

COMPUTER SCIENCE MONOGRAPHS

*A Publication
of
The Institute of Statistical Mathematics*

**SIMULATION AND ESTIMATION OF THE
NEYMAN-SCOTT TYPE SPATIAL CLUSTER MODELS**

by

U. Tanaka, Y. Ogata and K. Katsura

March 2008

Computer Science Monographs

Editors

Junji NANAOKO(*in chief*)

Satoshi ITO

Nobuhisa KASHIWAGI

Yumi TAKIZAWA

The series of Computer Science Monographs publishes the statistical software and the articles on computer applications originating at The Institute of Statistical Mathematics. All communications relating to this publication should be addressed to the Editorial Office, The Institute of Statistical Mathematics, 4-6-7 Minami-Azabu, Minato-ku, Tokyo 106-8569, Japan.

The electronic version of this issue is downloadable at
<http://www.ism.ac.jp/>

Simulation and estimation of the Neyman-Scott type spatial cluster models

Tanaka, U., Ogata, Y. and Katsura, K.

The Institute of Statistical Mathematics

Contents

1	Overview of the models	2
2	Simulation of the models	3
2.1	Thomas model	3
2.1.1	Simulation method	4
2.1.2	Inputs and outputs of the program [Thomas-Simulation]	4
2.2	Inverse-power type model	5
2.2.1	Simulation method	7
2.2.2	Inputs and outputs of the program [IP-Simulation]	7
2.3	Generalized Thomas model of type A	8
2.3.1	Simulation method	10
2.3.2	Inputs and outputs of the program [TypeA-Simulation]	10
2.4	Generalized Thomas model of type B	11
2.4.1	Simulation method	13
2.4.2	Inputs and outputs of the program [TypeB-Simulation]	13
2.5	Generalized Thomas model of type C	14
2.5.1	Simulation method	14
2.5.2	Inputs and outputs of the program [TypeC-Simulation]	16
3	Parameter estimation	17
3.1	Overview of the estimation procedure	17
3.1.1	Palm intensity functions	19
3.1.2	Palm log-likelihood functions	20
3.2	Inputs and outputs of the program [Simplex-Thomas]	22

3.3	Non-parametric estimate of the Palm intensity	24
3.4	Inputs and outputs of the program [Palm-Thomas]	25
3.5	Inputs and outputs of the program [Simplex-IP]	26
3.6	Inputs and outputs of the program [Palm-IP]	29
3.7	Inputs and outputs of the program [Simplex-TypeA]	30
3.8	Inputs and outputs of the program [Palm-TypeA]	33
3.9	Inputs and outputs of the program [Simplex-TypeB]	35
3.10	Inputs and outputs of the program [Palm-TypeB]	37
3.11	Inputs and outputs of the program [Simplex-TypeC]	39
3.12	Inputs and outputs of the program [Palm-TypeC]	41

1 Overview of the models

We provide FORTRAN programs for the maximum Palm likelihood estimation of parameters of the Neyman-Scott type spatial cluster model (Tanaka et al. [7]). Neyman-Scott models are frequently used to apply them to spatial cluster point patterns. The centers of clusters of point pattern, called parent points, form a stationary Poisson point field with intensity μ . Then, the Neyman-Scott model is defined as the set of all offspring points described as follows (e.g., Stoyan and Stoyan [5], [6]):

1. Each parent point produces a random number M of offspring points which are realized independently and identically according to the probabilities $p_m = P\{M = m\}, m = 0, 1, \dots$
2. Locations of offspring points are distributed independently and identically according to a density function $q_\tau(x, y)$, where (x, y) is coordinates of offspring points relative to their parent point, and the parameter τ indicates a set of coefficients of the density function.

Hereafter, we consider parameters $\theta = (\mu, \nu, \tau)$, where $\nu = E[M] = \sum mp_m$, as the parameters that characterize the various models of the Neyman-Scott type. In particular, throughout this article, we assume that M has the Poisson distribution with mean ν .

Furthermore, we consider multi-type cluster models, which are formed as follows (Tanaka et al. [7]): the models allow two types of clusters with the different cluster sizes and distance scales. The intensities of centres are

μ_1 and μ_2 , respectively. The corresponding offspring cluster size distributes according to Poisson random variable with means ν_1 and ν_2 , respectively.

In order to estimate the parameters and select the competing models, we have prepared some programs written in FORTRAN77. We also plotted figures which are delineated in PDF format.

While such point patterns can be considered in spaces of arbitrary dimension, we here restrict ourself to planar point patterns in observed rectangular region $W = [0, T_x] \times [0, T_y]$, assuming the periodic boundary condition. Furthermore, we restrict ourself to the standardized case where $T_x = 1$ and $T_y \geq 1$ without loss of generality except for the scaling. All the models of point patterns are assumed to be stationary and isotropic. Therefore the density function $q_\tau(x, y)$ is denoted by polar coordinate $q_\tau(r)$ of distance r from its parent point.

For the simulations, we use the pseudo-random number generation routine referred to Wichmann and Hill [8].

2 Simulation of the models

In order to simulate spatial clustered point pattern of the Neyman-Scott models and their extensions, this package includes the FORTRAN programs [**Thomas-Simulate.f**], [**IP-Simulate.f**], [**TypeA-Simulate.f**], [**TypeB-Simulate.f**] and [**TypeC-Simulate.f**]. We consider the periodic boundary condition (i.e., a two-dimensional torus) for each of simulations of models on a rectangular with normalized scale as described below.

2.1 Thomas model

The *Thomas model* has the density function that is given by the bivariate Gaussian distribution with zero mean and the covariance matrix $\sigma^2 I$ where I is 2×2 identity matrix as follows (e.g., Tanaka et al. [7]). Due to isotropy this is described in polar coordinate as a function

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

of distance r from the corresponding parent's location.

2.1.1 Simulation method

The simulation of this model is produced as follows: let random variable U be independently and uniformly distributed in $[0, 1]$. We put

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

Then we have

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

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 , the offspring coordinates (x_j^i, y_j^i) , $j = 1, 2, \dots, J_i$ is given by

$$\begin{aligned} x_j^i &= x_i^p + r \cos(2\pi U), \\ y_j^i &= y_i^p + r \sin(2\pi U), \end{aligned}$$

owing to the isotropy condition of the distribution.

We adopt the following procedure of generation of the Poisson random number $M = Poisson(\nu)$ (e.g., Devroye [2]): Given a positive number ν and let a sequence of a random variable $\{U_k\}$ be independently and uniformly distributed in $[0, 1]$, M is the smallest integer such that

$$\sum_{k=1}^{M+1} -\log U_k > \nu, \quad (2.1)$$

where \log represents natural logarithm.

2.1.2 Inputs and outputs of the program [Thomas-Simulation]

In order to execute the FORTRAN program [**Thomas-Simulate.f**] for this simulation, we have to prepare the following control file that specifies the

necessary parameter values and others, as input variables. Then we have output files as specified below.

INPUTS:

The control file [**Thomas.param**] includes following variables in the four lines.

- **1st line.** The first three positive integer variables **ix**, **iy** and **iz** in this file are initial seeds for a sequence of uniform random numbers. We can get different independent point patterns by changing these seeds.
- **2nd line.** The variable T_y (≥ 1) is provided here for the standardized coordinates of points in the rectangular region $[0, 1] \times [0, T_y]$ without loss of generality except for the scaling.
- **3rd line.** The parameter values μ , ν and σ are provided in the order from the left to the right.
- **4th line** is ignored in this case.

OUTPUTS:

- Calculated records of the simulation program are printed on line. For convenience of users to check, we left the print in the file [**Thomas-parents-offspring.print**], which shows numbers of offspring for each parent.
- The files named by [**Thomas-parents.xy**] and [**Thomas-offspring.xy**] provide the coordinates of simulated parent points and their offspring points, respectively, in the rectangular region $[0, 1] \times [0, T_y]$.

Here we simulate parent points with $\mu = 50.0$ (cf., **Figure 1**) and of the Thomas offspring with $(\mu, \nu, \sigma) = (50.0, 30.0, 0.03)$ (cf., **Figure 2**).

2.2 Inverse-power type model

The *Inverse-power type* density function of the distance is given as follows (Tanaka et al. [7]):

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

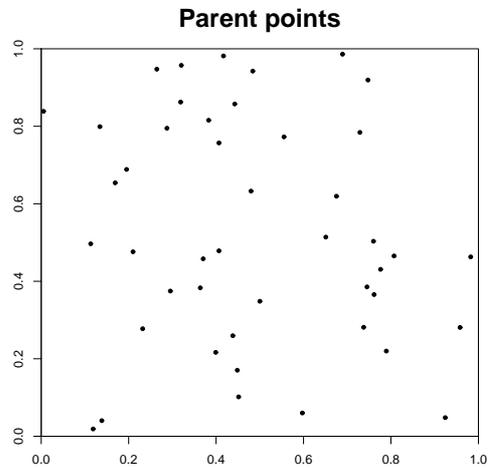


Figure 1: Simulated parent points of the Thomas model with $\mu = 50.0$.

