

Package ‘NScluster’

March 19, 2018

Version 1.2.0

Title Simulation and Estimation of the Neyman-Scott Type Spatial Cluster Models

Depends R (>= 3.0.0)

Imports graphics

Description Simulation and estimation for Neyman-Scott spatial cluster point process models and their extensions, based on the methodology in Tanaka, Ogata, and Stoyan (2008) <doi:10.1002/bimj.200610339>. For estimating parameters by the simplex method, parallel computation using 'OpenMP' application programming interface is available.

License GPL (>= 2)

MailingList Please send bug reports to ismrp@jasp.ism.ac.jp

NeedsCompilation yes

Author Ushio Tanaka [aut] (Fortran original),
Junji Nakano [aut],
Masami Saga [aut, cre]

Maintainer Masami Saga <msaga@mtb.biglobe.ne.jp>

Repository CRAN

Date/Publication 2018-03-19 08:37:41 UTC

R topics documented:

NScluster-package	2
EstimateIP	3
EstimateThomas	4
EstimateTypeA	6
EstimateTypeB	8
EstimateTypeC	10
PalmIP	12
PalmThomas	13
PalmTypeA	14
PalmTypeB	16

PalmTypeC	17
SimulateIP	19
SimulateThomas	20
SimulateTypeA	21
SimulateTypeB	23
SimulateTypeC	24

Index	26
--------------	-----------

NScluster-package	<i>Simulation and Estimation of the Neyman-Scott Type Spatial Cluster Models</i>
-------------------	--

Description

This package provides functions for simulation and estimation of spatial cluster point pattern of Neyman-Scott models and their extensions. We adopt the Simplex estimation to maximize the Palm likelihood function.

Details

The documentation 'NScluster: An R Package for Simulation and Estimation of the Neyman-Scott type spatial cluster models' is available in [../doc/NScluster-guide.pdf](#).

Simulation:

[SimulateThomas](#), [SimulateIP](#), [SimulateTypeA](#), [SimulateTypeB](#) and [SimulateTypeC](#) simulate spatial cluster point pattern of Neyman-Scott models and their extensions. We describe overview of those models briefly in the NScluster documentation [../doc/NScluster-guide.pdf](#).

Simulation method of each model is described under the corresponding topic.

Parameter estimation:

We adopt the Simplex estimation to maximize the Palm likelihood function (or minimize the negative Palm likelihood function). The maximum Palm likelihood estimate is called 'MPLE', for short. The Palm intensity function and the analytical form of the Palm log-likelihood of the Tomas model, Type B model and Type C model are described under the topic [EstimateThomas](#), [EstimateTypeB](#) and [EstimateTypeC](#), respectively. On the other hand, for [EstimateIP](#) and [EstimateTypeA](#), we need to take the alternative form without explicit representation of the Palm intensity function, which need very long c.p.u. time in the minimization procedure. We parallelize the minimization procedure with OpenMP.

[PalmThomas](#), [PalmIP](#), [PalmTypeA](#), [PalmTypeB](#) and [PalmTypeC](#) calculate the non-parametric Palm intensity function estimated directory from a set of point pattern data.

References

U. Tanaka, Y. Ogata and K. Katsura, Simulation and estimation of the Neyman-Scott type spatial cluster models, *Computer Science Monographs* **No.34**, 2008, 1-44, The Institute of Statistical Mathematics, Tokyo. <http://www.ism.ac.jp/editsec/csm/>

U.Tanaka, Y. Ogata and D. Stoyan, Parameter estimation and model selection for Neyman-Scott point processes, *Biometrical Journal*, **50**, 2008, 43-57.

EstimateIP

*Parameter Estimation of the Inverse-Power Type Model***Description**

Parameter estimation of the Inverse-power type model via numerical calculation of the Ripley's K -function.

Usage

```
EstimateIP(xy.points, pars, eps = 0.001, uplimit = 0.3, skip = 1,
           process.report = 0, plot = TRUE)
```

Arguments

<code>xy.points</code>	a matrix containing the coordinates (x, y) of points in a unit square: $W = [0, 1] \times [0, 1]$.
<code>pars</code>	a named vector of containing the initial guess of the model parameters (μ , ν , ρ , c), where μ is an intensity of parents, ν is an expected number of descendants for each parent, ρ is the decay order and c is the scaling parameter.
<code>eps</code>	the optimization procedure is iterated at most 1000 times until <code>process2\$stderr</code> becomes smaller than <code>eps</code> .
<code>uplimit</code>	upper limit value in place of ∞ in the integral in distribution function.
<code>skip</code>	the variable for the fast likelihood but rough approximation of the initial estimates. The <code>skip</code> calculate the Palm intensity function in the log-likelihood function for every <code>skip</code> -th r_{ij} in the ordered distances of the pairs i and j .
<code>process.report</code>	the level of reporting the process of minimizing. Allowed values are as follows: 0 no report (default). 1 output the process of minimizing the negative Palm log-likelihood function until the values converge to MPLEs. (<code>process1</code>) 2 output the process of optimizing by the simplex with the normalized parameters. (<code>process2</code>) 3 output both processes.
<code>plot</code>	logical. If TRUE (default), the process of optimizing by the simplex with the normalized parameters is plotted.

Value

<code>mple</code>	MPLE (maximum Palm likelihood estimate).
<code>process1</code>	a list with following components. (Only returned if <code>process.report = 1</code> or <code>3</code> .) cflg 1 ("update") or -1 ("testfn"), where "update" indicates that $-\log L$ value has attained the minimum so far, otherwise not. logl.palm the minimized $-\log L$ in the process of minimizing the negative Palm log-likelihood function.

mples corresponding MPLEs.

process2 a list with following components. (Only returned if `process.report = 2` or `3`.)

logl.simplex the minimized $-\log L$ by the simplex method.

stderr the standard deviations.

pa.normal the normalized variables corresponding the initial estimates.

Note

EstimateIP and EstimateTypeA have to use numerical integration and difference to compute the functions, which need very long c.p.u. time in the minimization procedure.

References

U. Tanaka, Y. Ogata and K. Katsura, Simulation and estimation of the Neyman-Scott type spatial cluster models, *Computer Science Monographs* **No.34**, 2008, 1-44. The Institute of Statistical Mathematics.

