for the computation. defaults to .01.

Value

The return value is a numeric vector with length equal to the number of displacements h passed

Author(s)

Thomas Shaw <shawtr@umich.edu>

References

T Shaw, J Møller, R Waagepetersen. 2020. "Globally Intensity-Reweighted Estimators for K- and pair correlation functions". arXiv:2004.00527 [stat.ME].

See Also

pcfglobal, Kglobal, which use these functions to compute the normalization functions γ.

`Kglobal` *(cross) K functions with a global intensity reweighting*

Description

Compute K_{global}

Usage

```
Kglobal(X, lambda=NULL, ..., sigma=bw.CvL(X), r=NULL, rmax=NULL, breaks=NULL,
            normtol=.005, discrete.lambda=FALSE,
            interpolate=TRUE, interpolate.fac=10, isotropic=TRUE,
            leaveoneout=TRUE, exp_prs=NULL,
            interpolate.maxdx=diameter(as.owin(X))/100, dump=FALSE)

Kcross.global(X, Y, lambdaX=NULL, lambdaY=NULL, ..., sigma=bw.CvL(X), r=NULL,
            rmax=NULL, breaks=NULL, normtol=.005,
            discrete.lambda=FALSE, interpolate=TRUE, isotropic=TRUE,
            interpolate.fac=10, leaveoneout=TRUE, exp_prs=NULL,
            interpolate.maxdx=diameter(as.owin(X))/100, dump=FALSE)
```

Arguments

`X, Y`	point process of type ppp, on which to evaluate the (cross) K-function
`lambda, lambdaX, lambdaY`	intensity function estimates corresponding to `X` and `Y`. If omitted, intensity functions will be computed using `density.ppp` or `densityfun.ppp` (see discrete.lambda below)
`...`	extra args passed to density.ppp or densityfun.ppp, if applicable.
`sigma`	Bandwidth value to use for kernel-based intensity estimation, intensity functions and `exp_prs` are not provided by the user.
`r`	Values of r to evaluate $K(r)$ at. If omitted, a sensible default is chosen, using the same conventions as `Kest` and `Kinhom`.
`rmax`	Maximum r to evaluate $K(r)$ at. `rmax` is used to generate values for `r`, if omitted. If missing, a sensible default is chosen.
`breaks`	For internal use only.
`normtol`	A tolerance to use for expectedPairs or expectedCrossPairs when computing monte-carlo estimates of the normalizing factor γ. Expressed as a maximum fractional standard error.
`discrete.lambda`	If `TRUE`, and intensity function(s) are not supplied, estimate intensities by interpolating the values on a discrete lattice (using `interp.im` and `density.ppp`), instead of exactly (using `densityfun.ppp`).
`interpolate`	If `TRUE`, evaluate the expectedCrossPairs on a lattice and interpolate, rather than at the exact displacements observed in the pattern.

`interpolate.fac`
: If `interpolate`, the lattice spacing will be `sigma/interpolate.fac`.

`isotropic`
: Set to `TRUE` to use the isotropic estimators γ_{iso}.

`leaveoneout`
: Use the leave-one-out estimator for γ. See Shaw et al, 2020 for details.

`exp_prs`
: A function that returns values for $\gamma_{\text{iso}}(r)$. If γ is known explicitly, or the same calculation is being used for several point patterns, it can be much faster to compute it once and provide the function as `exp_prs`, since the computation of γ is usually the slowest part.

`interpolate.maxdx`
: Upper bound on allowable lattice spacing for interpolation.

`dump`
: For debugging purposes, include computed values of γ with the output, as `attrs`.

Value

The return value is an object of class `fv`, just as for `Kest` and `Kinhom`. The object contains columns `r`, `theo`, and `global`, corresponding respectively to the argument r, the theoretical values of $K(r)$ for a Poisson process, and $K_{\text{global}}(r)$.

Author(s)

Thomas Shaw <shawtr@umich.edu>

References

T Shaw, J Møller, R Waagepetersen. 2020. "Globally Intensity-Reweighted Estimators for K- and pair correlation functions". arXiv:2004.00527 [stat.ME].