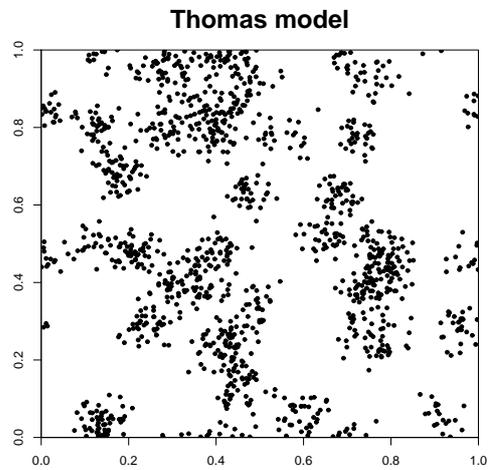


Figure 2: Simulated offspring points of the Thomas model with $(\mu, \nu, \sigma) = (50.0, 30.0, 0.03)$.

where the parameters p and c represent a decay order and scaling factor with respect to the distance, respectively.

2.2.1 Simulation method

The simulation of the inverse-power type model is carried out as follows: Let random variable U be independently and uniformly distributed in $[0, 1]$. For any $r \geq 0$,

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

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

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

Similarly, coordinate of the offspring points (x_j^i, y_j^i) , $j = 1, 2, \dots, Poisson(\nu)$ with the inverse-power type is given for each $i = 1, 2, \dots, Poisson(\mu)$,

$$\begin{aligned} x_j^i &= x_i^p + r \cos(2\pi U), \\ y_j^i &= y_i^p + r \sin(2\pi U), \end{aligned}$$

using series of different uniform random numbers $\{U\}$ for different i and j .

2.2.2 Inputs and outputs of the program [IP-Simulation]

In order to execute the program [IP-Simulate.f] for this simulation, we have to prepare the following control file that specifies the necessary parameter values and others, as inputs. Then we have outputs of files as specified below.

INPUTS:

The control file [IP.param] includes following variables in the five lines.

- **1st line.** The first three positive integer variables **ix**, **iy** and **iz** in this file are initial seeds for a sequence of uniform random numbers. We can get different independent point patterns by changing these seeds.

- **2nd line.** The variable T_y (≥ 1) is provided here for the standardized coordinates of points in the rectangular region $[0, 1] \times [0, T_y]$ without loss of generality except for the scaling.
- **3rd line.** The parameter values μ , ν , p and c are provided in the order from the left to the right.
- **4th line** is ignored in this case.
- **5th line** is ignored in this case.

OUTPUTS:

- Calculated records of the simulation program are printed on line. For convenience of users to check, we left the print in the file [**IP-parents-offspring.print**], which shows numbers of offspring for each parent.
- The files named by [**IP-parents.xy**] and [**IP-offspring.xy**] provide the coordinates of simulated parent points and their offspring points, respectively, in the rectangular region $[0, 1] \times [0, T_y]$.

Here we simulate parent points with $\mu = 50.0$ (cf., **Figure 3**) and of the inverse-power type offspring with $(\mu, \nu, p, c) = (50.0, 30.0, 1.5, 0.005)$ (cf., **Figure 4**).

2.3 Generalized Thomas model of type A

The *generalized Thomas model of type A*, or the *Type A model* for short, is given by the mixed Gaussian density function with σ_1 and σ_2 as follows (Tanaka et al. [7]):

$$q_{\tau}(r) = \left\{ \frac{ar}{\sigma_1^2} \exp\left(-\frac{r^2}{2\sigma_1^2}\right) + \frac{(1-a)r}{\sigma_2^2} \exp\left(-\frac{r^2}{2\sigma_2^2}\right) \right\}, \tau = (a, \sigma_1, \sigma_2),$$

where the parameter a indicates the mixture ratio of the density function.

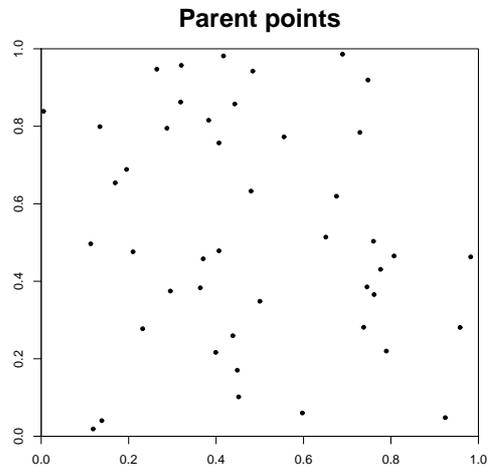


Figure 3: Simulated parent points of the inverse-power type with $\mu = 50.0$.

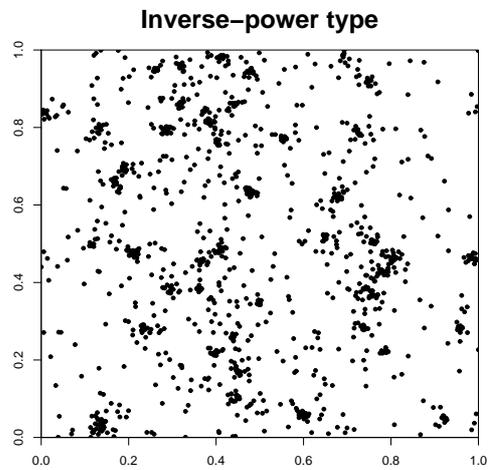


Figure 4: Simulated offspring points of the inverse-power type with $(\mu, \nu, p, c) = (50.0, 30.0, 1.5, 0.005)$.

2.3.1 Simulation method

Parents' configuration and numbers of the offspring cluster sizes are generated by the same way as the Thomas model (cf., (2.1)).

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

$$r = \begin{cases} \sigma_1 \sqrt{-2 \log(1 - U_1)}, & U_2 \leq a, \\ \sigma_2 \sqrt{-2 \log(1 - U_1)}, & \text{otherwise.} \end{cases}$$

Then, by the isotropy condition, for $i = 1, 2, \dots, Poisson(\mu)$, coordinate of the offspring points (x_j^i, y_j^i) , $j = 1, 2, \dots, Poisson(\nu)$ is given for each $i = 1, \dots, Poisson(\mu)$,

$$\begin{aligned} x_j^i &= x_i^p + r \cos(2\pi U), \\ y_j^i &= y_i^p + r \sin(2\pi U), \end{aligned}$$

using series of different uniform random numbers $\{U_1, U_2, U\}$ for different i and j .

2.3.2 Inputs and outputs of the program [TypeA-Simulation]

In order to execute the program [TypeA-Simulate.f] for this simulation, we have to prepare the following control file that specifies the necessary parameter values and others, as inputs. Then we have outputs of files as specified below.

INPUTS:

The control file [TypeA.param] includes following variables in the five lines.

- **1st line.** The first three positive integer variables **ix**, **iy** and **iz** in this file are initial seeds for a sequence of uniform random numbers. We can get different independent point patterns by changing these seeds.
- **2nd line.** The variable T_y (≥ 1) is provided here for the standardized coordinates of points in the rectangular region $[0, 1] \times [0, T_y]$ without loss of generality except for the scaling.

- **3rd line.** The parameter values μ , ν , a , σ_1 and σ_2 are provided in the order from the left to the right.
- **4th line** is ignored in this case.
- **5th line** is ignored in this case.

OUTPUTS:

- Calculated records of the simulation program are printed on line. For convenience of users to check, we left the print in the file [**TypeA-parents-offspring.print**], which shows numbers of offspring for each parent.
- The files named by [**TypeA-parents.xy**] and [**TypeA-offspring.xy**] provide the coordinates of simulated parent points and their offspring points, respectively, in the rectangular region $[0, 1] \times [0, T_y]$.

Here we simulate parent points with $\mu = 50.0$ (cf., **Figure 5**) and of the mixed Thomas offspring with $(\mu, \nu, a, \sigma_1, \sigma_2) = (50.0, 30.0, 0.3, 0.005, 0.1)$ (cf., **Figure 6**).

2.4 Generalized Thomas model of type B

Furthermore, we provide the extension of the Neyman-Scott model allowing two types of clusters with the different cluster sizes and the distance scales.

In particular, here, we assume two types of clusters with the different sizes and distributions of the Thomas type. The intensities of the respective parents (centres) are μ_1 and μ_2 , respectively. The respective clusters have the Poisson means of the offspring points ν_1 and ν_2 . The density functions of the offspring relative to respective parent are given by

$$q_{\sigma_1}(r) = \frac{r}{\sigma_1^2} \exp\left(-\frac{r^2}{2\sigma_1^2}\right), \quad q_{\sigma_2}(r) = \frac{r}{\sigma_2^2} \exp\left(-\frac{r^2}{2\sigma_2^2}\right), \quad (2.2)$$

where σ_1 and σ_2 indicate their scales. If the mean sizes of the two types of clusters are the same, then it is hereafter referred to as the *generalized Thomas model of type B*, or the *Type B model* for short (Tanaka et al. [7]).