Examples

```
## simulation
pars <- c(mu = 50.0, nu = 30.0, p = 1.5, c = 0.005)
z <- SimulateIP(pars, seed = 353)

## estimation
## need very long c.p.u time in the minimization procedure
## Not run:
init.pars <- c(mu = 55.0, nu = 35.0, p = 1.0, c = 0.01)
EstimateIP(z$offspring$xy, init.pars, skip = 100)

## End(Not run)
```

EstimateThomas

Parameter Estimation of the Thomas Model

Description

Parameter estimation of the Thomas model by using the Palm log-likelihood function.

Usage

```
EstimateThomas(xy.points, pars, eps = 0.001, process.report = 0, plot = TRUE)
```

Arguments

<code>xy.points</code>	a matrix containing the coordinates (x, y) of points in a unit square: $W = [0, 1] \times [0, 1]$.
<code>pars</code>	a named vector of containing the initial guess of the model parameters (<code>mu</code> , <code>nu</code> , <code>sigma</code>), where <code>mu</code> is an intensity of parents, <code>nu</code> is an expected number of descendants for each parent and <code>sigma</code> is a parameter of the dispersal kernel.
<code>eps</code>	the optimization procedure is iterated at most 1000 times until <code>process2\$stderr</code> becomes smaller than <code>eps</code> .
<code>process.report</code>	the level of reporting the process of minimizing. Allowed values are as follows: 0 no report (default). 1 output the process of minimizing the negative Palm log-likelihood function until the values converge to MPLEs. (<code>process1</code>) 2 output the process of optimizing by the simplex with the normalized parameters. (<code>process2</code>) 3 output both processes.
<code>plot</code>	logical. If TRUE (default), the process of optimizing by the simplex with the normalized parameters is plotted.

Details

The Palm intensity function of the Thomas model is calculated as follows:

For all $r \geq 0$,

$$\lambda_{\mathbf{o}}(r) = \mu\nu + \frac{\nu}{4\pi\sigma^2} \exp\left(-\frac{r^2}{4\sigma^2}\right).$$

The Palm log-likelihood function of the Thomas model on W is given by

$$\log L(\mu, \nu, \sigma) = \sum_{\{i,j;i < j, r_{ij} \leq 1/2\}} \log \nu \left\{ \mu + \frac{1}{4\pi\sigma^2} \exp\left(-\frac{r_{ij}^2}{4\sigma^2}\right) \right\} \\ - N(W)\nu \left\{ \frac{\pi\mu}{4} + 1 - \exp\left(-\frac{1}{16\sigma^2}\right) \right\}.$$

Value

<code>mple</code>	MPLE (maximum Palm likelihood estimate).
<code>process1</code>	a list with following components. (Only returned if <code>process.report = 1</code> or <code>3</code>). cflg 1 ("update") or -1 ("testfn"), where "update" indicates that $-\log L$ value has attained the minimum so far, otherwise not. logl.palm the minimized $-\log L$ in the process of minimizing the negative Palm log-likelihood function. mples corresponding MPLEs.

process2 a list with following components. (Only returned if process.report = 2 or 3.)
logl.simplex the minimized -log L by the simplex method.
stderr the standard deviations.
pa.normal the normalized variables corresponding the initial estimates.

References

U. Tanaka, Y. Ogata and K. Katsura, Simulation and estimation of the Neyman-Scott type spatial cluster models, *Computer Science Monographs* **No.34**, 2008, 1-44. The Institute of Statistical Mathematics.

Examples

```
## simulation
pars <- c(mu = 50.0, nu = 30.0, sigma = 0.03)
z <- SimulateThomas(pars, seed = 117)

## estimation
## need long c.p.u time in the minimization procedure
## Not run:
init.pars <- c(mu = 40.0, nu = 40.0, sigma = 0.05)
EstimateThomas(z$offspring$xy, init.pars)

## End(Not run)
```

EstimateTypeA

Parameter Estimation of the Type A Model

Description

Parameter estimation of the Type A model via numerical calculation of the Ripley's K -function.

Usage

```
EstimateTypeA(xy.points, pars, eps = 0.001, uplimit = 0.3, skip = 1,
              process.report = 0, plot=TRUE)
```

Arguments

xy.points a matrix containing the coordinates (x,y) of points in a unit square: $W = [0, 1] \times [0, 1]$.

pars a named vector of containing the initial guess of the model parameters (mu, nu, a, sigma1, sigma2), where mu is an intensity of parents, nu is an expected number of descendants for each parent, a is a mixture parameter, sigma1 and sigma2 are parameters of the dispersal kernel for each component.

eps the optimization procedure is iterated at most 1000 times until process2\$stderr becomes smaller than eps.

<code>uplimit</code>	upper limit value in place of ∞ in the integral in the distribution function.
<code>skip</code>	the variable for the fast likelihood but rough approximation of the initial estimates. The skip calculate the Palm intensity function in the log-likelihood unction for every skip-th r_{ij} in the ordered distances of the pairs i and j .
<code>process.report</code>	the level of reporting the process of minimizing. Allowed values are as follows: 0 no report (default). 1 output the process of minimizing the negative Palm log-likelihood function until the values converge to MPLEs. (<code>process1</code>) 2 output the process of optimizing by the simplex with the normalized parameters. (<code>process2</code>) 3 output both processes.
<code>plot</code>	logical. If TRUE (default), the process of optimizing by the simplex with the normalized parameters is plotted.

Value

<code>mple</code>	MPLE (maximum Palm likelihood estimate).
<code>process1</code>	a list with following components. (Only returned if <code>process.report = 1</code> or <code>3</code> .) cfg 1 ("update") or -1 ("testfn"), where "update" indicates that -log L value has attained the minimum so far, otherwise not. logl.palm the minimized -log L in the process of minimizing the negative Palm log-likelihood function. mples corresponding MPLEs.
<code>process2</code>	a list with following components. (Only returned if <code>process.report = 2</code> or <code>3</code> .) logl.simplex the minimized -log L by the simplex method. stderr the standard deviations. pa.normal the normalized variables corresponding the initial estimates.

Note

EstimateIP and EstimateTypeA have to use numerical integration and difference to compute the functions, which need very long c.p.u. time in the minimization procedure.

