Journal of Statistical Software

May 2021, Volume 98, Issue 6. *doi: 10.18637/jss.v098.i06*

NScluster: An R Package for Maximum Palm Likelihood Estimation for Cluster Point Process Models Using OpenMP

Ushio Tanaka
Osaka Prefecture University

Masami Saga
Indigo Corporation

Junji Nakano
Chuo University

Abstract

NScluster is an R package used for simulation and parameter estimation for Neyman-Scott cluster point process models and their extensions. For parameter estimation, **NScluster** uses the maximum Palm likelihood estimation procedure. As some estimation procedures proposed herein require heavy calculation, **NScluster** can use parallel computation via **OpenMP** and achieve significant speedup in some cases. In this paper, we discuss results obtained using a laptop PC and a shared memory supercomputer. In addition, we examine the performance characteristics of parallel computation via **OpenMP**.

Keywords: Thomas model, inverse-power type model, extended Thomas model, Palm intensity, maximum Palm likelihood estimation, **NScluster**, **OpenMP**, parallel computation, simplex method.

1. Introduction

In this paper, we explain the R package **NScluster** (Tanaka, Saga, and Nakano 2021), available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=NScluster`, which involves the maximum Palm likelihood estimation procedure for Neyman-Scott cluster point process models and their extensions with parallel computation using **OpenMP** technology (Dagum and Menon 1998). The Neyman-Scott cluster point process was originally applied to cosmology problems (Neyman and Scott 1958). Illian, Penttinen, Stoyan, and Stoyan (2008, p. 16) briefly discuss its historical background. Currently, the cluster point process is used in various fields, particularly for modeling point patterns observed in ecological data, such as that used in Cressie (1993, Section 8.2) and Diggle (1983, Appendix: Data).

Tanaka, Ogata, and Katsura (2008a) used Fortran to program the simulation and maximum Palm likelihood estimation procedure for cluster point process models. We developed an R package **NScluster** based on that Fortran program. In the maximum Palm likelihood estimation, some models require a large amount of calculation to estimate their parameters. Thus, we utilize **OpenMP** technology to enable parallel computation (Tanaka *et al.* 2021). Together with the likelihood estimation procedure, the package **NScluster** also provides a simulation procedure for cluster point process models.

Here, we briefly explain the approach to generate the observations followed by Neyman-Scott cluster point processes. First, we generate unobservable cluster centres located according to a homogeneous Poisson point process. Then, each cluster centre generates a random number of descendent points scattered around itself and distributed according to a given density function relative to the distance between each cluster centre and the descendent points associated with the given centre. In **NScluster**, we focus on planar cluster point processes and assume them to be simple, uniform (stationary) and isotropic. We consider several Neyman-Scott cluster point process models, such as the Thomas model, Inverse-power type model and the extended Thomas model of type A. To model a broad range of clustering point pattern data, we also consider the extended Thomas model of type B and C as superposed Neyman-Scott cluster point process models. Note that Shimatani (2010) considered an extended Neyman-Scott cluster point process and its application to ecology. However, his extension differs from our superposition.

The likelihood function for cluster point processes cannot be derived analytically. However, as a pseudo likelihood, the maximum Palm likelihood enables parameter estimation and model selection quantitatively. Note that Palm intensity (Section 3.1) plays an essential role in Palm likelihood. We refer readers to Tanaka, Ogata, and Stoyan (2008b) and Tanaka and Ogata (2014) for parameter estimation and model selection for Neyman-Scott cluster point processes including their extensions and Tanaka *et al.* (2008a) for detailed computational implementation.

In the following, we provide an overview of spatial statistics software. The important reference is "Special Volume: Software for Spatial Statistics" published by the *Journal of Statistical Software*, **63** (2015), which includes several papers primarily focused on R packages. Among them, **spatstat** (Baddeley, Turner, and Rubak 2021) is a frequently cited R package for point pattern analysis. Baddeley and Turner (2005) also referenced several packages, such as those proposed by Ripley (2001) and Peng (2003). The R package **ptproc** (Peng 2003) is based on an earlier version of package **PtProcess** (Harte 2010). In addition, we refer readers to an R package **palm** (Stevenson 2020), which deals with the maximum Palm likelihood estimation procedure for typical Neyman-Scott cluster point process models, such as the Thomas model. However, **NScluster** covers the extended Neyman-Scott cluster point process models, and, because we employ parallel computation, the computational speed of the maximum Palm likelihood estimation using **NScluster** should be faster than that of **palm**.

The remainder of this paper is organized as follows. In Sections 2 and 3, we briefly explain the theoretical framework for our work. Section 2 provides preliminaries and model descriptions, and, in Sections 2.1 and 2.2, we describe the Neyman-Scott cluster point process model and its extension, respectively. In Sections 3.1 and 3.3, we explain Palm intensity from the perspective of its theoretical property and non-parametric estimate, respectively. In Section 3.2, we describe the desired Palm likelihood function. In Section 4, we overview the **NScluster**. The package includes four functions regarding simulation of two Neyman-

Scott cluster point process models in Section 4.1, MPLE in Section 4.2, confidence interval of parameter estimates in Section 4.3, and display of normalized Palm intensity in Section 4.4. In Section 5, we describe the implementation of parallel computation of **OpenMP** in **NScluster**. In Section 6, we discuss the performance and precision of functions to estimate parameters by parallel computation using **OpenMP**. Furthermore, we also discuss the comparison of **NScluster** with **palm**. In Section 7, we discuss an application of the **NScluster** to an ecological data. The conclusions are presented in Section 8.

2. Model descriptions

Essentially, a point process is a stochastic model governing the location of events in a given set (Cressie 1993, p. 619). In this study, we consider the point process in a subset of Euclidean space. A point pattern is considered a realisation of the point process. To analyze the point pattern, we first plot it as observed in the subset, which is considered an observation window denoted W. For simplicity and following the overall preceding study, we assume that the observation window W has been standardized onto the unit square ($W = [0,1] \times [0,1]$). Thus, throughout **NScluster**, we employ a unit square as the observation window. If the real window is a rectangular domain or is irregularly shaped, we select the largest possible square from the window, and consider it as the unit. We assume that W satisfies a periodic boundary condition, i.e., W is considered to be a torus. Treating W is simply a method to resolve the complication of edge effects. See, e.g., Diggle (2003, Section 1.3) for details.

2.1. Neyman-Scott cluster point process model

First, we generate a homogeneous Poisson point process with intensity μ. The generated points are referred to as parent points. The upper left panel of Figure 1 displays a simulation of the parent points. Each parent point generates a random number M of descendent points, which are realized independently and identically. Let ν be the expectation of M. The descendent points are distributed isotropically around each parent point, and the distances between each parent point and its descendent points are distributed independently and identically according to a probability density function (PDF) relative to the distance from a parent point to its descendent point. We call the PDF a dispersal kernel and denote it by q_τ, where τ indicates the parameter set of the dispersal kernel. The Neyman-Scott cluster point process is a union of all descendent points, with the exception of all parent points. In other words, the cluster process is unobservable for each cluster centre. The Neyman-Scott cluster point process is also homogeneous, and its intensity λ equals $\mu\nu$. The parameter set to be estimated is (μ, ν, τ) (Section 3).

In the following, we describe three Neyman-Scott cluster point process models, i.e., the Thomas and Inverse-power type models and the extended Thomas model of type A. In addition, we display their simulations; see the upper right, lower left and lower right panels of Figure 1. The location of their corresponding parent points is common, see the upper left panel of Figure 1.