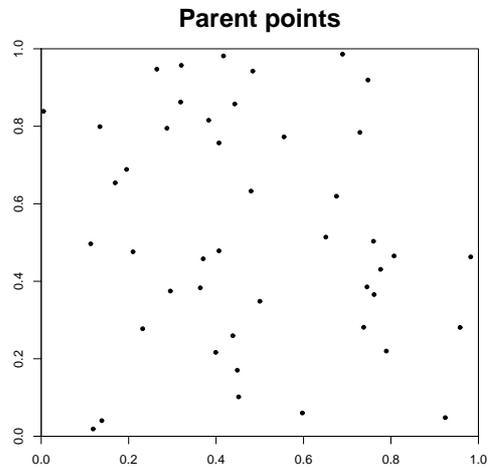


Figure 5: Simulated parent points of the Type A model with $\mu = 50.0$.

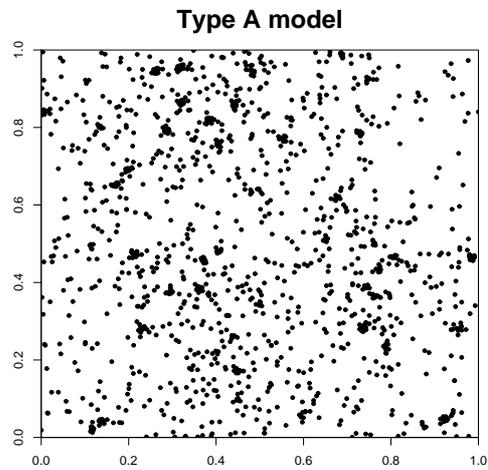


Figure 6: Simulated offspring points of the Type A model with $(\mu, \nu, a, \sigma_1, \sigma_2) = (50.0, 30.0, 0.3, 0.005, 0.1)$.

2.4.1 Simulation method

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, x_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 , 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

$$\begin{aligned}x_j^{k,i} &= x_i^k + r_k \cos(2\pi U), \\y_j^{k,i} &= y_i^k + r_k \sin(2\pi U),\end{aligned}$$

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 .

2.4.2 Inputs and outputs of the program [TypeB-Simulation]

In order to execute the program [TypeB-Simulate.f] for this simulation, we have to prepare the following control file that specifies the necessary parameter values and others, as inputs. Then we have outputs of files as specified below.

INPUTS:

The control file [TypeB.param] includes following variables in the four lines.

- **1st line.** The first three positive integer variables **ix**, **iy** and **iz** in this file are initial seeds for a sequence of uniform random numbers. We can get different independent point patterns by changing these seeds.
- **2nd line.** The variable T_y (≥ 1) is provided here for the standardized coordinates of points in the rectangular region $[0, 1] \times [0, T_y]$ without loss of generality except for the scaling.
- **3rd line.** The parameter values $\mu_1, \mu_2, \nu, \sigma_1$ and σ_2 are provided in the order from the left to the right.

- **4th line** is ignored in this case.

OUTPUTS:

- Calculated records of the simulation program are printed on line. For convenience of users to check, we left the print in the file [**TypeB-parents-offspring.print**], which shows numbers of offspring for each parent.
- The files named by [**TypeB-parents.xy**] and [**TypeB-offspring.xy**] provide the coordinates of simulated parent points and their offspring points, respectively, in the rectangular region $[0, 1] \times [0, T_y]$.

Here we simulate parent points with $\mu_1 = 10.0$ and $\mu_2 = 40.0$ (cf., **Figure 7**) and of the two types of the Thomas offspring with $(\mu_1, \mu_2, \nu, \sigma_1, \sigma_2) = (10.0, 40.0, 30.0, 0.01, 0, 0.3)$ (cf., **Figure 8**).

It is possible to change this simulation program for the general case where there is no restrictions among the parameters $\mu_1, \mu_2, \nu_1, \nu_2, \sigma_1$ and σ_2 , but the estimation by means of the Palm log-likelihood does not identify the parameters (cf., Tanaka et al. [7]).

2.5 Generalized Thomas model of type C

If $\nu_1 \neq \nu_2$ and each density function q_{σ_i} , $i = 1, 2$ have the same forms (2.2), then, in particular, for σ_1 and σ_2 we may constrain ν_1 and ν_2 to be

$$\nu_2 = \nu_1 \left(\frac{\sigma_2}{\sigma_1} \right),$$

and it is hereafter referred to as the *generalized Thomas model of type C* or the *Type C model*, for short (Tanaka et al. [7]).

2.5.1 Simulation method

The simulation method of the Type C model is similar to that of the Type B model because the Type C model consists of two types of the Thomas model with the different cluster sizes and the distance scales. Consider the two types of the Thomas model with parameters (μ_1, ν_1, σ_1) and (μ_2, ν_2, σ_2) .

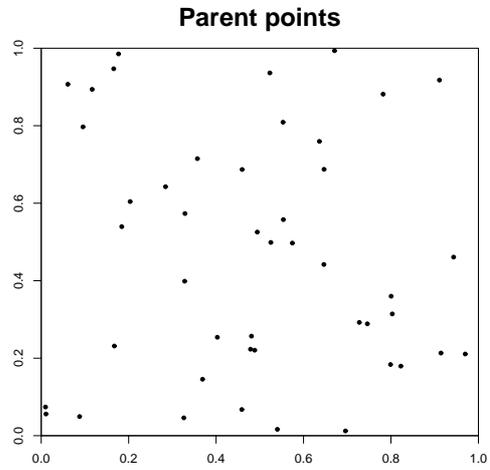


Figure 7: Simulated two types of parent points of the Type B model with $\mu_1 = 10.0$ and $\mu_2 = 40.0$.

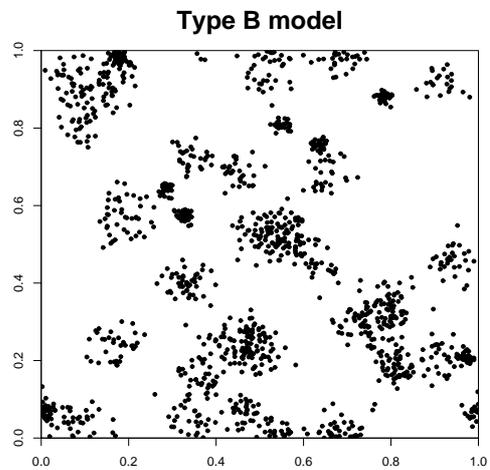


Figure 8: Simulated two types of offspring points of the Type B model with $(\mu_1, \mu_2, \nu, \sigma_1, \sigma_2) = (10.0, 40.0, 30.0, 0.01, 0.03)$.

Parents' configuration and numbers of the offspring cluster sizes are generated by the two types of uniformly distributed parents (x_i^k, x_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 , 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

$$\begin{aligned}x_j^{k,i} &= x_i^k + r_k \cos(2\pi U), \\y_j^{k,i} &= y_i^k + r_k \sin(2\pi U),\end{aligned}$$

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 .

2.5.2 Inputs and outputs of the program [TypeC-Simulation]

In order to execute the program [TypeC-Simulate.f] for this simulation, we have to prepare the following control file that specifies the necessary parameter values and others, as inputs. Then we have outputs of files as specified below.

INPUTS:

The control file [TypeC.param] includes following variables in the four lines.

- **1st line.** The first three positive integer variables **ix**, **iy** and **iz** in this file are initial seeds for a sequence of uniform random numbers. We can get different independent point patterns by changing these seeds.
- **2nd line.** The variable T_y (≥ 1) is provided here for the standardized coordinates of points in the rectangular region $[0, 1] \times [0, T_y]$ without loss of generality except for the scaling.
- **3rd line.** The parameter values $\mu_1, \mu_2, \nu_1, \nu_2, \sigma_1$ and σ_2 are provided in the order from the left to the right.
- **4th line** is ignored in this case.

OUTPUTS:

- Calculated records of the simulation program are printed on line. For convenience of users to check, we left the print in the file [**TypeC-parents-offspring.print**], which shows numbers of offspring for each parent.
- The files named by [**TypeC-parents.xy**] and [**TypeC-offspring.xy**] provide the coordinates of simulated parent points and their offspring points, respectively, in the rectangular region $[0, 1] \times [0, T_y]$.

Here we simulate parent points with $\mu_1 = 5.0$ and $\mu_2 = 9.0$ (cf., **Figure 9**) and then respective offsprings with $(\mu_1, \mu_2, \nu_1, \nu_2, \sigma_1, \sigma_2) = (5.0, 9.0, 30.0, 150.0, 0.01, 0.05)$ (cf., **Figure 10**).

It is possible to change this simulation program for the general case where there is no restrictions among the parameters $\mu_1, \mu_2, \nu_1, \nu_2, \sigma_1$ and σ_2 , but the estimation by means of the Palm log-likelihood does not identify the parameters (Tanaka et al. [7]).

3 Parameter estimation

3.1 Overview of the estimation procedure

It is very desirable to estimate model parameters as accurate as possible. However, it is difficult to obtain maximum likelihood estimates due to the structure and property of Neyman-Scott cluster models (e.g., Tanaka et al. [7]). In order to obtain the better estimates than the contract estimates using the difference between K -functions of the model and of the empirical, we shall consider a likelihood function based on the Palm intensity function (e.g., Tanaka et al. [7]) below.

Here the parameter estimation due to the Palm likelihood function in (3.6) and (3.7) is provided for various models. For the maximization procedure (equivalent to minimization without loss of generality), we adopt the *Simplex estimation method* (e.g., Kowalik and Osborne [3]) to maximize the Palm likelihood function (or minimize the negative Palm likelihood function).