References

U. Tanaka, Y. Ogata and K. Katsura, Simulation and estimation of the Neyman-Scott type spatial cluster models, *Computer Science Monographs* **No.34**, 2008, 1-44. The Institute of Statistical Mathematics.

Examples

```
## simulation
pars <- c(mu = 50.0, nu = 30.0, a = 0.3, sigma1 = 0.005, sigma2 = 0.1)
z <- SimulateTypeA(pars, seed = 575)

## estimation
```

```
## need very long c.p.u time in the minimization procedure
## Not run:
init.pars <- c(mu = 60.0, nu = 40.0, a = 0.5, sigma1 = 0.01, sigma2 = 0.1)
EstimateTypeA(z$offspring$xy, init.pars, skip = 100)

## End(Not run)
```

EstimateTypeB

Parameter Estimation of the Type B Model

Description

Parameter estimation of the Type B model by using the Palm Log-Likelihood Function.

Usage

```
EstimateTypeB(xy.points, pars, eps = 0.001, process.report = 0, plot = TRUE)
```

Arguments

xy.points	a matrix containing the coordinates (x,y) of points in a unit square: $W = [0, 1] \times [0, 1]$.
pars	a named vector of containing the initial guess of the model parameters (mu1, mu2, nu, sigma1, sigma2), where (mu i , nu, sigma i) is an intensity of parents, an expected number of descendants, a parameter of the dispersal kernel for superposed component i ($i = 1, 2$), respectively.
eps	the optimization procedure is iterated at most 1000 times until process2\$stderr becomes smaller than eps.
process.report	the level of reporting the process of minimizing. Allowed values are as follows: <ul style="list-style-type: none"> 0 no report (default). 1 output the process of minimizing the negative Palm log-likelihood function until the values converge to MPLEs. (process1) 2 output the process of optimizing by the simplex with the normalized parameters. (process2) 3 output both processes.
plot	logical. If TRUE (default), the process of optimizing by the simplex with the normalized parameters is plotted.

Details

The Palm intensity function of the Type B model is calculated as follows:

For all $r \geq 0$,

$$\lambda_o(r) = \lambda + \frac{\nu}{4\pi} \left\{ \frac{a}{\sigma_1^2} \exp\left(-\frac{r^2}{4\sigma_1^2}\right) + \frac{(1-a)}{\sigma_2^2} \exp\left(-\frac{r^2}{4\sigma_2^2}\right) \right\},$$

where $\lambda = \nu(\mu_1 + \mu_2)$ is the total population size and $a = \mu_1/(\mu_1 + \mu_2)$ is the ratio of the parent points of the smaller sized cluster to the total ones.

The Palm log-likelihood function of the Type B model on W is given by

$\log L(\lambda, \alpha, \beta, \sigma_1, \sigma_2)$

$$= \sum_{\{i,j:i < j, r_{ij} \leq 1/2\}} \log \left[\lambda + \frac{1}{4\pi} \left\{ \frac{\alpha}{\sigma_1^2} \exp\left(-\frac{r_{ij}^2}{4\sigma_1^2}\right) + \frac{\beta}{\sigma_2^2} \exp\left(-\frac{r_{ij}^2}{4\sigma_2^2}\right) \right\} \right] \\ - N(W) \left[\frac{\pi\lambda}{4} + \alpha \left\{ 1 - \exp\left(-\frac{1}{16\sigma_1^2}\right) \right\} + \beta \left\{ 1 - \exp\left(-\frac{1}{16\sigma_2^2}\right) \right\} \right],$$

where $\alpha = a\nu$ and $\beta = (1 - a)\nu$.

Value

<code>mple</code>	MPL (maximum Palm likelihood estimate).
<code>process1</code>	a list with following components. (Only returned if <code>process.report = 1</code> or <code>3</code> .) cfg 1 ("update") or -1 ("testfn"), where "update" indicates that -log L value has attained the minimum so far, otherwise not. logl.palm the minimized -log L in the process of minimizing the negative Palm log-likelihood function. mple corresponding MPLs (<code>mu</code> , <code>nu</code> , <code>a</code> , <code>sigma1</code> , <code>sigma2</code>), where <code>mu = mu1+mu2</code> and <code>a = mu1/(mu1+mu2)</code> .
<code>process2</code>	a list with following components. (Only returned if <code>process.report = 2</code> or <code>3</code> .) logl.simplex the minimized -log L by the simplex method. stderr the standard deviations. pa.normal the normalized variables (<code>mu</code> , <code>nu</code> , <code>a</code> , <code>sigma1</code> , <code>sigma2</code>) as described above.

References

U. Tanaka, Y. Ogata and K. Katsura, Simulation and estimation of the Neyman-Scott type spatial cluster models, *Computer Science Monographs* **No.34**, 2008, 1-44. The Institute of Statistical Mathematics.

Examples

```
## simulation
pars <- c(mu1 = 10.0, mu2 = 40.0, nu = 30.0, sigma1 = 0.01, sigma2 = 0.03)
z <- SimulateTypeB(pars, seed = 257)

## estimation
## need very long c.p.u time in the minimization procedure
## Not run:
init.pars <- c(mu1 = 20.0, mu2 = 30.0, nu = 30.0, sigma1 = 0.02, sigma2 = 0.02)
EstimateTypeB(z$offspring$xy, init.pars)

## End(Not run)
```

EstimateTypeC

*Parameter Estimation of the Type C Model***Description**

Parameter estimation of the Type C model by using the Palm Log-Likelihood Function.

Usage

```
EstimateTypeC(xy.points, pars, eps = 0.001, process.report = 0, plot = TRUE)
```

Arguments

xy.points	a matrix containing the coordinates (x,y) of points in a unit square: $W = [0, 1] \times [0, 1]$.
pars	a named vector of containing the initial guess of the model parameters (mu1, mu2, nu1, nu2, sigma1, sigma2), where (mu i , nu i , sigma i) is an intensity of parents, an expected number of descendants, a parameter of the dispersal kernel for superposed component i ($i = 1, 2$), respectively.
eps	the optimization procedure is iterated at most 1000 times until process2\$stderr becomes smaller than eps.
process.report	the level of reporting the process of minimizing. Allowed values are as follows: 0 no report (default). 1 output the process of minimizing the negative Palm log-likelihood function until the values converge to MPLEs. (process1) 2 output the process of optimizing by the simplex with the normalized parameters. (process2) 3 output both processes.
plot	logical. If TRUE (default), the process of optimizing by the simplex with the normalized parameters is plotted.

