Package ‘spatstat.Knet’

February 16, 2022

Type Package

Title Extension to 'spatstat' for Large Datasets on a Linear Network

Version 2.0-1

Date 2022-01-12

Depends R (>= 3.5.0), spatstat.data (>= 2.0), spatstat.sparse (>= 2.0), spatstat.geom (>= 2.0), spatstat.random (>= 2.0), spatstat.core (>= 2.3-2), spatstat.linnet (>= 2.3-1), spatstat (>= 2.0)

Imports spatstat.utils (>= 2.1), Matrix

Maintainer Adrian Baddeley <Adrian.Baddeley@curtin.edu.au>

Description Extension to the 'spatstat' family of packages, for analysing large datasets of spatial points on a network. The geometrically-corrected K function is computed using a memory-efficient tree-based algorithm described by Rakshit, Baddeley and Nair (2019).

License GPL (>= 2)

NeedsCompilation yes

ByteCompile true

Author Suman Rakshit [aut, cph] (<https://orcid.org/0000-0003-0052-128X>), Adrian Baddeley [cre, cph] (<https://orcid.org/0000-0001-9499-8382>)

Repository CRAN

Date/Publication 2022-02-16 13:40:05 UTC

R topics documented:

spatstat.Knet-package *Extension to 'spatstat' for Large Datasets on a Linear Network*

Description

Extension to the 'spatstat' family of packages, for analysing large datasets of spatial points on a network. The geometrically- corrected K function is computed using a memory-efficient tree-based algorithm described by Rakshit, Baddeley and Nair (2019).

Details

This is an extension to the **spatstat** package for the analysis of large data sets on linear networks.

Its main functionality is a memory-efficient algorithm for computing the estimate of the K function on a linear network, described in Rakshit et al (2019).

The main functions are `Knet` and `Knetinhom`. These are counterparts of the functions `linearK` and `linearKinhom` in the **spatstat.linnet** package.

The **spatstat.linnet** functions `linearK` and `linearKinhom` are usable (and slightly faster) for small datasets, but require substantial amounts of memory. For larger datasets, the functions `Knet` and `Knetinhom` are much more efficient.

The DESCRIPTION file:

Package:	spatstat.Knet
Type:	Package
Title:	Extension to 'spatstat' for Large Datasets on a Linear Network
Version:	2.0-1
Date:	2022-01-12
Depends:	R (>= 3.5.0), spatstat.data (>= 2.0), spatstat.sparse (>= 2.0), spatstat.geom (>= 2.0), spatstat.random (>=
Imports:	spatstat.utils (>= 2.1), Matrix
Authors@R:	c(person(given="Suman", family="Rakshit", role = c("aut", "cph"), email = "suman.rakshit@curtin.edu.
Maintainer:	Adrian Baddeley <Adrian.Baddeley@curtin.edu.au>
Description:	Extension to the 'spatstat' family of packages, for analysing large datasets of spatial points on a network
License:	GPL (>= 2)
NeedsCompilation:	yes
ByteCompile:	true
Author:	Suman Rakshit [aut, cph] (<https://orcid.org/0000-0003-0052-128X>), Adrian Baddeley [cre, cph] (<htt

Index of help topics:

```
Knet                    Geometrically-Corrected K Function on Network
Knetinhom               Geometrically-Corrected Inhomogeneous K
                        Function on Network
spatstat.Knet-package   Extension to 'spatstat' for Large Datasets on a
                        Linear Network
wacrashes               Road Accidents in Western Australia
```

Author(s)

NA

Maintainer: Adrian Baddeley <Adrian.Baddeley@curtin.edu.au>

References

Rakshit, S., Baddeley, A. and Nair, G. (2019) Efficient code for second order analysis of events on a linear network. *Journal of Statistical Software* **90** (1) 1–37. DOI: `10.18637/jss.v090.i01`

`Knet` *Geometrically-Corrected K Function on Network*

Description

Compute the geometrically-corrected K function for a point pattern on a linear network.

Usage

```
Knet(X, r = NULL, freq, ..., verbose=FALSE)
```

Arguments

`X`	Point pattern on a linear network (object of class `"lpp"`).
`r`	Optional. Numeric vector of values of the function argument r. There is a sensible default.
`freq`	Vector of frequencies corresponding to the point events on the network. The length of this vector should be equal to the number of points on the network. The default frequency is one for every point on the network.
`...`	Ignored.
`verbose`	A logical for printing iteration number corresponding to each point event on the network.