The simplex means a usual geometric figure, to simplify, which is a set of $n + 1$ points in the n -dimensional Euclidean space. The principal idea of this method is that we can easily form a new simplex from the current one

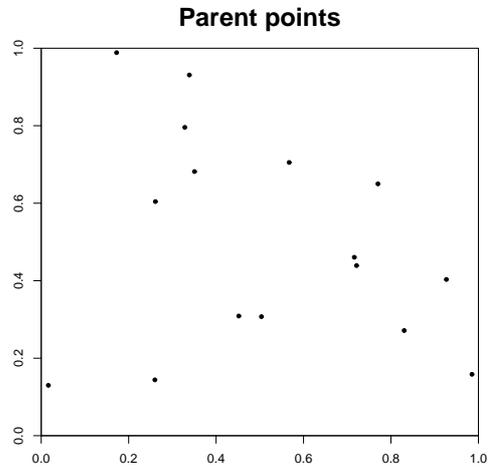


Figure 9: Simulated two types of parent points of the Type C model with $\mu_1 = 5.0$ and $\mu_2 = 9.0$.

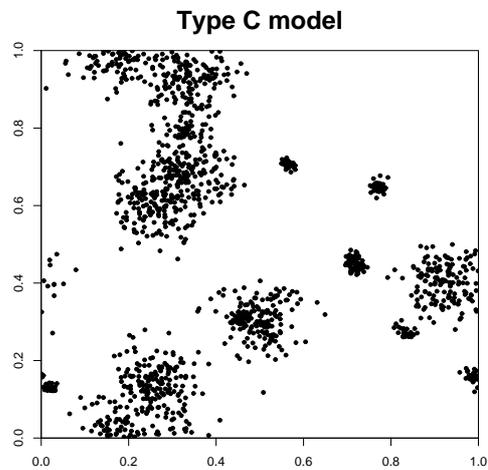


Figure 10: Simulated two types of offspring points of the Type C model with $(\mu_1, \mu_2, \nu_1, \nu_2, \sigma_1, \sigma_2) = (5.0, 9.0, 30.0, 150.0, 0.01, 0.05)$.

by reflecting one point in the hyperplane spanned by the remaining points. If we choose for this purpose the vertex of the simplex at which the function is greatest, we can expect that at the reflected vertex the function value will be lower. If this is the case then we can continue the process and move our simplex closer to the minimum.

In order to calculate the function, we need to give the derived formula of the Palm intensity function λ_o due to the Ripley's K -function (Tanaka et al. [7]) as follows:

$$\lambda_o(r) = \frac{\lambda}{2\pi r} \frac{dK(r)}{dr}, r > 0, \quad (3.1)$$

where $\lambda = \mu\nu$.

For any Neyman-Scott cluster models, the K -function satisfies

$$\lambda K(r) = \lambda\pi r^2 + \frac{m_{[2]}}{\nu} F_\tau(r), \quad r \geq 0, \quad (3.2)$$

where $m_{[2]} = E[M(M-1)]$ and $F_\tau(r)$ are the second-order factorial moment and the distribution function of distance between the offspring points in the same cluster, respectively (cf., Stoyan and Stoyan [5], [6]). If $M = \text{Poisson}(\nu)$, then $m_{[2]} = \nu^2$.

The distribution $F_\tau(r)$ of the distance between the offspring points of the same cluster is given by

$$\begin{aligned} F_\tau(r) = & 2 \left[\int_0^{r/2} \left(\int_{r-r_1}^{r+r_1} \frac{1}{\pi} \arccos \left(\frac{r_1^2 + r_2^2 - r^2}{2r_1r_2} \right) q_\tau(r_2) dr_2 \right) q_\tau(r_1) dr_1 \right. \\ & + \int_{r/2}^\infty \left(\int_{r_1}^{r_1+r} \frac{1}{\pi} \arccos \left(\frac{r_1^2 + r_2^2 - r^2}{2r_1r_2} \right) q_\tau(r_2) dr_2 \right) q_\tau(r_1) dr_1 \\ & \left. + \int_0^{r/2} \left(\int_{r_1}^{r-r_1} q_\tau(r_2) dr_2 \right) q_\tau(r_1) dr_1 \right], \quad (3.3) \end{aligned}$$

for any $r \geq 0$ and the parameter τ (e.g., Tanaka et al. [7]).

3.1.1 Palm intensity functions

Although analytical form of the Palm intensity function is not available for the inverse-power type model and the Type A model due to the complexity of the $f_\tau(r)$ that is the derivative of the $F_\tau(r)$, it is available for the Thomas model, Type B model and Type C model. According to the Tanaka et al. [7], the Palm intensity functions of the following models are calculated as follows:

Example 3.1 (Thomas model). (e.g., Daley and Vere-Jones [1]) For any $r \geq 0$,

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

Example 3.2 (Type B model). For any $r \geq 0$,

$$\lambda_o(r) = \lambda + \frac{\nu}{4\pi} \left\{ \frac{a}{\sigma_1^2} \exp\left(-\frac{r^2}{4\sigma_1^2}\right) + \frac{(1-a)}{\sigma_2^2} \exp\left(-\frac{r^2}{4\sigma_2^2}\right) \right\}, \quad (3.4)$$

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.

Example 3.3 (Type C model). For any $r \geq 0$,

$$\lambda_o(r) = \lambda + \frac{1}{4\pi} \left\{ \frac{a\nu_1}{\sigma_1^2} \exp\left(-\frac{r^2}{4\sigma_1^2}\right) + \frac{(1-a)\nu_2}{\sigma_2^2} \exp\left(-\frac{r^2}{4\sigma_2^2}\right) \right\}, \quad (3.5)$$

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.

3.1.2 Palm log-likelihood functions

Procedures of parameter estimation due to the Palm intensity functions depend on models.

Thus, the Palm log-likelihood function with the analytical Palm intensity function is given as follows (e.g., Tanaka et al. [7]):

$$\log L(\mu, \nu, \tau) = \sum_{\{i,j;i \neq j, r_{ij} < R\}} \log(N(W)\lambda_o(r_{ij})) - N(W) \int_0^R 2\pi\lambda_o(r)r dr, \quad (3.6)$$

where $N(W)$ is the number of points in W , and the sum is taken for any pair of points x_i and x_j , $i \neq j$ such that the distance r_{ij} between x_i and x_j is smaller than R , where R is sufficiently greater than the minimum value of r such that $\lambda_o(r)$ is equal to the intensity $\lambda = \mu\nu$ but assume $R \leq 1/2$ under the periodic boundary condition.

Here we note that R is shown from a non-parametric Palm intensity function as below according to the Tanaka et al. [7], where R is taken as $1/2$

at the largest. Also, we note that the constant $\log N(W)$ of the first term can be neglected in maximization of (3.6).

The followings analytic form of the log Palm-likelihood are provided for the Thomas, Type B and Type C models (cf., Tanaka et al. [7]).

Example 3.4 (Thomas model). The Palm log-likelihood function of the Thomas model is analytically calculated because the Thomas model has the analytical form of the Palm intensity function. By (3.1), (3.2) and (3.6) or (3.7), we have

$$\begin{aligned} \log L(\mu, \nu, \sigma) = & \sum_{\{i,j;i \neq j, r_{ij} < R\}} \log \nu \left\{ \mu + \frac{1}{4\pi\sigma^2} \exp\left(-\frac{r_{ij}^2}{4\sigma^2}\right) \right\} \\ & - N(W)\nu \left\{ \pi\mu R^2 + 1 - \exp\left(-\frac{R^2}{4\sigma^2}\right) \right\}, \end{aligned}$$

with $R = 1/2$ which means the half of the t_x (**TX**) in side length of the normalized rectangular.

Example 3.5 (Type B model). The Palm log-likelihood function of the Type B model is analytically calculated because the Type B model has the analytical form of the Palm intensity function. By (3.4) and (3.6), we have

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

with $R = 1/2$, where $\alpha = a\nu$ and $\beta = (1 - a)\nu$.

Example 3.6 (Type C model). The Palm log-likelihood function of the Type C model is analytically calculated because the Type C model has the

analytical form of the Palm intensity function. By (3.5) and (3.6), we have

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

with $R = 1/2$, where $\alpha = a\nu_1$ and $\beta = (1 - a)\nu_2$.

On the other hand, general model including the inverse-power type and Type A model such that Palm intensity function can not be calculated analytically due to the complexity of the differential of $F_\tau(r)$ require the numerical calculation of the Palm likelihood function. For such models, we need to take the alternative form without explicit representation of the Palm intensity function (Tanaka et al. [7]) as follows: By (3.1), the representation of the function using the K -function is written by

$$\log L(\mu, \nu, \tau) = \sum_{\{i,j;i \neq j, r_{ij} < R\}} \log\{N(W)\lambda_o(r_{ij})\} - N(W)\mu\nu K(R), \quad (3.7)$$

where we calculate (3.1), (3.2) and (3.3) numerically.

To summarize, the programs [**Simplex-Thomas.f**], [**Simplex-TypeB.f**] and [**Simplex-TypeC.f**] of the Thomas related models use analytical forms of the functions, while the programs [**Simplex-IP.f**] and [**Simplex-TypeA.f**] of other type models have to use numerical integration and difference to compute the functions (see below), which need very long c.p.u. time in the minimization procedure.