Details

The Palm intensity function of the Type C model is calculated as follows:

For all $r \geq 0$,

$$\lambda_o(r) = \lambda + \frac{1}{4\pi} \left\{ \frac{a\nu_1}{\sigma_1^2} \exp\left(-\frac{r^2}{4\sigma_1^2}\right) + \frac{(1-a)\nu_2}{\sigma_2^2} \exp\left(-\frac{r^2}{4\sigma_2^2}\right) \right\},$$

where $\lambda = \mu_1\nu_1 + \mu_2\nu_2$ is the total population size and $a = \mu_1\nu_1/\lambda$ is the ratio of the all offspring points of smaller sized cluster to the total population size.

The Palm log-likelihood function of the Type C model on W is given by

$$\begin{aligned} & \log L(\lambda, \alpha, \beta, \sigma_1, \sigma_2) \\ &= \sum_{\{i,j;i < j, r_{ij} \leq 1/2\}} \log \left[\lambda + \frac{1}{4\pi} \left\{ \frac{\alpha}{\sigma_1^2} \exp\left(-\frac{r_{ij}^2}{4\sigma_1^2}\right) + \frac{\beta}{\sigma_2^2} \exp\left(-\frac{r_{ij}^2}{4\sigma_2^2}\right) \right\} \right] \\ & \quad - N(W) \left[\frac{\pi\lambda}{4} + \alpha \left\{ 1 - \exp\left(-\frac{1}{16\sigma_1^2}\right) \right\} + \beta \left\{ 1 - \exp\left(-\frac{1}{16\sigma_2^2}\right) \right\} \right], \end{aligned}$$

where $\alpha = a\nu_1$ and $\beta = (1 - a)\nu_2$.

Value

mple MPLE (maximum Palm likelihood estimate).
process1 a list with following components. (Only returned if `process.report = 1` or `3`.)
cfg 1 ("update") or -1 ("testfn"), where "update" indicates that -log L value has attained the minimum so far, otherwise not.
logl.palm the minimized -log L in the process of minimizing the negative Palm log-likelihood function.
mple corresponding MPLEs (`lambda`, `nu1`, `a`, `sigma1`, `sigma2`), where `lambda = mu1*nu1+mu2*nu2` and `a = mu1*nu1/lambda`.
process2 a list with following components. (Only returned if `process.report = 2` or `3`.)
logl.simplex the minimized -log L by the simplex method.
stderr the standard deviations.
pa.normal the normalized variables (`lambda`, `nu1`, `a`, `sigma1`, `sigma2`) as described above.

References

U. Tanaka, Y. Ogata and K. Katsura, Simulation and estimation of the Neyman-Scott type spatial cluster models, *Computer Science Monographs* **No.34**, 2008, 1-44. The Institute of Statistical Mathematics.

Examples

```
## simulation
pars <- c(mu1 = 5.0, mu2 = 9.0, nu1 = 30.0, nu2 = 150.0,
         sigma1 = 0.01, sigma2 = 0.05)
z <- SimulateTypeC(pars, seed = 555)

## estimation
## need long c.p.u time in the minimization procedure
## Not run:
init.pars <- c(mu1 = 10.0, mu2 = 10.0, nu1 = 30.0, nu2 = 120.0,
             sigma1 = 0.03, sigma2 = 0.03)
EstimateTypeC(z$offspring$xy, init.pars)

## End(Not run)
```

PalmIP	<i>Non-Parametric and Parametric Estimate of the Palm Intensity of Inverse-Power Type Model</i>
--------	---

Description

Calculate the non-parametric and parametric Palm intensity function of the Inverse-power type model estimated directly from a set of point pattern data.

Usage

```
PalmIP(xy.points, pars1 = NULL, pars2 = NULL, delta = 0.001, uplimit = 0.3,
        plot=TRUE)
```

Arguments

xy.points	a matrix containing the coordinates (x, y) of points in a unit square: $W = [0, 1] \times [0, 1]$.
pars1	a named vector of true parameters (μ, ν, p, c) , where μ is an intensity of parents, ν is an expected number of descendants for each parent, p is the decay order and c is the scaling parameter.
pars2	a named vector of MPLEs (the maximum Palm likelihood estimates) (μ, ν, p, c) .
delta	a width for the non-parametric Palm intensity function.
uplimit	upper limit value in place of ∞ .
plot	logical. If TRUE (default), the non-parametric estimate and the curves of true parameters and MPLEs are shown.

Value

r	the distance $r = j\Delta$, where $j = 1, 2, \dots, [R/\Delta]$, where $[]$ is the Gauss' symbol and $R = 1/2$ is given in the program for the normalized rectangular region for the point pattern.
np.palm	the corresponding values of the non-parametric Palm intensity function of r , which is normalized by the total intensity estimate of the point pattern data.
palm.normal	the normalized Palm intensity functions $\lambda_o(r)/\hat{\lambda}$ calculated from the given sets of parameter values. See ../doc/NScluster-guide.pdf .

References

U. Tanaka, Y. Ogata and K. Katsura, Simulation and estimation of the Neyman-Scott type spatial cluster models, *Computer Science Monographs* **No.34**, 2008, 1-44. The Institute of Statistical Mathematics.

Examples

```
## simulation
pars <- c(mu = 50.0, nu = 30.0, p = 1.5, c = 0.005)
z <- SimulateIP(pars, seed = 353)

## estimation
## need very long c.p.u time in the minimization procedure
## Not run:
init.pars <- c(mu = 55.0, nu = 35.0, p = 1.0, c = 0.01)
z1 <- EstimateIP(z$offspring$xy, init.pars, skip = 100)
# Parameter      mu      nu      p      c
# Initial value  55.0000  35.0000  1.0000  0.0100
# MPLE           48.3143  30.5047  1.6467  0.0079

## End(Not run)

## Palm intensity
par1 <- c(50.0, 30.0, 1.5, 0.005) # pars
par2 <- c(48.3143, 30.5047, 1.6467, 0.0079) # z1$mple
PalmIP(z$offspring$xy, par1, par2)
```

PalmThomas

*Non-Parametric and Parametric Estimate of the Palm Intensity of
Thomas Model*

Description

Calculate the non-parametric and parametric Palm intensity function of the Thomas model estimated directly from a set of point pattern data.