Thomas model

The Thomas model (Thomas 1949) is the most utilized Neyman-Scott cluster point process model. In this model, descendent points are scattered according to bivariate Gaussian distri-

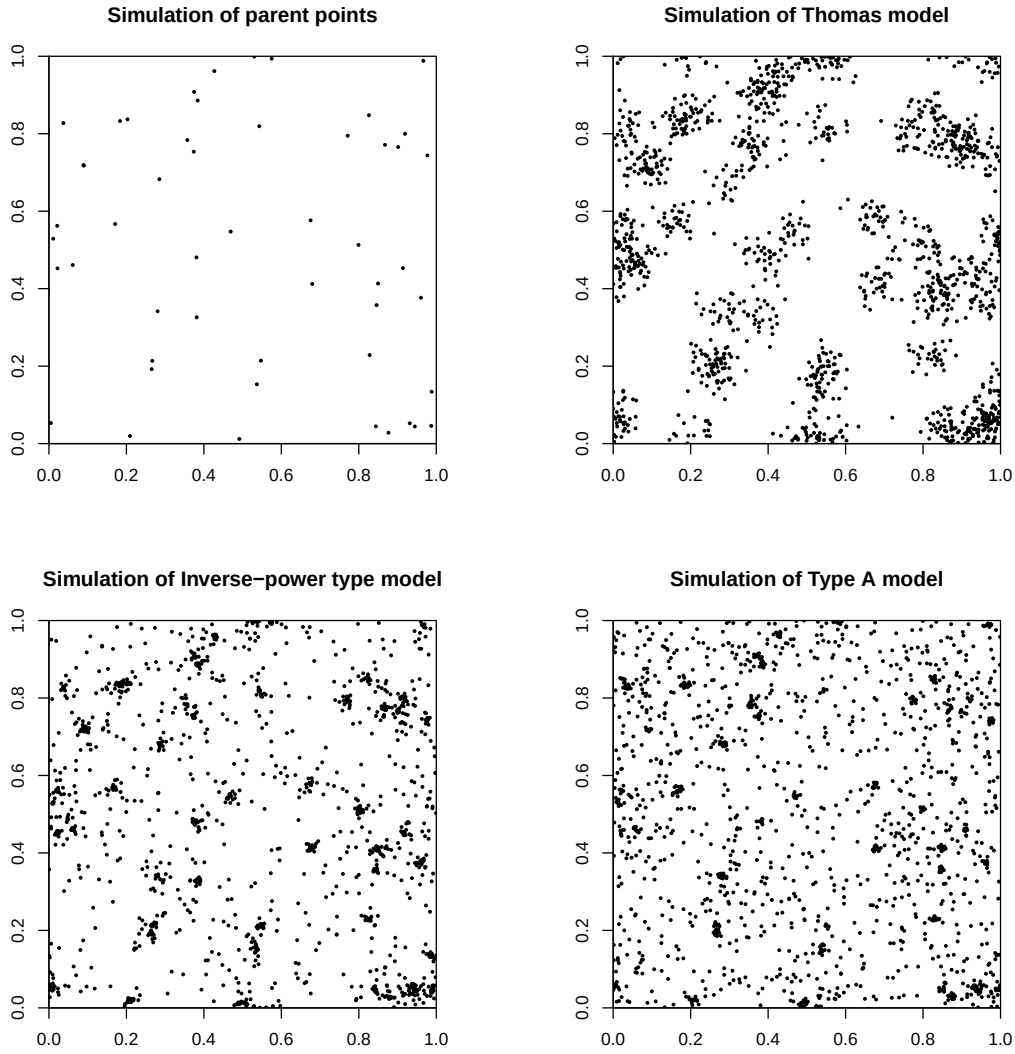

Figure 1: The upper left panel exhibits a simulation of the parent points with $\mu = 50.0$. The upper right, lower left and lower right panels exhibit simulations of the Thomas model with $(\mu, \nu, \sigma) = (50.0, 30.0, 0.03)$, the Inverse-power type model with $(\mu, \nu, p, c) = (50.0, 30.0, 1.5, 0.005)$ and the Type A model with $(\mu, \nu, a, \sigma_1, \sigma_2) = (50.0, 30.0, 0.3, 0.005, 0.1)$, respectively.

bution with zero mean and covariance matrix $\sigma^2 I$, $\sigma > 0$, where I is a 2×2 identity matrix. The corresponding dispersal kernel with $\tau = \sigma$ is given by

$$q_\sigma(r) := \frac{r}{\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right), \quad r \geq 0.$$

The upper right panel of Figure 1 displays a simulation of the Thomas model.

In previous studies that analyzed clustering point pattern data, the Thomas model was fitted to such data because one can explicitly derive classical summary statistics, e.g., Ripley's K-function of the Thomas model, which is closely related to the Palm intensity (Section 3.1).

Inverse-power type model

The Inverse-power type model originated from the frequency of aftershocks per unit time interval (one day, one month, etc.), which has been referred to as the "modified Omori formula" (Ogata 1988). In this model, descendent points are scattered according to Lomax distribution (Lomax 1954). The corresponding dispersal kernel with $\tau = (p, c)$ is given as follows:

$$q_{(p,c)}(r) := \frac{c^{p-1}(p-1)}{(r+c)^p}, \quad r \geq 0, \tag{1}$$

where $p > 1$ and $c > 0$ imply the decay order and scaling with respect to the distance between each parent point and its descendent points, respectively.

The lower left panel of Figure 1 displays a simulation of the Inverse-power type model. From the simulation, it can be inferred that the Inverse-power type model disperses more dense descendent points around parent points than the Thomas model.

Type A model

The extended Thomas model of type A (Type A model for short) is a Neyman-Scott cluster point process model where the dispersal kernel with $\tau = (a, \sigma_1, \sigma_2)$ is mixed by that of two Thomas models as follows:

$$q_{(a,\sigma_1,\sigma_2)}(r) := aq_{\sigma_1}(r) + (1-a)q_{\sigma_2}(r), \quad r \geq 0, \tag{2}$$

where a implies a mixture ratio parameter with $0 < a < 1$.

Conceptually, the location of each descendent point comes from a bivariate normal distribution centred on its parent point. The bivariate normal distribution has the covariance matrix ${\sigma_1}^2 I$ with probability a, and ${\sigma_2}^2 I$ with probability $1 - a$, respectively, where I is in the Thomas model. Arbitrary two descendent points from the same parent point do not necessarily come from the distributions with the same covariance matrix.

From Equation 2, it can be inferred that the Type A model is suitable for densely and vaguely clustering point pattern data to be fitted by mixing the Thomas model with the mixture ratio a. The lower right panel of Figure 1 displays a simulation of the Type A model.

2.2. Superposed Neyman-Scott cluster point process model

The Neyman-Scott cluster point process models can be extended through numerous approaches. Herein, we consider a special case of the superposed Neyman-Scott cluster point process models. In particular, we focus on superposed Thomas models. The parameter sets to be estimated are given by those of two Thomas models: (μ_i, ν_i, σ_i), where $i = 1, 2$. Note that the intensity λ of superposed uniform point processes with intensity λ_i $(= \mu_i \nu_i)$, $i = 1, 2$, is given by $\lambda = \lambda_1 + \lambda_2$.

Type B and C models

We handle two types of the superposed Thomas model, which are referred to as the extended Thomas model of type B (Type B model for short) if $\nu_1 = \nu_2$ and the extended Thomas model of type C (Type C model for short) if $\nu_1 \neq \nu_2$.

Two simulations of the Type B and C models are displayed in the right panels of Figure 2.