3.2 Inputs and outputs of the program [**Simplex-Thomas**]

In order to execute the program [**Simplex-Thomas.f**] for a dataset, we have to prepare the following control file that specifies the necessary parameter values, name of data and other variables, as inputs. Then we have outputs of files as specified below.

INPUTS:

The following variables are given in the four lines of the control file [**Thomas.param**] that is also used for the simulation program of the Thomas model.

- **1st line.** All the variables in this line are ignored in the present case.
- **2nd line.** The variable T_y (≥ 1) is provided here for the standardized coordinates of points in the rectangular region $[0, 1] \times [0, T_y]$ without loss of generality except for the scaling. The coordinates of points in the rectangular region are standardized. Note here that the coordinates of given data are considered the periodic boundary condition.
- **3rd line.** The initial guesses of the parameters μ , ν and σ are provided in order to get the MPLE values.
- **4th line** describes the name of dataset that are standardized by the rectangular length of the smaller side that should be set as the x-coordinate.

Here we set the true values $(\mu_0, \nu_0, \sigma_0) = (50.0, 30.0, 0.03)$ to apply the simulated data [**Thomas-offspring.xy**] (see **Figure2**).

OUTPUTS:

The output file [**Thomas.simplex.print**] provides the process of minimizing the negative Palm log-likelihood function until the values converge to the MPLE values for the given data. The outputs are arrayed as follows:

- In the **1st column**, characters **update** or **testfn** is written, where the **update** indicates that $-\log L$ value in the **2nd column** has attained the minimum so far, otherwise not.
- The **3rd column** and the rests show the values of the corresponding parameters as indicated in the **1st row**.
- From the **2nd row** to the **last row** show the optimization procedure starting from the initial values in the **2nd row**. The **last row** shows the minimized $-\log L$ and the corresponding MPLE values.

The other output file [**Thomas.simplex.x.print**] shows the same optimization procedure by the simplex method as indicated in above output file

but with the normalized parameters depending on the initial scaling guess of the parameters. Namely, the actual estimates are obtained by the indicated \mathbf{x} -values times the corresponding initial values.

- The **1st column** shows the smallest $-\log L$ value at the stage so far.
- The **2nd column** shows the standard deviations among function values of the vertices $\mathbf{x}(\mathbf{i})$ of the simplex.
- The **3rd column** and the rests show the values of the corresponding normalized parameters $\mathbf{x}(\mathbf{i})$ as indicated in the **1st row**.
- From the **2nd row** to the **last row** show the optimization procedure starting from the initial values in the **2nd row**, all of which should be 1.0. The **last row** shows the minimized $-\log L$ and the normalized variables corresponding to the MPLE values relative to the initial estimates. The **2nd column** is also expected to be small enough.

3.3 Non-parametric estimate of the Palm intensity

The programs [**Palm-Thomas.f**], [**Palm-IP.f**], [**Palm-TypeA.f**], [**Palm-TypeB.f**] and [**Palm-TypeC.f**] below include the subroutine that calculates the non-parametric Palm intensity function estimated directly from a set of point pattern data.

The non-parametric Palm intensity function is realized as follows: let ψ_1, \dots, ψ_n be the Cartesian vector coordinates of given points. Then the differences of the vector coordinates

$$\Psi_{i,j} \equiv \psi_j - \psi_i, \quad i = 1, \dots, n, \quad j = 1, \dots, n, \quad i \neq j,$$

is regarded as a realized configuration under the Palm intensity function, the procedure of which is the so-called *difference process*.

As a non-parametric estimation of the Palm intensity function $\lambda_0(r)$ in the polar coordinate, we count the number $N(r, r + \Delta)$ of such vectors in each annular set whose radii ends are r and $r + \Delta$. Namely, roughly speaking, the Palm intensity function notices not configurations but differences of the vector coordinates of the dataset.

Then the ratio of this number to the area of the annulus, that is, $N(r, r + \Delta)/N(W)/(2\pi r\Delta)$, provides an estimate of the Palm intensity function of r in the polar coordinates.

3.4 Inputs and outputs of the program [Palm-Thomas]

INPUTS:

The control file [Thomasparam.palm] specifies the followings in order to delineate non-parametric and parametric Palm intensity functions that are corresponding to the specified values.

- **1st line** describes the dataset name of point pattern to delineate the non-parametric Palm intensity function.
- **2nd line.** A value of Δ shows a width for the non-parametric Palm intensity function. The value depends on the number of given point pattern data. Here we set $\Delta = 0.001$. The variable $T_y (\geq 1)$ is provided here for the standardized coordinates of points in the rectangular region $[0, 1] \times [0, T_y]$ without loss of generality except for the scaling.
- From the **3rd line** to the **bottom line** (not more than 10th lines) describe respective value of the given parameters (μ_i, ν_i, σ_i) , $i = 1, 2, \dots, I \leq 8$, for respective estimates of the Palm intensity functions of the Thomas model. If there are nothing in these lines (3rd-10th lines), the following outputs provide only the non-parametric Palm intensity function, namely, only the 1st and 2nd columns.

OUTPUTS:

Then we have the outputs-file [Palm-Thomas.txt] in the following format.

- The **1st column** provides 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.
- The **2nd column** provides the corresponding values of the non-parametric Palm intensity function of r , which is normalized by the total intensity estimate (the mean number of points in a unit area) of the point pattern data.
- The **3rd-last columns** show the normalized Palm intensity functions $\lambda_o(r)/\hat{\lambda}$ calculated from the given sets of parameter values (μ_i, ν_i, σ_i) .

Here, as an example, we provide the non-parametric Palm intensity function with $\Delta = 0.001$ and two Palm intensity functions corresponding to the true and MPLE parameters, respectively (**Figure 11**).

3.5 Inputs and outputs of the program [Simplex-IP]

This program computes MPLEs via numerical calculation of the Ripley's K -function for the Palm-likelihood function (3.7). For the detail, Tanaka et al. [7].

INPUTS:

The following variables are given in the five lines of the control file [**IP.param**], which is also used for the simulation program of the inverse-power type.

- **1st line.** The first three variables in this line are ignored in the present case. The last column variable indicates **iskip** for the fast likelihood but rough approximation of the initial estimates, where **iskip = 1** is set in the final procedure. Because the calculation takes a long time due to large number of repetitions of the numerical integration (see Tanaka et al. [7]), a suitable initial estimate is effective in the minimizing procedure. For example, **iskip = 1000** is set to obtain a good initial estimate. The **iskip** calculate the Palm intensity function in the log-likelihood function for every **iskip**-th $r_{i,j}$ in the ordered distances of the pairs i and j , and the smaller **iskip** value provides the more accurate estimate but takes longer time for the computation.
- **2nd line.** The variable T_y (≥ 1) is provided here for the standardized coordinates of points in the rectangular region $[0, 1] \times [0, T_y]$ without loss of generality except for the scaling. The coordinates of points in the rectangular region are standardized. Note here that the coordinates of given data are considered the periodic boundary condition.
- **3rd line.** The initial guesses of the parameters μ , ν , p and c are provided in order to get the MPLEs.
- **4th line** describes the name of dataset that are standardized by the rectangular length of the smaller side that should be set as the x-coordinate.

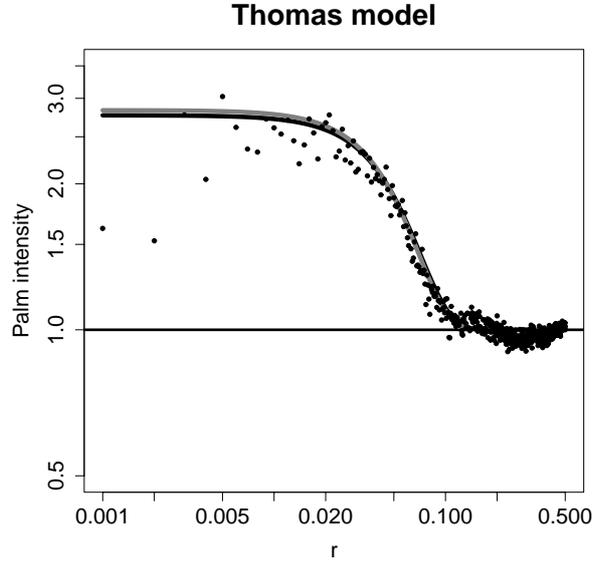


Figure 11: Palm intensity functions of the Thomas model due to the parameters in **Table 1** in doubly logarithmic scales, dots are the non-parametric estimate, the black and gray curves are of the true and MPLE parameters, respectively, and the horizontal line shows the normalized intensity of the simulated point pattern. All the plots are normalized with respect to the mean intensity (the mean number of events per unit area).

Table 1: MPLE values of the Thomas model from the simulated data.

Parameter	μ	ν	σ
True value	50.0	30.0	0.03
MPLE	51.3	26.1	0.0291

- **5th line** indicates upper limit value **X2** in place of ∞ in the integral in (3.3), which is provided in such a way that $\mathbf{X2} > R/2$ where R is from (3.6). Here we set $\mathbf{X2} = 0.3 (> 1/4)$.