Usage

```
PalmThomas(xy.points, pars1 = NULL, pars2 = NULL, delta = 0.001, plot = TRUE)
```

Arguments

xy.points	a matrix containing the coordinates (x,y) of points in a unit square: $W = [0, 1] \times [0, 1]$.
pars1	a named vector of the true parameters (mu, nu, sigma), where mu is an intensity of parents, nu is an expected number of descendants for each parent and sigma is a parameter of the dispersal kernel.
pars2	a named vector of MPLEs (the maximum Palm likelihood estimates) (mu, nu, sigma).
delta	a width for the non-parametric Palm intensity function.
plot	logical. If TRUE (default), the non-parametric estimate and the curves of true parameters and MPLEs are shown.

Value

r	the distance $r = j\Delta$, where $j = 1, 2, \dots, [R/\Delta]$, where $[]$ is the Gauss' symbol and $R = 1/2$ is given in the program for the normalized rectangular region for the point pattern.
np.palm	the corresponding values of the non-parametric Palm intensity function of r , which is normalized by the total intensity estimate of the point pattern data.
palm.normal	the normalized Palm intensity functions $\lambda_o(r)/\hat{\lambda}$ calculated from the given sets of parameter values. See 'Details' in EstimateThomas .

References

U. Tanaka, Y. Ogata and K. Katsura, Simulation and estimation of the Neyman-Scott type spatial cluster models, *Computer Science Monographs* **No.34**, 2008, 1-44. The Institute of Statistical Mathematics.

Examples

```
## simulation
pars <- c(mu = 50.0, nu = 30.0, sigma = 0.03)
z <- SimulateThomas(pars, seed = 117)

## estimation
## need long c.p.u time in the minimization procedure
## Not run:
init.pars <- c(mu = 40.0, nu = 40.0, sigma = 0.05)
z1 <- EstimateThomas(z$offspring$xy, init.pars)
# Parameter      mu      nu      sigma
# Initial value  40.0000  40.0000  0.0500
# MPLE           46.7657  27.2079  0.0279

## End(Not run)

## Palm intensity
par1 <- c(50.0, 30.0, 0.03) # pars
par2 <- c(46.7657, 27.2079, 0.0279) # z1$mple
PalmThomas(z$offspring$xy, par1, par2)
```

PalmTypeA

Non-Parametric and Parametric Estimate of the Palm Intensity of Type A Model

Description

Calculate the non-parametric and parametric Palm intensity function of Type A model estimated directly from a set of point pattern data.

Usage

```
PalmTypeA(xy.points, pars1 = NULL, pars2 = NULL, delta = 0.001, uplimit = 0.3,
          plot = TRUE)
```

Arguments

xy.points	a matrix containing the coordinates (x,y) of points in a unit square: $W = [0, 1] \times [0, 1]$.
pars1	a named vector of the true parameters (mu, nu, a, sigma1, sigma2), where mu is an intensity of parents, nu is an expected number of descendants for each parent, a is a mixture parameter, sigma1 and sigma2 are parameters of the dispersal kernel for each component.
pars2	a named vector of MPLEs (the maximum Palm likelihood estimates) (mu, nu, a, sigma1, sigma2).
delta	a width for the non-parametric Palm intensity function.
uplimit	upper limit value in place of ∞ .
plot	logical. If TRUE (default), the non-parametric estimate and the curves of true parameters and MPLEs are shown.

Value

r	the distance $r = j\Delta$, where $j = 1, 2, \dots, [R/\Delta]$, where $[]$ is the Gauss' symbol and $R = 1/2$ is given in the program for the normalized rectangular region for the point pattern.
np.palm	the corresponding values of the non-parametric Palm intensity function of r , which is normalized by the total intensity estimate of the point pattern data.
palm.normal	the normalized Palm intensity functions $\lambda_o(r)/\hat{\lambda}$ calculated from the given sets of parameter values. See ../doc/NScluster-guide.pdf .

References

U. Tanaka, Y. Ogata and K. Katsura, Simulation and estimation of the Neyman-Scott type spatial cluster models, *Computer Science Monographs* **No.34**, 2008, 1-44. The Institute of Statistical Mathematics.

Examples

```
## simulation
pars <- c(mu = 50.0, nu = 30.0, a = 0.3, sigma1 = 0.005, sigma2 = 0.1)
z <- SimulateTypeA(pars, seed=575)

## estimation
## need very long c.p.u time in the minimization procedure
## Not run:
init.pars <- c(mu=60.0, nu=40.0, a=0.5, sigma1=0.01, sigma2=0.1)
z1 <- EstimateTypeA(z$offspring$xy, init.pars, skip=100)
# Parameter      mu      nu      a      sigma1      sigma2
# Initial value  60.0000  40.0000  0.5000  0.0100  0.1000
```

```
# MPLE          51.2441      25.1439      0.3431      0.0054      0.0824

## End(Not run)

## Palm intensity
par1 <- c(50.0, 30.0, 0.3, 0.005, 0.1) # pars
par2 <- c(51.2441, 25.1439, 0.3431, 0.0054, 0.0824) # z1$mple
PalmTypeA(z$offspring$xy, par1, par2)
```

PalmTypeB	<i>Non-Parametric and Parametric Estimate of the Palm Intensity of Type B Model</i>
-----------	---

Description

Calculate the non-parametric and parametric Palm intensity function of Type B model estimated directly from a set of point pattern data.

Usage

```
PalmTypeB(xy.points, pars1 = NULL, pars2 = NULL, delta = 0.001, plot = TRUE)
```

Arguments

xy.points	a matrix containing the coordinates (x,y) of points in a unit square: $W = [0, 1] \times [0, 1]$.
pars1	a named vector of the true parameters (mu1, mu2, nu, sigma1, sigma2), where (mu i , nu, sigma i) is an intensity of parents, an expected number of descendants, a parameter of the dispersal kernel for superposed component i ($i = 1, 2$), respectively.
pars2	a named vector of MPLEs (the maximum Palm likelihood estimates) (mu1, mu2, nu, sigma1, sigma2).
delta	a width for the non-parametric Palm intensity function.
plot	logical. If TRUE (default), the non-parametric estimate and the curves of true parameters and MPLEs are shown.