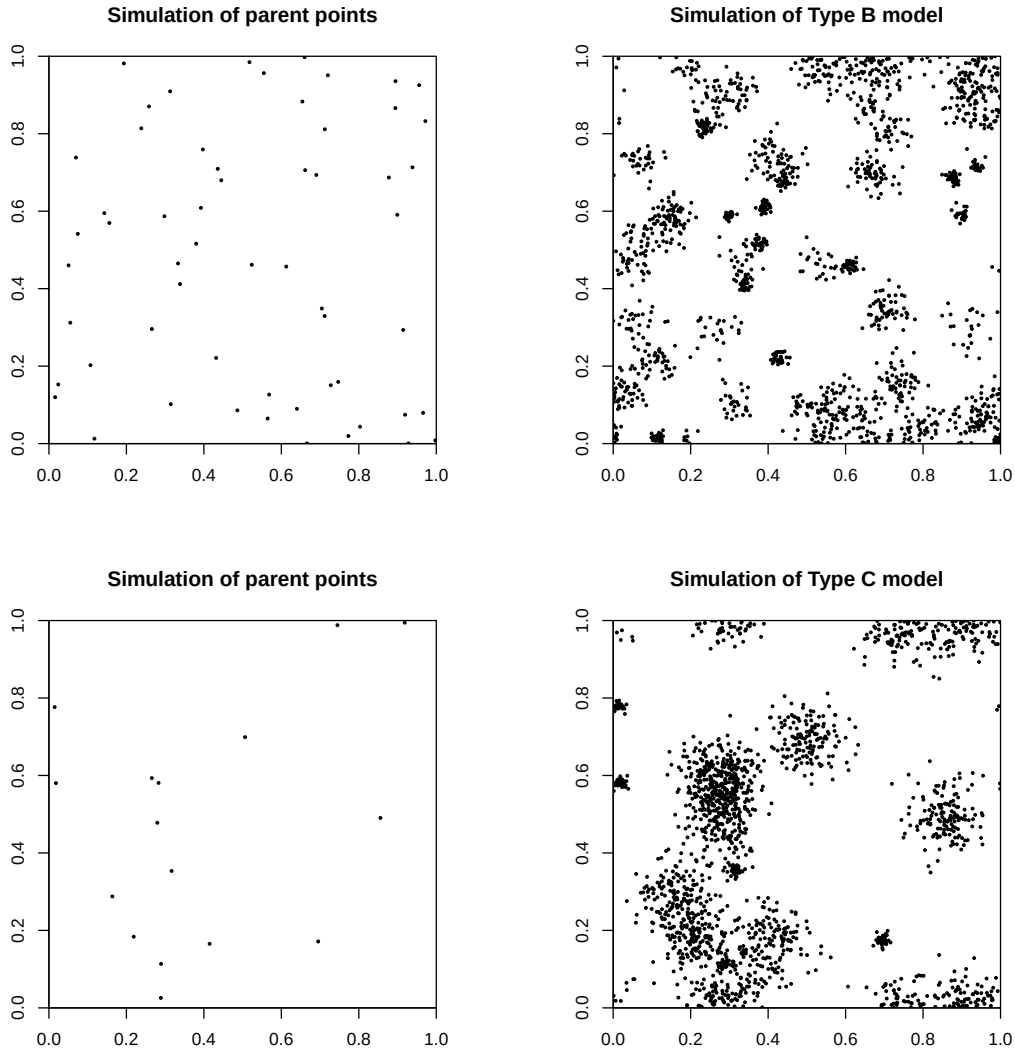

Figure 2: The upper left and right panels present simulation results of the parent points with $(\mu_1, \mu_2) = (10.0, 40.0)$ and the Type B model with $(\mu_1, \mu_2, \nu, \sigma_1, \sigma_2) = (10.0, 40.0, 30.0, 0.01, 0.03)$, respectively. The lower left and right panels show simulation results of the parent points with $(\mu_1, \mu_2) = (5.0, 9.0)$ and the Type C model with $(\mu_1, \mu_2, \nu_1, \nu_2, \sigma_1, \sigma_2) = (5.0, 9.0, 30.0, 150.0, 0.01, 0.05)$, respectively.

3. Maximum Palm likelihood estimation

The maximum Palm likelihood estimation has been recognized as an innovative procedure for parameter estimation for the Neyman-Scott cluster point process models and their superposition. Prior to reviewing this procedure, we introduce the notion of Palm intensity.

3.1. Palm intensity

Henceforth, we assume point processes on W satisfy conditions of local finiteness, simplicity, uniformity and isotropy. Here local finiteness and simplicity for point processes are briefly

reviewed. A locally finite point processes is a model for point patterns with a locally finite number of points. A point process is deemed simple if two points of the corresponding point pattern never coincide. For details, refer to Møller and Waagepetersen (2003). Note that by virtue of uniformity, point processes are homogeneous, i.e., they are of constant intensity.

We begin with a brief overview of the Palm intensity of such point processes, which leads to the log-Palm likelihood function discussed in Section 3.2. The Palm intensity was based on the work of Conrad Palm for the study of telephone traffic (Palm 1943). Translating each point of the given point process into the origin $\boldsymbol{o} \in \mathbb{R}^2$, we obtain a superposed point process at $\boldsymbol{o}$. We call it the difference process. The difference process is symmetric with respect to $\boldsymbol{o}$. The Palm intensity focuses on the difference process induced from pairwise coordinates of the original process.

Let us define the Palm intensity. Our definition of Palm intensity agrees with that of Ogata and Katsura (1991). We denote by N a counting measure, i.e., the total mass of random geometrical objects such as the number of points, lengths of fibres, areas of surfaces and volume of grains within Borel sets. The Palm intensity $\lambda_{\boldsymbol{o}}$ is defined as follows:

$$\lambda_{\boldsymbol{o}}(\boldsymbol{x}) := \frac{\mathsf{P}(\{\, N(\boldsymbol{dx}) \geq 1 \mid N(\{\, \boldsymbol{o} \,\}) = 1 \,\})}{\mathrm{Vol}(\boldsymbol{dx})}, \tag{3}$$

where $\boldsymbol{dx}$ signifies an infinitesimal set containing an arbitrary given point $\boldsymbol{x} \in W$. Refer to Tanaka (2013) for a mathematical argument for the Palm intensity of planar Neyman-Scott cluster point processes. Here, we examine Equation 3. $\lambda_{\boldsymbol{o}}$ implies the occurrence rate at an arbitrary given point $\boldsymbol{x}$ provided that a point is at $\boldsymbol{o}$. Let r be the distance from $\boldsymbol{o}$ to $\boldsymbol{x}$. We see that $\lambda_{\boldsymbol{o}}$ depends only on r. Thus, we obtain its polar coordinate representation with respect to distance r as follows:

$$\lambda_{\boldsymbol{o}}(\boldsymbol{x}) = \lambda_{\boldsymbol{o}}(r, \theta) = \lambda_{\boldsymbol{o}}(r), \quad r \geq 0, \quad 0 \leq \theta < 2\pi.$$

Here, we further assume the point processes to be orderly, i.e., $\mathsf{P}(\{\, N(\boldsymbol{dx}) \geq 2 \,\})$ is of a smaller order of magnitude than $\mathrm{Vol}(\boldsymbol{dx})$. The orderliness allows us to represent the Palm intensity in terms of Ripley's K-function, which is defined as the average number of other points that have appeared within the distance from the typical point (Illian *et al.* 2008, pp. 214–215). In fact, we see that

$$\lambda_{\boldsymbol{o}}(r) = \frac{\lambda}{2\pi r} \frac{dK(r)}{dr}, \quad r \geq 0, \tag{4}$$

where λ is the intensity of the given point process.