Here we set the true values $(\mu_0, \nu_0, p_0, c_0) = (50.0, 30.0, 1.5, 0.005)$ to apply the simulated data [**IP-offspring.xy**] (see **Figure4**).

OUTPUTS:

The output file [**IP.simplex.print**] provides the process of minimizing the negative Palm log-likelihood function until the values converge to the MPLEs for the given data. The outputs are arrayed as follows:

- In the **1st column**, characters **update** or **testfn** is written, where the **update** indicates that $-\log L$ value in the **2nd column** has attained the minimum so far, otherwise not.
- The **3rd column** and the rests show the values of the corresponding parameters as indicated in the **1st row**.
- From the **2nd row** to the **last row** show the optimization procedure starting from the initial values in the **2nd row**. The **last row** shows the minimized $-\log L$ and the corresponding MPLEs.

The other output file [**IP.simplex.x.print**] shows the same optimization procedure by the simplex method as indicated in above output file but with the normalized parameters depending on the initial scaling guess of the parameters. Namely, the actual estimates are obtained by the indicated **x**-values times the corresponding initial values.

- The **1st column** shows the smallest $-\log L$ value at the stage so far.
- The **2nd column** shows the standard deviations among function values of the vertices **x(i)** of the simplex.
- The **3rd column** and the rests show the values of the corresponding normalized parameters **x(i)** as indicated in the **1st row**.
- From the **2nd row** to the **last row** show the optimization procedure starting from the initial values in the **2nd row**, all of which should be 1.0. The **last row** shows the minimized $-\log L$ and the normalized variables corresponding to the MPLEs relative to the initial estimates. The **2nd column** is also expected to be small enough.

In the directory, in addition, the output files [**IP1000.simplex.print**] and [**IP1000.simplex.x.print**] for the case of **iskip = 1000** are also provided for the sake of users to check within a shorter c.p.u. time.

3.6 Inputs and outputs of the program [**Palm-IP**]

The program [**Palm-IP.f**] calculates the Palm intensity functions due to the numerical differential calculation of the $F_\tau(r)$ (cf., Tanaka et al. [7]) for given parameter sets of the inverse-power type associated with the non-parametric Palm intensity function estimated directly from any point pattern data.

INPUTS:

The following variables are given in the control file [**IPparam.palm**].

- **1st line** describes the dataset name of point pattern to delineate the non-parametric Palm intensity function.
- **2nd line.** A common value of Δ shows a width for the non-parametric Palm intensity function and the increment of numerical differential calculation of the $F_\tau(r)$. The value depends on the number of given point pattern data. Here we set $\Delta = 0.001$. The variable T_y (≥ 1) is provided here for the standardized coordinates of points in the rectangular region $[0, 1] \times [0, T_y]$ without loss of generality except for the scaling.
- **3rd line** indicates upper limit value **X2** in place of ∞ in the integral in (3.3), which is provided in such a way that **X2** $> R/2$ where R is from (3.6). Here we set **X2** = 0.3 ($> 1/4$).
- From the **4th line** to the **bottom line** (not more than 10th lines) describe respective value of the given parameters (μ_i, ν_i, p_i, c_i) , $i = 1, 2, \dots, I \leq 7$, for respective estimates of the Palm intensity functions of the inverse-power type. If there are nothing in these lines (4th-10th lines), the following outputs provide only the non-parametric Palm intensity function, namely, only the 1st and 2nd columns.

OUTPUTS:

Then we have the outputs-file [**Palm-IP.txt**] in the following format.

- The **1st column** provides 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.
- The **2nd column** provides the corresponding values of the non-parametric Palm intensity function of r , which is normalized by the total intensity estimate (the mean number of points in a unit area) of the point pattern data.
- The **3rd-last columns** show the normalized Palm intensity functions $\lambda_o(r)/\hat{\lambda}$ calculated from the given sets of parameter values (μ_i, ν_i, p_i, c_i) .

Here, as an example, we provide the non-parametric Palm intensity function with $\Delta = 0.001$ and two Palm intensity functions corresponding to the true and MPLE parameters, respectively (**Figure 12**).

3.7 Inputs and outputs of the program [Simplex-TypeA]

The same as the inverse-power type, this program also computes MPLEs via the Ripley's K -function, numerically.

INPUTS:

The following variables are given in the five lines of the control file [**TypeA.param**], which is also used for the simulation program of the Type A model.

- **1st line.** The first three variables in this line are ignored in the present case. The last column variable indicates **iskip** for the fast likelihood but rough approximation of the initial estimates, where **iskip = 1** is set in the final procedure. Because the calculation takes a long time due to large number of repetitions of the numerical integration (see Tanaka et al., [7]), a suitable initial estimate is effective in the minimizing procedure. Therefore **iskip = 1000**, for instance, is set to obtain a good initial estimate. The **iskip** calculate the Palm intensity function in the log-likelihood function for every **iskip**-th $r_{i,j}$ in the ordered distances of the pairs i and j , and the smaller **iskip** value provides the more accurate estimate but takes longer time for the computation.

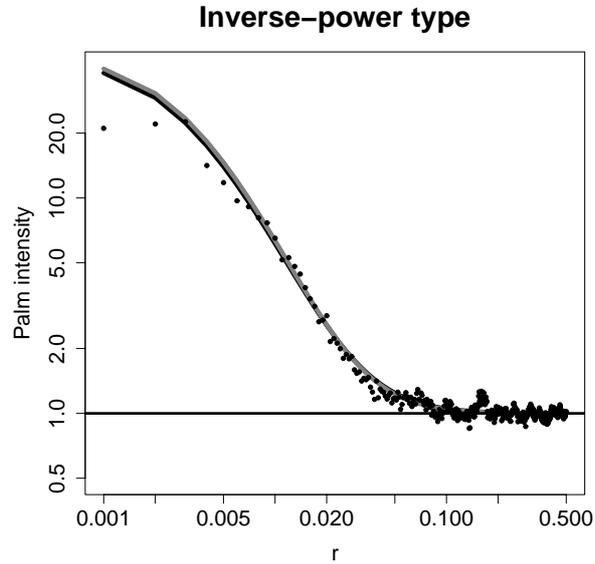


Figure 12: Palm intensity functions of the inverse-power type due to the parameters in **Table 2** in doubly logarithmic scales, dots are the non-parametric estimate. The curves of the true and MPLE parameters are overlapped to each other, and the horizontal line shows the normalized intensity of the simulated point pattern. All the plots are normalized with respect to the mean intensity (the mean number of events per unit area).

Table 2: MPLE values of the inverse-power type from the simulated data.

Parameter	μ	ν	p	c
True value	50.0	30.0	1.5	0.005
MPLE	56.5	23.9	1.60	0.00557

- **2nd line.** The variable T_y (≥ 1) is provided here for the standardized coordinates of points in the rectangular region $[0, 1] \times [0, T_y]$ without loss of generality except for the scaling. The coordinates of points in the rectangular region are standardized. Note here that the coordinates of given data are considered the periodic boundary condition.
- **3rd line.** The initial guesses of the parameters μ , ν , a , σ_1 and σ_2 are provided in order to get the MPLEs.
- **4th line** describes the name of dataset that are standardized by the rectangular length of the smaller side that should be set as the x-coordinate.
- **5th line** indicates upper limit value **X2** in place of ∞ in the integral in (3.3), which is provided in such a way that $\mathbf{X2} > R/2$ where R is from (3.6). Here we set $\mathbf{X2} = 0.3$ ($> 1/4$).

Here we set the true values $(\mu, \nu, a, \sigma_1, \sigma_2) = (50.0, 30.0, 0.3, 0.005, 0.1)$ to apply the simulated data [**TypeA-offspring.xy**] (see **Figure6**).

OUTPUTS:

The output file [**TypeA.simplex.print**] provides the process of minimizing the negative Palm log-likelihood function until the values converge to the MPLEs for the given data. The outputs are arrayed as follows:

- In the **1st column**, characters **update** or **testfn** is written, where the **update** indicates that $-\log L$ value in the **2nd column** has attained the minimum so far, otherwise not.
- The **3rd column** and the rests show the values of the corresponding parameters as indicated in the **1st row**.
- From the **2nd row** to the **last row** show the optimization procedure starting from the initial values in the **2nd row**. The **last row** shows the minimized $-\log L$ and the corresponding MPLEs.

The other output file [**TypeA.simplex.x.print**] shows the same optimization procedure by the simplex method as indicated in above output file but with the normalized parameters depending on the initial scaling guess of the parameters. Namely, the actual estimates are obtained by the indicated **x-values** times the corresponding initial values.

- The **1st column** shows the smallest $-\log L$ value at the stage so far.
- The **2nd column** shows the standard deviations among function values of the vertices $\mathbf{x}(\mathbf{i})$ of the simplex.
- The **3rd column** and the rests show the values of the corresponding normalized parameters $\mathbf{x}(\mathbf{i})$ as indicated in the **1st row**.
- From the **2nd row** to the **last row** show the optimization procedure starting from the initial values in the **2nd row**, all of which should be 1.0. The **last row** shows the minimized $-\log L$ and the normalized variables corresponding to the MPLEs relative to the initial estimates. The **2nd column** is also expected to be small enough.