Value

r	the distance $r = j\Delta$, where $j = 1, 2, \dots, [R/\Delta]$, where $[\]$ is the Gauss' symbol and $R = 1/2$ is given in the program for the normalized rectangular region for the point pattern.
np.palm	the corresponding values of the non-parametric Palm intensity function of r , which is normalized by the total intensity estimate of the point pattern data.
palm.normal	the normalized Palm intensity functions $\lambda_o(r)/\hat{\lambda}$ calculated from the given sets of parameter values. See 'Details' in EstimateTypeB .

References

U. Tanaka, Y. Ogata and K. Katsura, Simulation and estimation of the Neyman-Scott type spatial cluster models, *Computer Science Monographs* **No.34**, 2008, 1-44. The Institute of Statistical Mathematics.

Examples

```
## simulation
pars <- c(mu1 = 10.0, mu2 = 40.0, nu = 30.0, sigma1 = 0.01, sigma2 = 0.03)
z <- SimulateTypeB(pars, seed = 257)

## estimation
## need very long c.p.u time in the minimization procedure
## Not run:
init.pars <- c(mu1 = 20.0, mu2 = 30.0, nu = 30.0, sigma1 = 0.02, sigma2 = 0.02)
z1 <- EstimateTypeB(z$offspring$xy, init.pars)
# Parameter      mu1      mu2      nu      sigma1      sigma2
# Initial value  20.0000  30.0000  30.0000  0.0200  0.0200
# MPLE           16.1778  44.3974  28.3942  0.0101  0.0312

## End(Not run)

## Palm intensity
par1 <- c(10.0, 40.0, 30.0, 0.01, 0.03) # pars
par2 <- c(16.1778, 44.3974, 28.3942, 0.0101, 0.0312) # z1$mple
PalmTypeB(z$offspring$xy, par1, par2)
```

PalmTypeC

Non-Parametric and Parametric Estimate of the Palm Intensity of Type C Model

Description

Calculate the non-parametric and parametric Palm intensity function of Type C model estimated directly from a set of point pattern data.

Usage

```
PalmTypeC(xy.points, pars1 = NULL, pars2 = NULL, delta = 0.001, plot = TRUE)
```

Arguments

`xy.points` a matrix containing the coordinates (x,y) of points in a unit square: $W = [0, 1] \times [0, 1]$.

`pars1` a named vector of the true parameters (mu1, mu2, nu1, nu2, sigma1, sigma2), where (mu i , nu i , sigma i) is an intensity of parents, an expected number of descendants, a parameter of the dispersal kernel for superposed component i ($i = 1, 2$), respectively.

pars2	a named vector of MPLEs (the maximum Palm likelihood estimates) (mu1, mu2, nu1, nu2, sigma1, sigma2).
delta	a width for the non-parametric Palm intensity function.
plot	logical. If TRUE (default), the non-parametric estimate and the curves of true parameters and MPLEs are shown.

Value

r	the distance $r = j\Delta$, where $j = 1, 2, \dots, [R/\Delta]$, where $[]$ is the Gauss' symbol and $R = 1/2$ is given in the program for the normalized rectangular region for the point pattern.
np.palm	the corresponding values of the non-parametric Palm intensity function of r , which is normalized by the total intensity estimate of the point pattern data.
palm.normal	the normalized Palm intensity functions $\lambda_o(r)/\hat{\lambda}$ calculated from the given sets of parameter values. See 'Details' in EstimateTypeC .

References

U. Tanaka, Y. Ogata and K. Katsura, Simulation and estimation of the Neyman-Scott type spatial cluster models, *Computer Science Monographs* **No.34**, 2008, 1-44. The Institute of Statistical Mathematics.

Examples

```
## simulation
pars <- c(mu1 = 5.0, mu2 = 9.0, nu1 = 30.0, nu2 = 150.0,
         sigma1 = 0.01, sigma2 = 0.05)
z <- SimulateTypeC(pars, seed = 555)

## estimation
## need long c.p.u time in the minimization procedure
## Not run:
init.pars <- c(mu1 = 10.0, mu2 = 10.0, nu1 = 30.0, nu2 = 120.0,
              sigma1 = 0.03, sigma2 = 0.03)
z1 <- EstimateTypeC(z$offspring$xy, init.pars)
## Parameter      mu1      mu2      nu1      nu2      sigma1      sigma2
## Initial value  10.0000  10.0000  30.0000  120.0200  0.0300     0.0300
## MPLE           4.6481   5.1690   28.1250  296.7022  0.0088     0.0477

## End(Not run)

## Palm intensity
par1 <- c(5.0, 9.0, 30.0, 150.0, 0.01, 0.05) # pars
par2 <- c(4.6481, 5.1690, 28.1250, 296.7022, 0.0088, 0.0477) # z1$mple
PalmTypeC(z$offspring$xy, par1, par2)
```

 SimulateIP

Simulation of the Inverse-Power Type Model

Description

Simulation of the Inverse-power type model.

Usage

```
SimulateIP(pars, seed = NULL, plot = TRUE)
```

Arguments

pars	a named vector of containing the values of the model parameters (mu, nu, p, c), where mu is an intensity of parents, nu is an expected number of descendants for each parent, p is the decay order and c is the scaling parameter.
seed	a positive integer, which is the seed for a sequence of uniform random numbers. The default seed is based on the current time.
plot	logical. If TRUE (default), simulated parent points and offspring points are plotted.

Details

Let random variable U be independently and uniformly distributed in $[0,1]$.