For Neyman-Scott cluster point processes, one can compute the right-hand side of Equation 4 to get

$$\lambda_{\boldsymbol{o}}(r) = \lambda + \frac{\nu f_\tau(r)}{2\pi r}, \quad r \geq 0, \tag{5}$$

where f_τ is the PDF relative to the random distance between two descendent points within the same cluster. Let F_τ be the probability cumulative distribution function relative to the

random distance, i.e., $f_\tau(r) = dF_\tau(r)/dr$ for all $r \geq 0$. F_τ takes the following form:

$$\begin{aligned} F_\tau(r) = 2 &\left\{ \int_0^{\frac{r}{2}} \left\{ \int_{r-r_1}^{r+r_1} \frac{1}{\pi} \arccos\left(\frac{{r_1}^2 + {r_2}^2 - r^2}{2r_1 r_2} \right) q_\tau(r_2)\, dr_2 \right\} q_\tau(r_1)\, dr_1 \right. \\ &+ \int_{\frac{r}{2}}^{\infty} \left\{ \int_{r_1}^{r+r_1} \frac{1}{\pi} \arccos\left(\frac{{r_1}^2 + {r_2}^2 - r^2}{2r_1 r_2} \right) q_\tau(r_2)\, dr_2 \right\} q_\tau(r_1)\, dr_1 \\ &\left. + \int_0^{\frac{r}{2}} \left\{ \int_{r_1}^{r-r_1} q_\tau(r_2)\, dr_2 \right\} q_\tau(r_1)\, dr_1 \right\}, \quad r \geq 0, \end{aligned} \tag{6}$$

where r_i, $i = 1, 2$, is the distance from a parent point to its descendent points. For the detailed verification of Equation 6, refer to Stoyan and Stoyan (1994, Section 16.2.2).

To derive the Palm intensity using Equation 5, we need to consider f_τ.

For example, for the Thomas model (i.e., $\tau = \sigma$), one can explicitly derive f_σ as follows:

$$f_\sigma(r) = \frac{r}{2\sigma^2} \exp\left(-\frac{r^2}{4\sigma^2} \right), \quad r \geq 0, \tag{7}$$

from which

$$F_\sigma(r) = 1 - \exp\left(-\frac{r^2}{4\sigma^2} \right), \quad r \geq 0.$$

Combining Equation 7 with Equation 5, we get the Palm intensity $\lambda_{\boldsymbol{o}}$ of the Thomas model:

$$\lambda_{\boldsymbol{o}}(r) = \lambda + \frac{\nu}{4\pi\sigma^2} \exp\left(-\frac{r^2}{4\sigma^2} \right), \quad r \geq 0. \tag{8}$$

Using the argument of Illian *et al.* (2008, p. 220 and Section 6.2.3), one can obtain from Equation 8 each individual Palm intensity of the Type B and C models as follows: For the Type C model,

$$\lambda_{\boldsymbol{o}}(r) = \lambda + \frac{1}{4\pi} \left(\frac{a_1 \nu_1}{{\sigma_1}^2} \exp\left(-\frac{r^2}{4{\sigma_1}^2} \right) + \frac{a_2 \nu_2}{{\sigma_2}^2} \exp\left(-\frac{r^2}{4{\sigma_2}^2} \right) \right), \quad r \geq 0, \tag{9}$$

where $a_i := \lambda_i/\lambda$, $i = 1, 2$, see Section 2.2 for notation.

For the Type B model, especially if $\nu_1 = \nu_2 =: \nu$, Equation 9 meets $\lambda_{\boldsymbol{o}}$ of the Type B model, i.e.,

$$\lambda_{\boldsymbol{o}}(r) = \lambda + \frac{\nu}{4\pi} \left(\frac{a_1}{{\sigma_1}^2} \exp\left(-\frac{r^2}{4{\sigma_1}^2} \right) + \frac{a_2}{{\sigma_2}^2} \exp\left(-\frac{r^2}{4{\sigma_2}^2} \right) \right), \quad r \geq 0.$$

For the Inverse-power type and Type A models, we need to compute Equation 6 because it is unable to derive their f_τ's analytically. In Section 5, we will discuss the numerical computation of Equation 6.

3.2. Palm likelihood function

It is impossible to specify an exact likelihood function in an analytically closed form for Neyman-Scott cluster point processes and their extensions owing to the following difficulties:

the dataset does not contain any parent points, the relationship between descendent points and the attribution of their parent points are not specified in the given dataset and the ranges of each cluster overlap such that their ranges are non-specific.

Therefore, we must propose a pseudo maximum likelihood estimation procedure for the cluster point processes, i.e., maximum Palm likelihood estimation. The maximum Palm likelihood estimation procedure is based on the assumption that the difference process is well approximated by an isotropic and inhomogeneous Poisson point process with intensity function $N(W)\lambda_{\boldsymbol{o}}(r)$, which is centred at $\boldsymbol{o}$.

Now, we are positioned to state the log-Palm likelihood function. Let $\boldsymbol{\theta}$ denote the parameter set of the cluster point process models. The log-Palm likelihood function, denoted $\ln L$ based on the Palm intensity $\lambda_{\boldsymbol{o}}$ (including $\boldsymbol{\theta}$) is given as follows:

$$\ln L(\boldsymbol{\theta}) = \sum_{i,j;i<j,\, 0<r_{ij}\leq R} \ln\left(N(W)\lambda_{\boldsymbol{o}}(r_{ij})\right) - 2\pi N(W)\int_0^R \lambda_{\boldsymbol{o}}(r) r\, dr. \tag{10}$$

Here, the summation is taken over each pair (i, j) with $i < j$ such that the distance r_{ij} between distinct points x_i and x_j of the cluster point processes satisfies $0 < r_{ij} \leq R$, where R is greater than or equal to the range of correlation. The distance is measured with respect to periodic boundary condition. The range of correlation is defined as follows: if there is a finite distance $r_o \geq 0$ such that $\lambda_{\boldsymbol{o}}(r) = \lambda$ for all $r \geq r_o$, r_o is referred to as the range of correlation. The range of correlation implies that there are no correlations between distinct points of point processes whose distances are greater than r_o. For details, refer to Illian *et al.* (2008, p. 220). Note that $i < j$ in Equation 10 is due to the symmetry of difference processes. Strictly speaking, from the periodic boundary condition for W, it follows that $r_o \leq 1/2$ when W is a unit square. Herein, we consider $R = 1/2$. Refer to Tanaka (2013) for a mathematical argument regarding the range of correlation.

The maximum Palm likelihood estimates (MPLEs for short) are those that maximize Equation 10. Prokešová and Vedel Jensen (2012) verified the asymptotic properties of MPLEs.

To facilitate wide application of the maximum Palm likelihood estimation procedure to several cluster point process models, we must rewrite Equation 10 because, as mentioned in Section 3.1, generally, the Palm intensity of cluster point processes cannot be derived analytically. Combining Equations 10 and 5, we obtain the following expression:

$$\ln L(\boldsymbol{\theta}) = \sum_{i,j;i<j,\, 0<r_{ij}\leq 1/2} \ln\left(\lambda + \frac{\nu f_\tau(r_{ij})}{2\pi r_{ij}}\right) - N(W)(\pi\lambda/4 + \nu F_\tau(1/2)). \tag{11}$$

Note that maximising $\ln L(\boldsymbol{\theta})$ in Equation 10 to obtain MPLEs, $N(W)$ assigning the non-parametric part of Equation 10 is removable. Therefore, Equation 11 is handled as shown above. Equation 11 can be evaluated by computing F_τ and its derivative f_τ in Equation 11 using **NScluster**. See the R code described in Section 5.