See Also

`expectedPairs`

Examples

```
rho <- funxy(function(x,y) 80*(1+x), owin())
X <- rpoispp(rho)
K <- Kglobal(X)
#plot(K)
```

`pcfglobal` *(cross) pair correlation functions with a global intensity reweighting*

Description

Compute g_{global} or c_{global}

Usage

```
pcfglobal(X, lambda=NULL, ..., sigma=bw.CvL(X), r=NULL, rmax=NULL,
    kernel="epanechnikov", bw=NULL, stoyan=0.15, normtol=.005, ratio=FALSE,
    discrete.lambda=FALSE, divisor=c("r", "d"),
    leaveoneout=TRUE, interpolate=TRUE, interpolate.fac=10, exp_prs=NULL,
    interpolate.maxdx=diameter(as.owin(X))/100, dump=FALSE)

pcfcross.global(X,Y, lambdaX=NULL, lambdaY=NULL, ...,
    sigma=bw.CvL(X), r=NULL, rmax=NULL, kernel="epanechnikov", bw=NULL,
    stoyan=0.15, normtol=.005, ratio=FALSE, discrete.lambda=FALSE,
    divisor=c("r", "d"), analytical=NULL, interpolate=TRUE,
    interpolate.fac=10, exp_prs=NULL,
    interpolate.maxdx=diameter(as.owin(X))/100, dump=FALSE)
```

Arguments

X, Y	point process of type ppp, on which to evaluate the (cross) K-function
lambda, lambdaX, lambdaY	intensity function estimates corresponding to X and Y. If omitted, intensity functions will be computed using density.ppp or densityfun.ppp (see discrete.lambda below)
...	extra args passed to density.ppp or densityfun.ppp, if applicable.
sigma	Bandwidth value to use for kernel-based intensity estimation, intensity functions and exp_prs are not provided by the user.
r	Values of r to evaluate $K(r)$ at. If omitted, a sensible default is chosen, using the same conventions as Kest and Kinhom.
rmax	Maximum r to evaluate $K(r)$ at. rmax is used to generate values for r, if omitted. If missing, a sensible default is chosen.
kernel	Kernel type for smoothing of pcf.
bw	Kernel bandwidth for smoothing of pcf.
stoyan	Coefficient for Stoyan's bandwidth selection rule. See pcf.ppp.
normtol	A tolerance to use for expectedPairs or expectedCrossPairs when computing monte-carlo estimates of the normalizing factor γ. Expressed as a maximum fractional standard error.
ratio	If TRUE, assemble numerator and denominator of pcf estimator separately.
divisor	Whether to use the evaluation distance ("r") or the distance between points ("d") to normalize the contribution of each point pair.
analytical	If TRUE, use Diggle-Jones weights
discrete.lambda	If TRUE, and intensity function(s) are not supplied, estimate intensities by interpolating the values on a discrete lattice (using interp.im and density.ppp), instead of exactly (using densityfun.ppp).
interpolate	If TRUE, evaluate the expectedCrossPairs on a lattice and interpolate, rather than at the exact displacements observed in the pattern.

`interpolate.fac`
: If `interpolate`, the lattice spacing will be `sigma/interpolate.fac`.

`leaveoneout`
: Use the leave-one-out estimator for γ. See Shaw et al 2020 for details.

`exp_prs`
: A function that returns values for $\gamma_{\text{iso}}(r)$. If γ is known explicitly, or the same calculation is being used for several point patterns, it can be much faster to compute it once and provide the function as `exp_prs`, since the computation of γ is usually the slowest part.

`interpolate.maxdx`
: Upper bound on allowable lattice spacing for interpolation.

`dump`
: For debugging purposes, include computed values of γ with the output, as `attrs`.

Value

The return value is an object of class `fv`, just as for `pcf` and `pcfinhom`. The object contains columns `r`, `theo`, and `global`, corresponding respectively to the argument r, the theoretical values of $g(r)$ for a Poisson process, and $g_{\text{global}}(r)$.

Author(s)

Thomas Shaw <shawtr@umich.edu>

References

T Shaw, J Møller, R Waagepetersen. 2020. "Globally Intensity-Reweighted Estimators for K- and pair correlation functions". arXiv:2004.00527 [stat.ME].

See Also

`expectedPairs`

Examples

```
rho <- funxy(function(x,y) 80*(1+x), owin())
X <- rpoispp(rho)
g <- pcfglobal(X)
#plot(g)
```

Index