For all $r \geq 0$,

$$\begin{aligned}
 Q_{p,c}(r) &:= \int_0^r q_{p,c}(t) dt \\
 &= c^{p-1}(p-1) \frac{(r+c)^{1-p} - c^{1-p}}{1-p} \\
 &= 1 - c^{p-1}(r+c)^{1-p}.
 \end{aligned}$$

Here, we put $Q_{p,c}(r) = U$. From this, we have

$$r = c\{(1-U)^{1/(1-p)} - 1\}.$$

Let $(x_i^p, y_i^p), i = 1, 2, \dots, I$ be a coordinate of each parent point where the integer I is generated from the Poisson random variable $Poisson(\mu)$ with mean μ from now on. Then, for each i , the number of offspring J_i is generated by the random variable $Poisson(\nu)$ with mean ν . Then, using series of different uniform random numbers $\{U\}$ for different i and j , each of the offspring coordinates $(x_j^i, y_j^i), j = 1, 2, \dots, J_i$ is given by

$$x_j^i = x_i^p + r \cos(2\pi U),$$

$$y_j^i = y_i^p + r \sin(2\pi U),$$

owing to the isotropy condition of the distribution.

Given a positive number ν and let a sequence of a random variable $\{U_k\}$ be independently and uniformly distributed in $[0,1]$, the Poisson random number M is the smallest integer such that

$$\sum_{k=1}^{M+1} -\log U_k > \nu,$$

where \log represents natural logarithm.

Value

parents	a list containing two components named "n" and "xy" giving the number and the matrix of (x, y) coordinates of simulated parents points respectively.
offspring	a list containing two components named "n" and "xy" giving the number and the matrix of (x, y) coordinates of simulated offspring points respectively.

References

U. Tanaka, Y. Ogata and K. Katsura, Simulation and estimation of the Neyman-Scott type spatial cluster models, *Computer Science Monographs* **No.34**, 2008, 1-44. The Institute of Statistical Mathematics.

Examples

```
pars <- c(mu = 50.0, nu = 30.0, p = 1.5, c = 0.005)
SimulateIP(pars, seed = 353)
```

SimulateThomas	<i>Simulation of the Thomas Model</i>
----------------	---------------------------------------

Description

Simulation of the Thomas model.

Usage

```
SimulateThomas(pars, seed = NULL, plot = TRUE)
```

Arguments

pars	a named vector of containing the values of the model parameters (mu, nu, sigma), where mu is an intensity of parents, nu is an expected number of descendants for each parent and sigma is a parameter of the dispersal kernel.
seed	a positive integer, which is the seed for a sequence of uniform random numbers. The default seed is based on the current time.
plot	logical. If TRUE (default), simulated parent points and offspring points are plotted.

Details

Let random variable U be independently and uniformly distributed in $[0,1]$. We put

$$U = \int_0^r q_\sigma(t) dt = 1 - \exp\left(-\frac{r^2}{2\sigma^2}\right).$$

Then we have

$$r = \sigma \sqrt{-2 \log(1 - U)}.$$

Each of the offspring coordinates (x_j^i, y_j^i) is given like that of [SimulateIP](#).

Value

parents	a list containing two components named "n" and "xy" giving the number and the matrix of (x, y) coordinates of simulated parents points respectively.
offspring	a list containing two components named "n" and "xy" giving the number and the matrix of (x, y) coordinates of simulated offspring points respectively.

References

U. Tanaka, Y. Ogata and K. Katsura, Simulation and estimation of the Neyman-Scott type spatial cluster models, *Computer Science Monographs* **No.34**, 2008, 1-44. The Institute of Statistical Mathematics.

Examples

```
pars <- c(mu = 50.0, nu = 30.0, sigma = 0.03)
SimulateThomas(pars, seed = 117)
```

 SimulateTypeA

Simulation of the Generalized Thomas Model of Type A

Description

Simulation of the generalized Thomas model of type A.

Usage

```
SimulateTypeA(pars, seed = NULL, plot = TRUE)
```

Arguments

pars	a named vector of containing the values of the model parameters (mu, nu, a, sigma1, sigma2), where mu is an intensity of parents, nu is an expected number of descendants for each parent, a is a mixture parameter, sigma1 and sigma2 are parameters of the dispersal kernel for each component.
seed	a positive integer, which is the seed for a sequence of uniform random numbers. The default seed is based on the current time.
plot	logical. If TRUE (default), simulated parent points and offspring points are plotted.

Details

Parents' configuration and numbers of the offspring cluster sizes are generated by the same way as the Thomas model.

Let random variable $U_k, k = 1, 2$ be independently and uniformly distributed in $[0,1]$. Then r satisfies as follows:

$$r = \sigma_1 \sqrt{-2 \log(1 - U_1)}, \quad U_2 \leq a,$$

$$r = \sigma_2 \sqrt{-2 \log(1 - U_1)}, \quad \text{otherwise.}$$

Each of the offspring coordinates (x_j^i, y_j^i) is given like that of [SimulateIP](#). Using series of different uniform random numbers $\{U_1, U_2, U\}$ for different i and j .

Value

parents	a list containing two components named "n" and "xy" giving the number and the matrix of (x, y) coordinates of simulated parents points respectively.
offspring	a list containing two components named "n" and "xy" giving the number and the matrix of (x, y) coordinates of simulated offspring points respectively.

References

U. Tanaka, Y. Ogata and K. Katsura, Simulation and estimation of the Neyman-Scott type spatial cluster models, *Computer Science Monographs* **No.34**, 2008, 1-44. The Institute of Statistical Mathematics.

Examples

```
pars <- c(mu = 50.0, nu = 30.0, a = 0.03, sigma1 = 0.005, sigma2 = 0.1)
SimulateTypeA(pars, seed = 575)
```

 SimulateTypeB

Simulation of the Generalized Thomas Model of Type B

Description

Simulation of the generalized Thomas model of type B.

Usage

```
SimulateTypeB(pars, seed = NULL, parents.distinct = FALSE, plot = TRUE)
```

Arguments

<code>pars</code>	a named vector of containing the values of the model parameters (<code>mu1</code> , <code>mu2</code> , <code>nu</code> , <code>sigma1</code> , <code>sigma2</code>), where (<code>mu_i</code> , <code>nu</code> , <code>sigma_i</code>) is an intensity of parents, an expected number of descendants, a parameter of the dispersal kernel for superposed component i ($i = 1, 2$), respectively.
<code>seed</code>	a positive integer, which is the seed for a sequence of uniform random numbers. The default seed is based on the current time.
<code>parents.distinct</code>	logical. If TRUE, points are distinguished by two groups specified by parameters (<code>mu1</code> , <code>nu</code> , <code>sigma1</code>) and (<code>mu2</code> , <code>nu</code> , <code>sigma2</code>).
<code>plot</code>	logical. If TRUE (default), simulated parent points and offspring points are plotted.