3.3. Non-parametric estimation

To determine the adequacy of MPLEs, **NScluster** provides users with a non-parametric estimation of the Palm intensity. Reviewing Ogata and Katsura (1991), we outline the non-parametric Palm intensity. First, let us count the number of points of a point pattern given

in each annular set with uniform width. One can see that the following quantity meets the non-parametric estimate for the Palm intensity. Let $\Delta(r)$ be a disk of radius $r > 0$. We obtain the following: for a sufficiently small $\delta > 0$,

$$\frac{1}{N(W)} \left(\frac{N(\Delta(r+\delta) \setminus \Delta(r))}{\mathrm{Vol}(\Delta(r+\delta)) - \mathrm{Vol}(\Delta(r))} \right). \tag{12}$$

Note that $\Delta(r+\delta) \setminus \Delta(r)$ is an annular set with uniform width δ. The leftmost $1/N(W)$ in Equation 12 is due to $N(\{\boldsymbol{o}\}) = 1$ in Equation 3.

4. Overview of NScluster

The package **NScluster** comprises of four tasks, i.e., simulation, MPLE, confidence interval estimation and non-parametric and parametric normalized Palm intensity comparison.

4.1. Simulation

The first and most intuitive step to understand the model characteristics is to observe the data generated by the model. This can be realized using `sim.cppm`.

For example, data based on the Thomas model can be generalized as follows:

```
R> pars <- c(mu = 50.0, nu = 30.0, sigma = 0.03)
R> cppThomas <- sim.cppm("Thomas", pars, seed = 353)
R> cppThomas

Number of parent points = 51.0
Total number of offspring points = 1494.0

R> plot(cppThomas)
```

This code generates the upper right panel of Figure 1. Here, `mu`, `nu` and `sigma` indicate the intensity of parent points, the mean number of descendent points per parent point and a parameter of the dispersal kernel of the Thomas model, respectively (Section 2.1). In addition, `seed` is specified to set the random number seed for the Mersenne-Twister random number generator.

In the same manner, for the Inverse-power type model, we must specify the parameter set (`p`, `c`) of the dispersal kernel given in Equation 1. For example, consider the following code to generate the lower left panel of Figure 1.

```
R> parsIP <- c(mu = 50.0, nu = 30.0, p = 1.5, c = 0.005)
R> cppIP <- sim.cppm("IP", parsIP, seed = 353)
R> cppIP

Number of parent points = 51.0
Total number of offspring points = 1494.0

R> plot(cppIP)
```

4.2. MPLE

We can fit models when we have data, e.g., we can use the `mple.cppm` function to estimate cluster point process models. This function can estimate parameter values from the given initial parameter sets using the simplex method (Kowalik and Osborne 1968) to maximize the log-Palm likelihood function of Equation 11. If initial values are not specified, they are automatically given as the default values calculated from non-parametric Palm intensity. As the default initial values do not always give good values, we strongly suggest using several initial parameter sets manually. Example R code to estimate model parameters from the data generated by the above code is given below.

```
R> xy.IP <- cppIP$offspring$xy
R> initp.IP <- c(mu = 55.0, nu = 35.0, p = 1.0, c = 0.01)
R> mpleIP <- mple.cppm("IP", xy.IP, initp.IP, skip = 100)
R> summary(mpleIP)

Inverse-power type model
The number of point pattern:   1494

         MPLE
mu   48.31433
nu   30.50470
p     1.64667
c     0.00795

Log(MPL):   11189076.362
AIC:       -22378144.724

R> plot(mpleIP)
```

In the above computer output, `Log(MPL)` stands for the maximum of the log-Palm likelihood function (Equations 10 and 11).

We must compute Equation 11 to obtain MPLEs; however, the summation computation relative to (i, j) in Equation 11 is time consuming. Thus, we skip some pairs of (i, j) and specify the skip rate using the `skip` argument to reduce the computational burden. Specifically, all pairs of (i, j) are appropriately ordered and every `skip` pair is used for calculation.

An example of improving processes for the parameter set (`mu`, `nu`, `p`, `c`) is shown in Figure 3. Note that we show values that estimated parameters are divided by their initial parameters to illustrate their convergence in each iteration. As can be seen, all parameters converge at around 100 iterations.

We observed that it took more than 10 minutes for the parameter values to converge, which is much longer than the Thomas model estimation. We can reduce the calculation time by parallelising the computation.

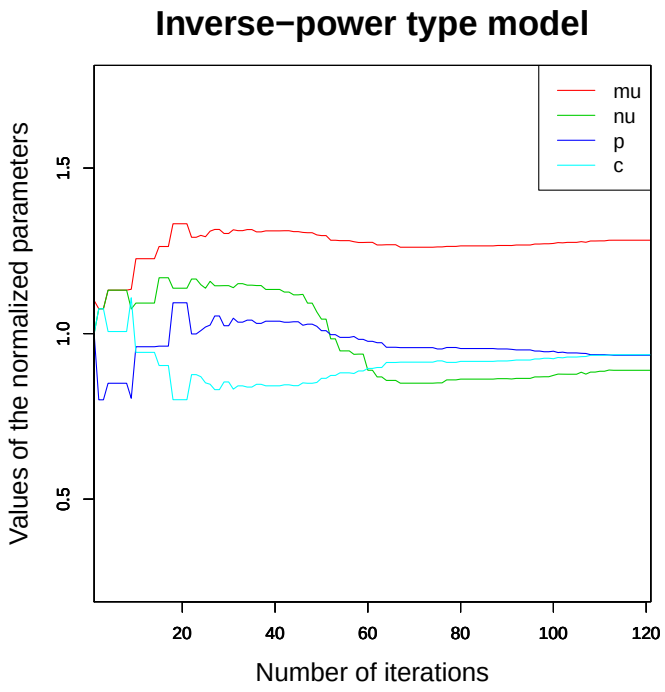

Figure 3: Computational behavior of the convergence of parameter set (μ, ν, p, c) of the Inverse-power type model.

4.3. Confidence interval of parameter estimates

We developed a confidence interval of parameters using bootstrap method. When we estimate one model, we generate simulated data several times for the estimated model, then, we estimate the parameters repeatedly. The empirical distribution of given parameters can be used to decide the interval estimation of the parameter.

For example, the data generated from the Thomas model given in Section 4.1 is used to demonstrate the interval estimation of the parameters using the function `boot.mple`.

```
R> xy.Thomas <- cppThomas$offspring$xy
R> initp.Thomas <- c(mu = 40, nu = 40, sigma = 0.05)
R> mpleThomas <- mple.cppm("Thomas", xy.Thomas, initp.Thomas)
R> set.seed(12345)
R> bootThomas <- boot.mple(mpleThomas)
R> summary(bootThomas)
```

```
              MPLE        2.5 %       97.5 %       std.err
mu     44.04948235  21.93542495  67.00781201  1.2551551272
nu     33.40395790  22.59024031  55.37621730  1.3403600766
sigma   0.02843395   0.02371928   0.04179367  0.0006343702
```

Note that, for some models, the execution of `boot.mple` requires considerable time owing to extensive calculations.

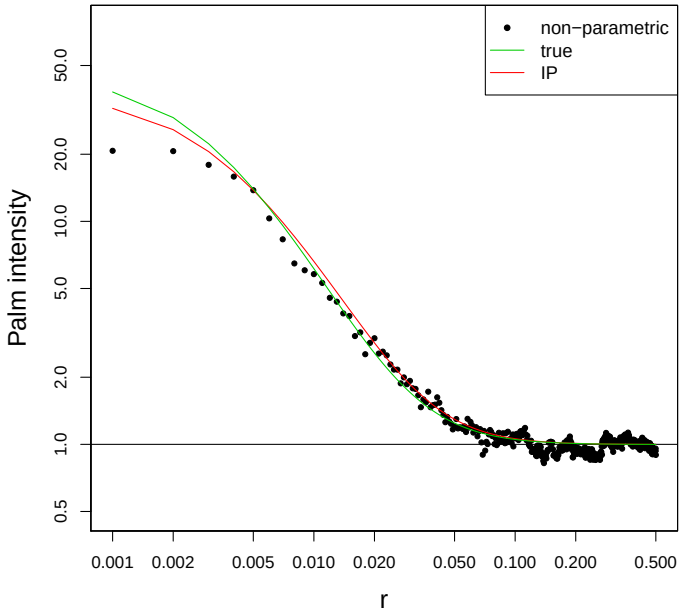

Figure 4: Palm intensity of the Inverse-power type model corresponding to the respective parameters.