Details

This command computes the geometrically-corrected K function, proposed by Ang et al (2012), from point pattern data on a linear network. The algorithm used in this computation is discussed in Rakshit et al (2019).

The **spatstat** function `linearK` is usable (and slightly faster) for the same purpose for small datasets, but requires substantial amounts of memory. For larger datasets, the function `Knet` is much more efficient.

Value

Function value table (object of class `"fv"`).

Author(s)

Suman Rakshit (modified by Adrian Baddeley)

References

Ang, Q.W., Baddeley, A. and Nair, G. (2012) Geometrically corrected second-order analysis of events on a linear network, with applications to ecology and criminology. *Scandinavian Journal of Statistics* **39**, 591–617.

Rakshit, S., Baddeley, A. and Nair, G. (2019) Efficient code for second order analysis of events on a linear network. *Journal of Statistical Software* **90** (1) 1–37. DOI: `10.18637/jss.v090.i01`

Examples

```
UC <- unmark(chicago)
r <- seq(0, 1000, length = 41)
K <- Knet(UC, r = r)
```

`Knetinhom` *Geometrically-Corrected Inhomogeneous K Function on Network*

Description

Compute the geometrically-corrected inhomogeneous K function for a point pattern on a linear network.

Usage

```
Knetinhom(X, lambda, r = NULL, freq, ..., verbose=FALSE)
```

Arguments

`X`	Point pattern on a linear network (object of class `"lpp"`).
`lambda`	Fitted intensity of the point process. Either a numeric vector giving values of the fitted intensity at each data point of `X`, or an object of class `"linim"`, `"linfun"` or `"lppm"` from which the fitted intensity at each data point can be extracted.
`r`	Optional. Numeric vector of values of the function argument r. There is a sensible default.
`freq`	Vector of frequencies corresponding to the point events on the network. The length of this vector should be equal to the number of points on the network. The default frequency is one for every point on the network.
`...`	Ignored.
`verbose`	Logical value indicating whether to print progress reports during the computation.

Details

This command computes the inhomogeneous version of the geometrically-corrected K function, proposed by Ang et al (2012), from point pattern data on a linear network.

The algorithm used in this computation is described in Rakshit et al (2019).

The **spatstat** function `linearKinhom` is usable (and slightly faster) for this purpose for small datasets, but requires substantial amounts of memory. For larger datasets, the function `Knetinhom` is much more efficient.

Value

Function value table (object of class `"fv"`).

Author(s)

Suman Rakshit (modified by Adrian Baddeley)

References

Ang, Q.W., Baddeley, A. and Nair, G. (2012) Geometrically corrected second-order analysis of events on a linear network, with applications to ecology and criminology. *Scandinavian Journal of Statistics* **39**, 591–617.

Rakshit, S., Baddeley, A. and Nair, G. (2019) Efficient code for second order analysis of events on a linear network. *Journal of Statistical Software* **90** (1) 1–37. DOI: `10.18637/jss.v090.i01`

Examples

```
UC <- unmark(chicago)
fit <- lppm(UC ~ x+y)
r <- seq(0, 1000, length = 41)
K <- Knetinhom(UC, lambda=fit, r = r)
```

`wacrashes` *Road Accidents in Western Australia*

Description

This dataset gives the spatial locations of all road accidents recorded in the state of Western Australia for the year 2011, on the state road network.

These data were published and analysed in Rakshit et al (2019).

Usage

```
data(wacrashes)
```

Format

A object of class `"lpp"` representing the spatial point pattern of accident locations on the network of roads in Western Australia.

Details

The road network has 88,512 intersections and 115,169 road segments. The spatial coordinates are expressed in metres, and the total network length is 97,165,540 metres (97,165 km). The number of accident locations on the network is 14,562.

Source

Main Roads, Western Australia. Made available as part of the Western Australian Whole of Government Open Data Policy.

References

Rakshit, S., Baddeley, A. and Nair, G. (2019) Efficient code for second order analysis of events on a linear network. *Journal of Statistical Software* **90** (1) 1–37. DOI: `10.18637/jss.v090.i01`

Examples

```
data(wacrashes)
wacrashes
summary(wacrashes)
plot(wacrashes, cols="red", cex=0.5)
```

Index