Details

Consider the two types of the Thomas model with parameters (μ_1, ν, σ_1) and (μ_2, ν, σ_2) . Parents' configuration and numbers of the offspring cluster sizes are generated by the two types of uniformly distributed parents (x_i^k, y_i^k) with $i = 1, 2, \dots, Poisson(\mu_k)$ for $k = 1, 2$, respectively.

Then, using series of different uniform random numbers $\{U\}$ for different i and j , each of the offspring coordinates $(x_j^{k,i}, y_j^{k,i})$ of the parents (k, i) with $k = 1, 2$ and $j = 1, 2, \dots, Poisson(\nu)$ is given by

$$x_j^{k,i} = x_i^k + r_k \cos(2\pi U),$$

$$y_j^{k,i} = y_i^k + r_k \sin(2\pi U),$$

where

$$r_k = \sigma_k \sqrt{-2 \log(1 - U_k)}, \quad k = 1, 2,$$

with different random numbers $\{U_k, U\}$ for different k, i , and j .

Value

parents	a list containing two components named "n" and "xy" giving the number and the matrix of (x, y) coordinates of simulated parents points respectively. <code>xy[1:n[1], 1:2]</code> are generated from parameters (μ_1, ν, σ_1) and the remainder are generated from (μ_2, ν, σ_2) .
offspring	a list containing two components named "n" and "xy" giving the number and the matrix of (x, y) coordinates of simulated offspring points respectively. <code>xy[1:n[1], 1:2]</code> are generated from parameters (μ_1, ν, σ_1) and the remainder are generated from (μ_2, ν, σ_2) .

References

U. Tanaka, Y. Ogata and K. Katsura, Simulation and estimation of the Neyman-Scott type spatial cluster models, *Computer Science Monographs* **No.34**, 2008, 1-44. The Institute of Statistical Mathematics.

Examples

```
pars <- c(mu1 = 10.0, mu2 = 40.0, nu = 30.0, sigma1 = 0.01, sigma2 = 0.03)
SimulateTypeB(pars, seed = 257)
```

 SimulateTypeC

Simulation of the Generalized Thomas Model of Type C

Description

Simulation of the generalized Thomas model of type C.

Usage

```
SimulateTypeC(pars, seed = NULL, parents.distinct = FALSE, plot = TRUE)
```

Arguments

pars	a named vector of containing the values of the model parameters $(\mu_1, \mu_2, \nu_1, \nu_2, \sigma_1, \sigma_2)$, where (μ_i, ν_i, σ_i) is an intensity of parents, an expected number of descendants, a parameter of the dispersal kernel for superposed component i ($i = 1, 2$), respectively.
seed	a positive integer, which is the seed for a sequence of uniform random numbers. The default seed is based on the current time.
parents.distinct	logical. If TRUE, points are distinguished by two groups specified by parameters (μ_1, ν_1, σ_1) and (μ_2, ν_2, σ_2) .
plot	logical. If TRUE (default), simulated parent points and offspring points are plotted.

Details

Consider the two types of the Thomas model with parameters (μ_1, ν_1, σ_1) and (μ_2, ν_2, σ_2) . Parents' configuration and numbers of the offspring cluster sizes are generated by the two types of uniformly distributed parents (x_i^k, y_i^k) with $i = 1, 2, \dots, Poisson(\mu_k)$ for $k = 1, 2$, respectively.

Then, using series of different uniform random numbers $\{U\}$ for different i and j , each of the offspring coordinates $(x_j^{k,i}, y_j^{k,i})$, $j = 1, 2, \dots, Poisson(\nu_k)$ of the parents (k, i) with $k = 1, 2$ is given by

$$x_j^{k,i} = x_i^k + r_k \cos(2\pi U),$$

$$y_j^{k,i} = y_i^k + r_k \sin(2\pi U),$$

where

$$r_k = \sigma_k \sqrt{-2 \log(1 - U_k)}, \quad k = 1, 2,$$

with different random numbers $\{U_k, U\}$ for different k, i and j .

Value

parents	a list containing two components named "n" and "xy" giving the number and the matrix of (x, y) coordinates of simulated parents points respectively. xy[1:n[1], 1:2] are generated from parameters (mu1, nu1, sigma1) and the remainder are generated from (mu2, nu2, sigma2).
offspring	a list containing two components named "n" and "xy" giving the number and the matrix of (x, y) coordinates of simulated offspring points respectively. xy[1:n[1], 1:2] are generated from parameters (mu1, nu1, sigma1) and the remainder are generated from (mu2, nu2, sigma2).

References

U. Tanaka, Y. Ogata and K. Katsura, Simulation and estimation of the Neyman-Scott type spatial cluster models, *Computer Science Monographs* **No.34**, 2008, 1-44. The Institute of Statistical Mathematics.

Examples

```
pars <- c(mu1 = 5.0, mu2 = 9.0, nu1 = 30.0, nu2 = 150.0,
         sigma1 = 0.01, sigma2 = 0.05)
SimulateTypeC(pars, seed = 555)
```

Index

*Topic **package**

NScluster-package, 2

*Topic **spatial**

EstimateIP, 3

EstimateThomas, 4

EstimateTypeA, 6

EstimateTypeB, 8

EstimateTypeC, 10

PalmIP, 12

PalmThomas, 13

PalmTypeA, 14

PalmTypeB, 16

PalmTypeC, 17

SimulateIP, 19

SimulateThomas, 20

SimulateTypeA, 21

SimulateTypeB, 23

SimulateTypeC, 24

EstimateIP, 2, 3

EstimateThomas, 2, 4, 14

EstimateTypeA, 2, 6

EstimateTypeB, 2, 8, 16

EstimateTypeC, 2, 10, 18

NScluster (NScluster-package), 2

NScluster-package, 2

PalmIP, 2, 12

PalmThomas, 2, 13

PalmTypeA, 2, 14

PalmTypeB, 2, 16

PalmTypeC, 2, 17

SimulateIP, 2, 19, 21, 22

SimulateThomas, 2, 20

SimulateTypeA, 2, 21

SimulateTypeB, 2, 23

SimulateTypeC, 2, 24