In the directory, in addition, the output files [**TypeA1000.simplex.print**] and [**TypeA1000.simplex.x.print**] for the case of **iskip = 1000** are also provided for the sake of users to check within a shorter c.p.u. time.

3.8 Inputs and outputs of the program [**Palm-TypeA**]

The program [**Palm-TypeA.f**] calculates the Palm intensity functions due to the numerical differential calculation of the $F_\tau(r)$ for given parameter sets of the Type A model associated with the non-parametric Palm intensity function estimated directly from any point pattern data.

INPUTS:

The following are given in the control file [**TypeAparam.palm**].

- **1st line** describes the dataset name of point pattern to delineate the non-parametric Palm intensity function.
- **2nd line.** A common value of Δ shows a width for the non-parametric Palm intensity function and the increment of numerical differential calculation of the $F_\tau(r)$. The value depends on the number of given point pattern data. Here we set $\Delta = 0.001$. The variable T_y (≥ 1) is provided here for the standardized coordinates of points in the rectangular region $[0, 1] \times [0, T_y]$ without loss of generality except for the scaling.
- **3rd line** indicates upper limit value **X2** in place of ∞ in the integral in (3.3), which is provided in such a way that **X2** $> R/2$ where R is from (3.6). Here we set **X2** = 0.3 ($> 1/4$).

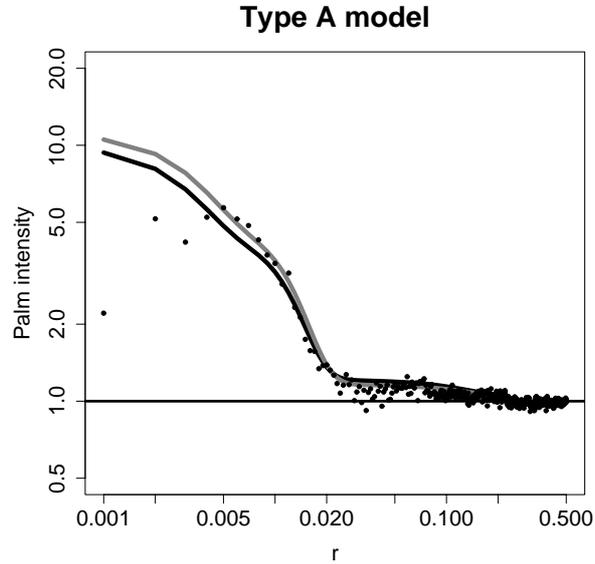


Figure 13: Palm intensity functions of the Type A model due to the parameters in **Table 3** in doubly logarithmic scales, dots are the non-parametric estimate, the black and gray curves are of the true and MPLE parameters, respectively, and the horizontal line shows the normalized intensity of the simulated point pattern. All the plots are normalized with respect to the mean intensity (the mean number of events per unit area).

Table 3: MPLE values of the Type A model from the simulated data.

Parameter	μ	ν	a	σ_1	σ_2
True value	50.0	30.0	0.3	0.005	0.1
MPLE	56.4	23.6	0.355	0.00520	0.107

- From the **4th line** to the **bottom line** (not more than 10th lines) describe respective value of the given parameters $(\mu_i, \nu_i, a_i, \sigma_{1,i}, \sigma_{2,i})$, $i = 1, 2, \dots, I \leq 7$, for respective estimates of the Palm intensity functions of the Type A model. If there are nothing in these lines (4th-10th lines), the following outputs provide only the non-parametric Palm intensity function, namely, only the 1st and 2nd columns.

OUTPUTS:

Then we have the outputs-file [**Palm-TypeA.txt**] in the following format.

- The **1st column** provides 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.
- The **2nd column** provides the corresponding values of the non-parametric Palm intensity function of r , which is normalized by the total intensity estimate (the mean number of points in a unit area) of the point pattern data.
- The **3rd-last columns** show the normalized Palm intensity functions $\lambda_o(r)/\hat{\lambda}$ calculated from the given sets of parameter values $(\mu_i, \nu_i, a_i, \sigma_{1,i}, \sigma_{2,i})$.

Here, as an example, we provide the non-parametric Palm intensity function with $\Delta = 0.001$ and two Palm intensity functions corresponding to the true and MPLE parameters, respectively (**Figure 13**).

3.9 Inputs and outputs of the program [**Simplex-TypeB**]

INPUTS:

The following variables are given in the four lines of the control file [**TypeB.param**], which is also used for the simulation program of the Type B model.

- **1st line.** All the variables in this line are ignored in the present case.

- **2nd line.** The variable T_y (≥ 1) is provided here for the standardized coordinates of points in the rectangular region $[0, 1] \times [0, T_y]$ without loss of generality except for the scaling. The coordinates of points in the rectangular region are standardized. Note here that the coordinates of given data are considered the periodic boundary condition.
- **3rd line.** The initial guesses of the parameters μ_1 , μ_2 , ν , σ_1 and σ_2 are provided in order to get the MPLEs.
- **4th line** describes the name of dataset that are standardized by the rectangular length of the smaller side that should be set as the x-coordinate.

Here we set the true values $(\mu_1, \mu_2, \nu, \sigma_1, \sigma_2) = (10.0, 40.0, 30.0, 0.01, 0.03)$ to apply the simulated data [**TypeB-offspring.xy**] (see **Figure 8**).

OUTPUTS:

The output file [**TypeB.simplex.print**] provides the process of minimizing the negative Palm log-likelihood function until the values converge to the MPLEs for the given data. The outputs are arrayed as follows:

- In the **1st column**, characters **update** or **testfn** is written, where the **update** indicates that $-\log L$ value in the **2nd column** has attained the minimum so far, otherwise not.
- The **3rd column** and the rests show the values of the corresponding parameters as indicated in the **1st row**.
- From the **2nd row** to the **last row** show the optimization procedure starting from the initial values in the **2nd row**. The **last row** shows the minimized $-\log L$ and the corresponding MPLEs

The other output file [**TypeB.simplex.x.print**] shows the same optimization procedure by the simplex method as indicated in above output file but with the normalized parameters depending on the initial scaling guess of the parameters. Namely, the actual estimates are obtained by the indicated **x-values** times the corresponding initial values.

- The **1st column** shows the smallest $-\log L$ value at the stage so far.

- The **2nd column** shows the standard deviations among function values of the vertices $\mathbf{x}(\mathbf{i})$ of the simplex.
- The **3rd column** and the rests show the values of the corresponding normalized parameters $\mathbf{x}(\mathbf{i})$ as indicated in the **1st row**.
- From the **2nd row** to the **last row** show the optimization procedure starting from the initial values in the **2nd row**, all of which should be 1.0. The **last row** shows the minimized $-\log L$ and the normalized variables corresponding to the MPLEs relative to the initial estimates. The **2nd column** is also expected to be small enough.

3.10 Inputs and outputs of the program [Palm-TypeB]

The program [Palm-TypeB.f] calculates the Palm intensity functions for given parameter sets of the Type B model associated with the non-parametric Palm intensity function estimated directly from any point pattern data.

INPUTS:

The following variables are given in the control file [TypeBparam.palm].

- **1st line** describes the dataset name of point pattern to delineate the non-parametric Palm intensity function.
- **2nd line.** A value of Δ shows a width for the non-parametric Palm intensity function. The value depends on the number of given point pattern data. Here we set $\Delta = 0.001$. The variable $T_y (\geq 1)$ is provided here for the standardized coordinates of points in the rectangular region $[0, 1] \times [0, T_y]$ without loss of generality except for the scaling.
- From the **3rd line** to the **bottom line** (not more than 10th lines) describe respective value of the given parameters $(\mu_i, \nu_i, a_i, \sigma_{1,i}, \sigma_{2,i})$, where $\mu_i = \mu_{1,i} + \mu_{2,i}$, $i = 1, 2, \dots, I \leq 8$, for respective estimates of the Palm intensity functions of the Type B model. If there are nothing in these lines (3rd-10th lines), the following outputs provide only the non-parametric Palm intensity function, namely, only the 1st and 2nd columns.

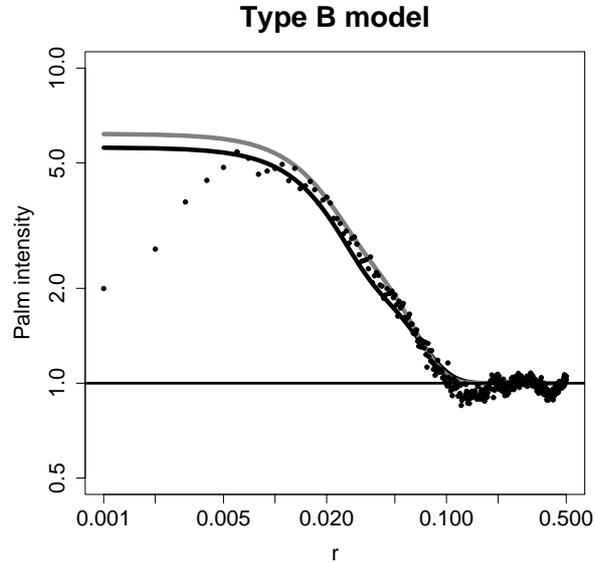


Figure 14: Palm intensity functions of the Type B model due to the parameters in **Table 4** in doubly logarithmic scales, dots are the non-parametric estimate, the black and gray curves are of the true and MPLE parameters, respectively, and the horizontal line shows the normalized intensity of the simulated point pattern. All the plots are normalized with respect to the mean intensity (the mean number of events per unit area).