4.4. Display of normalized Palm intensity

The package **NScluster** can depict the estimates of the normalized Palm intensity, i.e., $\lambda_{\boldsymbol{o}}(r)/\lambda$, which is just the pair correlation function. Here, we consider the Inverse-power type model, whose Palm intensity cannot be derived analytically. Therefore, to obtain the MPLEs, Equation 11 must be computed rather than Equation 10. For this purpose, the R code is as follows:

```
R> palmIP <- palm.cppm(mpleIP, parsIP)
R> plot(palmIP)
```

Arguments `mpleIP` and `parsIP` specify the output object from the `mple.cppm` and the parameters given in Section 4.1, respectively.

Henceforth, we use the word "Palm intensity" to denote "normalized Palm intensity" for simplicity. In Figure 4, the green curve represents Palm intensity with true parameters given by `parsIP`. The red curve represents Palm intensity with the parameters given by `mpleIP$mple`. Dots represent a non-parametric Palm intensity. These values are plotted in logarithmic scales. As can be seen in Figure 4, the true and estimated Palm intensities coincide well.

5. Parallel computation implementation

The computationally intensive part of the estimation of model parameters can be parallelized to reduce calculation time. The package **NScluster** was implemented to employ **OpenMP**,

which is a simple framework for shared memory parallel computation. We refer readers to Feng and Tierney (2011) for a concise introduction to **OpenMP**.

Here, we demonstrate the implementation of **OpenMP** in **NScluster** to speed up the `mple.cppm` procedure. We know that the most time-consuming part of the original Fortran code (given below) is calculating the log-Palm likelihood function of the Inverse-power type model (Equation 11),

```
sum = 0.0
lambda = mu*nu
nu2pi = nu/2/pi
do 30 i = 1, nn
  call ippower(r(i), Frmax, dFr)
  f = lambda + nu2pi*dFr/r(i)
  if(f .le. 0.0) go to 190
  sum = sum + log(f)
30 continue
```

The variable `dFr` in the above code signifies f_τ in Equation 5. We numerically compute F_τ using the method presented by Press, Teukolsky, Flannery, and Vetterling (1992). The `if` sentence in the above code is for error handling. This part can be parallelized rather easily using **OpenMP** directives. We add two directive lines to the original code as follows:

```
ier = 0
!$omp parallel do private(dFr, f) reduction(+:sum)
do 30 i = 1, nn
  call ippowerMP(r(i), Frmax, dFr)
  f = lambda + nu2pi*dFr/r(i)
  if(f .le. 0.0) then
    ier = -1
  else
    sum = sum + log(f)
  end if
30 continue
!$omp end parallel do
if(ier .eq. -1) go to 190
```

Error handling becomes a little complicated. The part inside the directive is performed in many threads simultaneously. If an error occurs in one thread, i.e., `ier` is set to -1, it causes error handling. For **OpenMP** to function properly, we must also rewrite the original Fortran code for the `ippower` subroutine:

```
subroutine ippower(ri, Fr, dFr)
real(8) :: ri, Fr, dFr
            ...
integer :: kk
real(8) :: r0
common/distance/r0
common/case/kk
```

as follows:

```
subroutine ippowerMP(ri, Fr, dFr)
real(8) :: ri, Fr, dFr
             ...
integer :: kk
real(8) :: r0
common/distancep/r0
common/casep/kk
!$omp threadprivate(/distancep/)
!$omp threadprivate(/casep/)
```

These threadprivate directives declare that common blocks are private for each thread.

6. Performance and precision of parallel computation

The program discussed in Section 4.2 was executed on a laptop PC (Intel Core i7 6700HQ) and the SGI UV2000 (Intel Xeon E5-2650v2) shared memory supercomputer (named ISM-A) at the Institute of Statistical Mathematics. The number of threads of parallel computation was controlled by the environment variable `OMP_NUM_THREADS`, refer to Feng and Tierney (2011). If `OMP_NUM_THREADS` is not specified explicitly, all available cores are used for calculation. For example, to change the number of **OpenMP** threads using a bash shell on Linux, type `export OMP_NUM_THREADS=n` prior to starting R, where `n` is the number of threads to use. For macOS, the same command works in a terminal window. For Windows, the `set` command can be used instead of the `export` command if a command prompt window is used. The speedup obtained by **OpenMP** as measured by the `system.time` function is shown in Figure 5.

An increased number of threads in **OpenMP** does not always yield better results. In fact, as can be seen, the method obtained the best results with around 16 threads.

Different parallelisation can affect the calculation results slightly. For example, the following R code was executed for different number of threads.

```
R> pars <- c(mu1 = 10.0, mu2 = 40.0, nu = 30.0, sigma1 = 0.01, sigma2 = 0.03)
R> cppTypeB <- sim.cppm("TypeB", pars, seed = 257)
R> xy.TypeB <- cppTypeB$offspring$xy
R> initp.TypeB <- c(mu1 = 20.0, mu2 = 30.0, nu = 30.0, sigma1 = 0.02,
+    sigma2 = 0.02)
R> mpleTypeB <- mple.cppm("TypeB", xy.TypeB, initp.TypeB)
R> coef(mpleTypeB)
```

The results are summarized in Table 1.

The difference is primarily based on the floating-point calculation mechanism; however, the influence on the results is potentially negligible as can be seen in the table.

Finally, we compare two packages **NScluster** with **palm** to see their similarity and dissimilarity. The Neyman-Scott cluster point pattern data generated by the above code in Section 4.1 is used. We perform parameter estimation with 8 threads using the following R code of **NScluster**.

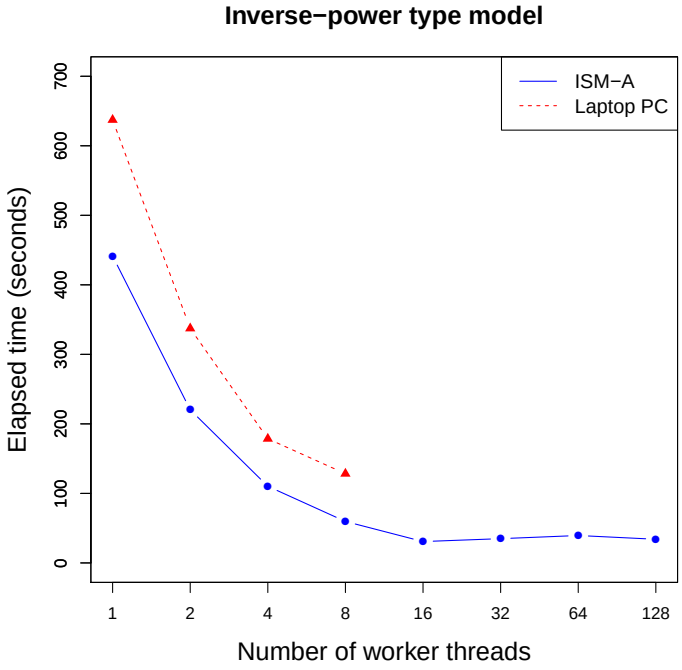

Figure 5: Effectiveness of parallel computation.

	Parameters				
# threads	mu1	mu2	nu	sigma1	sigma2
1	16.17442591	44.36274392	28.40971041	0.01008415	0.03122518
2	16.17777568	44.39743498	28.39416344	0.01007905	0.03119565
4	16.17538671	44.40146193	28.39339402	0.01007858	0.03119297
8	16.17777568	44.39743498	28.39416344	0.01007905	0.03119565

Table 1: Different outputs of MPLEs of Type B model, which is dependent on the number of threads.

```
R> startv <- c(40, 40, 0.05)
R> system.time(mpleThomas <- mple.cppm("Thomas", xy.Thomas, startv))
R> coef(mpleThomas)
```

`fit.ns` function of **palm** can conduct similar analysis via the following R code using a single thread.

```
R> system.time(fit <- fit.ns(xy.Thomas, lims = rbind(c(0, 1), c(0, 1)),
+    R = 0.5, start = startv))
R> coef(fit)
```

Table 2 summarizes the results, where some parameter names for the Thomas model are different for each package. We note that estimated parameters are nearly identical for all parameters. However, elapsed times are different because of the difference in thread numbers.

	mu	nu	sigma	elapsed time
NScluster	44.04948235	33.40395790	0.02843395	0.64
	D	lambda	sigma	elapsed time
palm	44.05379224	33.39956626	0.02843696	4.09

Table 2: Results on **NScluster** and **palm**.

7. Application of NScluster to ecological data

In this section, we apply **NScluster** to ecological data, which are the locations of 359 newly emergent bramble canes shown in Figure 6. The points are presented here in the unit square (Diggle 1983, Appendix: Data), whereas the original data were collected in a 9 m × 9 m square (Hutchings 1978).

Tanaka *et al.* (2008b) fitted the five cluster point process models to the bramble canes data, and estimated their respective parameters via the maximum Palm likelihood estimation procedure. We re-analyze the data by **NScluster** using 8 threads.

```
R> canes <- read.table("BrambleCanes.txt")
R> model <- c("Thomas", "IP", "TypeA", "TypeB", "TypeC")
R> for (mtype in model) {
+    cat(mtype, "\n")
+    cmd1 <- paste("print(system.time(mple", mtype, " <-
+        mple.cppm(", "'", mtype, "'", ", canes)))", sep = "")
+    eval(parse(text = cmd1))
+  }
R> cmd2 <- paste("summary(mple", model,")", sep = "")
R> eval(parse(text = cmd2))
R> cmd3 <- paste("palm",model, " <- palm.cppm(mple", model,")", sep = "")
R> eval(parse(text = cmd3))
R> plot(palmThomas, palmIP, palmTypeA, palmTypeB, palmTypeC)
```

We can obtain the MPLEs together with calculation time, AIC and the figure of Palm intensities by executing above R code.

Herein, as described in Section 3.2, we note that for the cluster point process models, specifying an exact likelihood function in an analytically closed form is impossible. Therefore, for model selection, following Tanaka *et al.* (2008b), we employ AIC replacing the ordinary log-likelihood with the log-Palm likelihood.

The Type A model takes the longest elapsed time (927.256 s.) and the Thomas model has the shortest elapsed time (0.240 s.).

We note that these five cluster models provide a result nearly identical to that of Tanaka *et al.* (2008b). In fact, the minimum AIC is given by the Type B and C models, i.e., the best fit is attained using these two models (Figure 7). This demonstrates the identification problem. For details, refer to Tanaka and Ogata (2014).

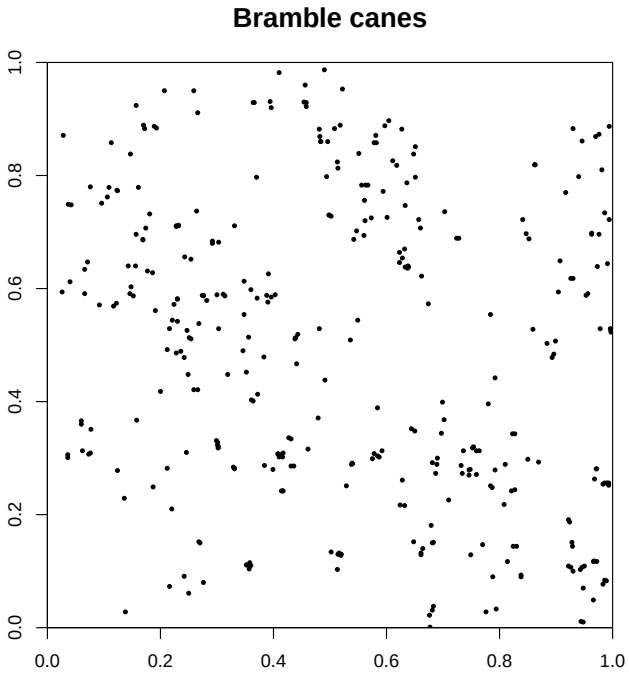

Figure 6: Locations of 359 newly emergent bramble canes in W.

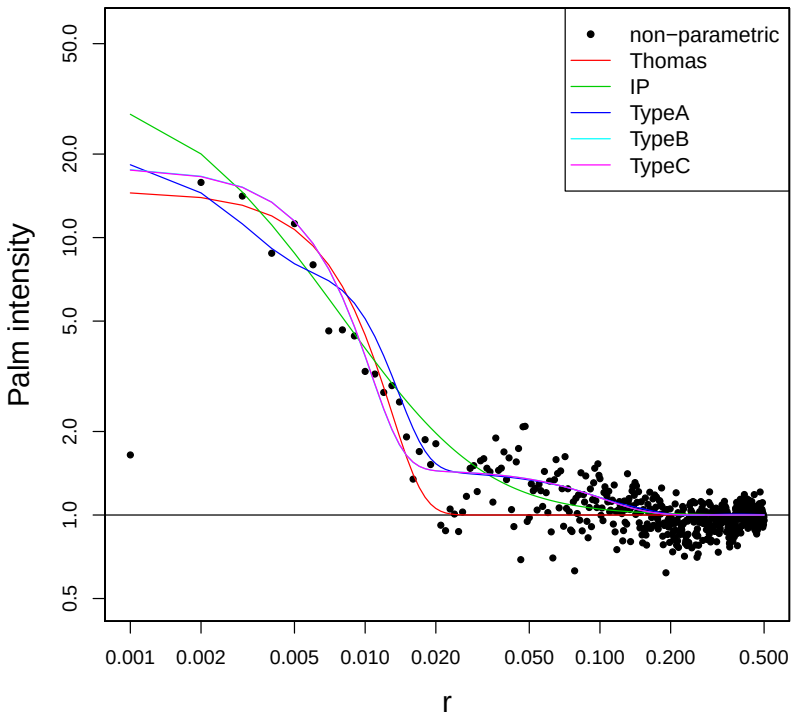

Figure 7: Palm intensities of the respective models.

8. Concluding remarks

The package **NScluster** provides users with functions to simulate Neyman-Scott cluster point process models, such as the Thomas model, the Inverse-power type model and several extended Thomas models, to estimate their parameter set using the Palm likelihood procedure and illustrate their Palm intensities. The computation of MPLEs is implemented by simplex maximisation with parallel computation via **OpenMP**.

It is straightforward to parallelize the original Fortran code using **OpenMP** directives, and we have shown that the parallelisation can increase calculation speed, especially when the number of threads is not so large. The results indicate that it is not useful to overly increase the number of threads, e.g., beyond 16 threads. This is a typical situation for shared memory parallel computation. When the number of threads is increased excessively, the communication among threads incurs significant cost and the total computation time increases.