Table 4: MPLE values of the Type B model from the simulated data.

Parameter	μ	ν	a	σ_1	σ_2
True value	50.0	30.0	0.20	0.01	0.03
MPLE	44.1	30.7	0.151	0.00945	0.0268

OUTPUTS:

Then we have the outputs-file [**Palm-TypeB.txt**] in the following format.

- The **1st column** provides 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.
- The **2nd column** provides the corresponding values of the non-parametric Palm intensity function of r , which is normalized by the total intensity estimate (the mean number of points in a unit area) of the point pattern data.
- The **3rd-last columns** show the normalized Palm intensity functions $\lambda_o(r)/\hat{\lambda}$ calculated from the given sets of parameter values $(\mu_i, \nu_i, a_i, \sigma_{1,i}, \sigma_{2,i})$.

Here, as an example, we provide the non-parametric Palm intensity function with $\Delta = 0.001$ and two Palm intensity functions corresponding to the true and MPLE parameters, respectively (**Figure 14**).

3.11 Inputs and outputs of the program [**Simplex-TypeC**]

INPUTS:

The following variables are given in the four lines of the control file [**TypeC.param**], which is also used for the simulation program of the Type C model.

- **1st line.** The variables in this line are ignored in the present case.
- **2nd line.** The variable T_y (≥ 1) is provided here for the standardized coordinates of points in the rectangular region $[0, 1] \times [0, T_y]$ without loss of generality except for the scaling. The coordinates of points in the rectangular region are standardized. Note here that the coordinates of given data are considered the periodic boundary condition.
- **3rd line.** The initial guesses of the parameters $\mu_1, \mu_2, \nu_1, \nu_2, \sigma_1$ and σ_2 are provided in order to get the MPLEs.

- **4th line** describes the name of dataset that are standardized by the rectangular length of the smaller side that should be set as the x-coordinate.

Here, for the initial guess, we set the true values $(\mu_1, \mu_2, \nu_1, \nu_2, \sigma_1, \sigma_2) = (5.0, 9.0, 30.0, 150.0, 0.01, 0.05)$ to apply the simulated data [**TypeC-offspring.xy**] (see **Figure10**).

OUTPUTS:

The output file [**TypeC.simplex.print**] provides the process of minimizing the negative Palm log-likelihood function until the values converge to the MPLEs for the given data. The outputs are arrayed as follows:

- In the **1st column**, characters **update** or **testfn** is written, where the **update** indicates that $-\log L$ value in the **2nd column** has attained the minimum so far, otherwise not.
- The **3rd column** and the rests show the values of the corresponding parameters as indicated in the **1st row**.
- From the **2nd row** to the **last row** show the optimization procedure starting from the initial values in the **2nd row**. The **last row** shows the minimized $-\log L$ and the corresponding MPLEs.

The other output file [**TypeC.simplex.x.print**] shows the same optimization procedure by the simplex method as indicated in above output file but with the normalized parameters depending on the initial scaling guess of the parameters. Namely, the actual estimates are obtained by the indicated **x**-values times the corresponding initial values.

- The **1st column** shows the smallest $-\log L$ value at the stage so far.
- The **2nd column** shows the standard deviations among function values of the vertices **x(i)** of the simplex.
- The **3rd column** and the rests show the values of the corresponding normalized parameters **x(i)** as indicated in the **1st row**.

- From the **2nd row** to the **last row** show the optimization procedure starting from the initial values in the **2nd row**, all of which should be 1.0. The **last row** shows the minimized $-\log L$ and the normalized variables corresponding to the MPLEs relative to the initial estimates. The **2nd column** is also expected to be small enough.

3.12 Inputs and outputs of the program [Palm-TypeC]

The program [Palm-TypeC.f] calculates the Palm intensity functions for given parameter sets of the Type C model associated with the non-parametric Palm intensity function estimated directly from any point pattern data.

INPUTS:

The following variables are given in the control file [TypeCparam.palm].

- **1st line** describes the dataset name of point pattern to delineate the non-parametric Palm intensity function.
- **2nd line.** A value of Δ shows a width for the non-parametric Palm intensity function. The value depends on the number of given point pattern data. Here we set $\Delta = 0.001$. The variable T_y (≥ 1) is provided here for the standardized coordinates of points in the rectangular region $[0, 1] \times [0, T_y]$ without loss of generality except for the scaling.
- From the **3rd line** to the **bottom line** (not more than 10th lines) describe respective value of the given parameters $(\lambda_i, \nu_{1,i}, a_i, \sigma_{1,i}, \sigma_{2,i})$, where $\lambda_i = \mu_{1,i}\nu_{1,i} + \mu_{2,i}\nu_{2,i}$, $i = 1, 2, \dots, I \leq 8$, for respective estimates of the Palm intensity functions of the Type C model. If there are nothing in these lines (3rd-10th lines), the following outputs provide only the non-parametric Palm intensity function, namely, only the 1st and 2nd columns.

OUTPUTS:

Then we have the outputs-file [Palm-TypeC.txt] in the following format.

- The **1st column** provides 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.

- The **2nd column** provides the corresponding values of the non-parametric Palm intensity function of r , which is normalized by the total intensity estimate (the mean number of points in a unit area) of the point pattern data.
- The **3rd-last columns** show the normalized Palm intensity functions $\lambda_o(r)/\hat{\lambda}$ calculated from the given sets of parameter values $(\lambda_i, \nu_{1,i}, a_i, \sigma_{1,i}, \sigma_{2,i})$.

Here, as an example, we provide the non-parametric Palm intensity function with $\Delta = 0.001$ and two Palm intensity functions corresponding to the true and MPLE parameters, respectively (**Figure 15**).

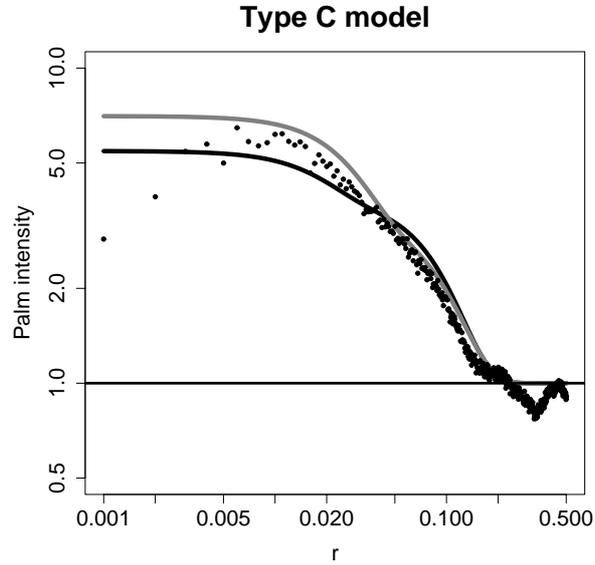


Figure 15: Palm intensity functions of the Type C model due to the parameters in **Table 5** in doubly logarithmic scales, dots are the non-parametric estimate, the black and gray curves are of the true and MPLE parameters, respectively, and the horizontal line shows the normalized intensity of the simulated point pattern. All the plots are normalized with respect to the mean intensity (the mean number of events per unit area).

Table 5: MPLE values of the Type C model from the simulated data.

Parameter	λ	ν_1	a	σ_1	σ_2
True value	1500.0	30.0	0.10	0.01	0.05
MPLE	1281.0	43.1	0.30	0.0149	0.0517

Acknowledgement

We are grateful to Jiancang Zhuang for his advices on R commands in drawing the figures.

References

- [1] Daley, D.J. and Vere-Jones, D. (2008). *An Introduction to the Theory of Point Processes*, Vol. II, General Theory and Structure, 2nd Ed., Springer-Verlag, New York.
- [2] Devroye, L. (1986). *Non-Uniform Random Variate Generation*, Springer-Verlag, New York.
- [3] Kowalik, J. and Osborne, M. R. (1968). *Methods for unconstrained optimization problems*, American elsevier publishing company, Inc. New York.
- [4] Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P. (1992). *Numerical Recipes in Fortran 77*, 2nd Ed., Cambridge University Press, Cambridge.
- [5] Stoyan, D., Kendall, W. S. and Mecke, J. (1995). *Stochastic Geometry and its Applications*, 2nd Ed., John Wiley & Sons, Chichester.
- [6] Stoyan, D. and Stoyan, H. (1994). *Fractals, Random Shapes and Point Fields*, John Wiley & Sons, Chichester.
- [7] Tanaka, U., Ogata, Y. and Stoyan, D. (2008). Parameter estimation and model selection for Neyman-Scott point processes, *Biometrical Journal* **50**, 43–57.
- [8] Wichmann, B. A. and Hill, I. D. (1982). An Efficient and Portable Pseudo-random Number Generator, *J. R. Statist. Soc. C*, **31**, 188–190.

The Institute of Statistical Mathematics

4-6-7 Minami-Azabu, Minato-ku

Tokyo 106-8569, Japan