In future work, we will attempt to parallelize `boot.mple` using a package such as **snow** (Tierney, Rossini, Li, and Sevcikova 2018). Such parallelisation is particularly useful for cluster computer systems, if it is used together with the **OpenMP** parallelized `mple.cppm`.

Acknowledgements

The first author was supported in part by Grant-in-Aid for Young Scientists (B) Grant Number JP25730022 and Grant-in-Aid for Scientific Research (C) Grant Number JP19K11865 from the Japan Society for the Promotion of Science (JSPS). This study was performed under the ISM General Cooperative Research 2 (2016 ISM·GCR-29), (2018 ISM·GCR-30) and (2019-ISMCRP-2059). We would like to thank the editors and anonymous referees for their thoughtful and constructive comments.

References

Baddeley A, Turner R (2005). "**spatstat**: An R Package for Analyzing Spatial Point Patterns." *Journal of Statistical Software, Articles*, **12**(6), 1–42. doi:10.18637/jss.v012.i06.

Baddeley A, Turner R, Rubak E (2021). ***spatstat**: Spatial Point Pattern Analysis, Model-Fitting, Simulation, Tests*. R package version 2.1-0, URL https://CRAN.R-project.org/package=spatstat.

Cressie NAC (1993). *Statistics of Spatial Data*. Revised edition. John Wiley & Sons. doi:10.1002/9781119115151.

Dagum L, Menon R (1998). "**OpenMP**: an industry standard API for shared-memory programming." *IEEE Computational Science and Engineering*, **5**(1), 46–55. doi:10.1109/99.660313.

Diggle P (1983). *Statistical Analysis of Spatial Point Patterns*. 1st edition. Academic Press.

Diggle P (2003). *Statistical Analysis of Spatial Point Patterns*. 2nd edition. Arnold.

Feng D, Tierney L (2011). "**mritc**: A Package for MRI Tissue Classification." *Journal of Statistical Software*, **44**(7), 1–20. doi:10.18637/jss.v044.i07.

Harte D (2010). "**PtProcess**: An R Package for Modelling Marked Point Processes Indexed by Time." *Journal of Statistical Software*, **35**(8), 1–32. doi:10.18637/jss.v035.i08.

Hutchings MJ (1978). "Standing Crop and Pattern in Pure Stands of Mercurialis Perennis and Rubus Fruticosus in Mixed Deciduous Woodland." *Oikos*, **31**(3), 351–357. doi:10.2307/3543662.

Illian J, Penttinen A, Stoyan H, Stoyan D (2008). *Spatial Analysis and Modelling of Spatial Point Patterns*. John Wiley & Sons. doi:10.1002/9780470725160.

Kowalik JS, Osborne MR (1968). *Methods for Unconstrained Optimization Problems*. American Elsevier Publishing Company.

Lomax KS (1954). "Business Failures: Another Example of the Analysis of Failure Data." *Journal of the American Statistical Association*, **49**(268), 847–852. doi:10.2307/2281544.

Møller J, Waagepetersen RP (2003). *Statistical Inference and Simulation for Spatial Point Processes*. Chapman & Hall/CRC. doi:10.1201/9780203496930.

Neyman J, Scott EL (1958). "Statistical Approach to Problems of Cosmology." *Journal of the Royal Statistical Society B*, **20**(1), 1–29. doi:10.1111/j.2517-6161.1958.tb00272.x.

Ogata Y (1988). "Statistical Models for Earthquake Occurrences and Residual Analysis for Point Processes." *Journal of the American Statistical Association*, **83**(401), 9–27. doi:10.1080/01621459.1988.10478560.

Ogata Y, Katsura K (1991). "Maximum Likelihood Estimates of the Fractal Dimension for Random Spatial Patterns." *Biometrika*, **78**(3), 463–474. doi:10.1093/biomet/78.3.463.

Palm C (1943). "Intensitätsschwankungen Im Fernsprechverkehr." *Technical report*, Ericsson Technics.

Peng R (2003). "Multi-Dimensional Point Process Models in R." *Journal of Statistical Software*, **8**(16), 1–27. doi:10.18637/jss.v008.i16.

Press WH, Teukolsky SA, Flannery BP, Vetterling WT (1992). *Numerical Recipes in Fortran 77*. 2nd edition. Cambridge University Press.

Prokešová M, Vedel Jensen EB (2012). "Asymptotic Palm Likelihood Theory for Stationary Point Processes." *Annals of the Institute of Statistical Mathematics*, **65**(2), 387–412. doi:10.1007/s10463-012-0376-7.

Ripley BD (2001). "Spatial Statistics in R." *R News*, **1**(2), 14–15. URL https://CRAN.R-project.org/doc/Rnews/.

Shimatani IK (2010). "Spatially Explicit Neutral Models for Population Genetics and Community Ecology: Extensions of the Neyman-Scott Clustering Process." *Theoretical Population Biology*, **77**(1), 32–41. doi:10.1016/j.tpb.2009.10.006.

Stevenson B (2020). ***palm**: Fitting Point Process Models via the Palm Likelihood*. R package version 1.1.4, URL https://CRAN.R-project.org/package=palm.

Stoyan D, Stoyan H (1994). *Fractals, Random Shapes and Point Fields: Methods of Geometrical Statistics*. John Wiley & Sons.

Tanaka U (2013). "Remark on the Palm Intensity of Neyman-Scott Cluster Point Processes." *International Journal of Applied Mathematics*, **26**(4), 433–445. doi:10.12732/ijam.v26i4.3.

Tanaka U, Ogata Y (2014). "Identification and Estimation of Superposed Neyman-Scott Spatial Cluster Processes." *Annals of the Institute of Statistical Mathematics*, **66**(4), 687–702. doi:10.1007/s10463-013-0431-z.

Tanaka U, Ogata Y, Katsura K (2008a). *Simulation and Estimation of the Neyman-Scott Type Spatial Cluster Models*. Number 34 in Computer Science Monographs. The Institute of Statistical Mathematics, Tokyo. URL https://www.ism.ac.jp/editsec/csm/.

Tanaka U, Ogata Y, Stoyan D (2008b). "Parameter Estimation and Model Selection for Neyman-Scott Point Processes." *Biometrical Journal*, **50**(1), 43–57. doi:10.1002/bimj.200610339.

Tanaka U, Saga M, Nakano J (2021). ***NScluster**: Simulation and Estimation of the Neyman-Scott Type Spatial Cluster Models*. R package version 1.3.5, URL https://CRAN.R-project.org/package=NScluster.

Thomas M (1949). "A Generalization of Poisson's Binomial Limit For Use in Ecology." *Biometrika*, **36**(1/2), 18–25. doi:10.2307/2332526.

Tierney L, Rossini AJ, Li N, Sevcikova H (2018). ***snow**: Simple Network of Workstations*. R package version 0.4-3, URL https://CRAN.R-project.org/package=snow.

Affiliation:

Ushio Tanaka
Department of Mathematical Sciences
Osaka Prefecture University
1-1 Gakuen-cho, Naka-ku, Sakai-shi, Osaka 599-8531, Japan
E-mail: utanaka@mi.s.osakafu-u.ac.jp

Masami Saga
Indigo Corporation
2-11-22 Sangenjaya, Setagaya-ku, Tokyo 154-0024, Japan
E-mail: msaga@mtb.biglobe.ne.jp

Junji Nakano
Faculty of Global Management
Chuo University
742-1 Higashinakano, Hachioji-shi, Tokyo 192-0393, Japan
E-mail: nakanoj@tamacc.chuo-u.ac.jp

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
May 2021, Volume 98, Issue 6 *Submitted:* 2018-04-20
doi:10.18637/jss.v098.i06 *Accepted:* 2020-02-07