Modelling Spatial Point Patterns in R

Adrian Baddeley[1] and Rolf Turner[2]

[1] School of Mathematics and Statistics
University of Western Australia
Crawley 6009 WA, Australia
adrian@maths.uwa.edu.au

[2] Department of Mathematics and Statistics
University of New Brunswick
Fredericton, N.B., Canada E3B 5A3
rolf@math.unb.ca

Summary. We describe practical techniques for fitting stochastic models to spatial point pattern data in the statistical package R. The techniques have been implemented in our package spatstat in R. They are demonstrated on two example datasets.

1 Introduction

This paper describes practical techniques for fitting stochastic models to spatial point pattern data using the statistical language R. The techniques are demonstrated with a detailed analysis of two real datasets.

We have implemented the techniques as a package spatstat in the R language. Both spatstat and R are freely available from the R website [19].

Sections 2 and 3 introduce the spatstat package. Theory of point process models is covered in Section 4, while Section 5 describes how to fit models in spatstat, and Section 6 explains how to interpret the fitted models obtained from the package. Models involving external covariates are discussed in Section 7, and models for multitype point patterns in Section 8. Estimation of irregular parameters is discussed in Section 9. Section 10 discusses formal inference for models. Examples are analysed in Sections 11–12.

2 The spatstat Package

We assume the reader is conversant with basic ideas of spatial point pattern analysis [28, 65] and with the R language [40, 38, 53].

Spatstat is a contributed R package for the analysis of spatial point pattern data [4]. It contains facilities for data manipulation, tools for exploratory data analysis, convenient graphical facilities, tools to simulate a wide range of point pattern models, versatile model-fitting capabilities, and model diagnostics. A detailed introduction to spatstat has been provided in [4]. Here we give a brief overview of the package.

2.1 Scope

Spatstat supports the following activities. Firstly **basic manipulation** of point patterns is supported; a point pattern dataset can easily be created, plotted, inspected, transformed and modified. **Exploratory data analysis** is possible using summary functions such as the K function, pair correlation function, empty space function, kernel-smoothed intensity maps, etc. (see e.g. [28, 65]). A key feature of spatstat is its generic algorithm for **parametric model-fitting** of spatial point process models to point pattern data. Models may exhibit spatial inhomogeneity, interpoint interaction (of arbitrary order), dependence on covariates, and interdependence between marks. Finally, **simulation** of point process models, including models fitted to data, is supported.

Figure 1 shows an example of a point pattern dataset which can be handled by the package; it consists of points of two types (plotted as two different symbols) and is observed within an irregular sampling region which has a hole in it. The label or 'mark' attached to each point may be a categorical variable, as in Fig. 1, or a continuous variable.

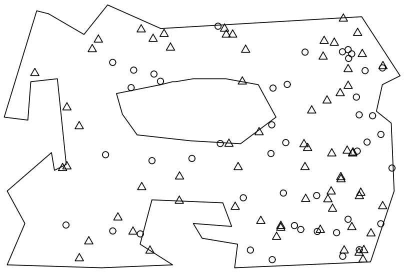


Fig. 1. Artificial example demonstrating the complexity of datasets which spatstat can handle.

Point patterns analysed in spatstat may also be spatially inhomogeneous, and may exhibit dependence on covariates. The package can deal with a variety of covariate data structures. It will fit point process models which depend on the covariates in a general way, and can also simulate such models.

2.2 Data Types in spatstat

A point pattern dataset is stored as a single 'object' `X` which may be plotted simply by typing `plot(X)`. Here spatstat uses the object-oriented features of

R ("classes and methods") to make it easy to manipulate, analyse, and plot datasets.

The basic data types in spatstat are Point Patterns, Windows, and Pixel Images. See Fig. 2. A point pattern is a dataset recording the spatial locations of all 'events' or 'individuals' observed in a certain region. A window is a region in two-dimensional space. It usually represents the 'study area'. A pixel image is an array of "brightness" values for each grid point in a rectangular grid inside a certain region. It may contain covariate data (such as a satellite image) or it may be the result of calculations (such as kernel smoothing).

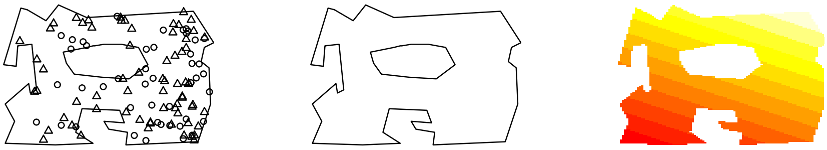


Fig. 2. A point pattern, a window, and a pixel image.

A point pattern is represented in spatstat by an object of the class `"ppp"`. A dataset in this format contains the coordinates of the points, optional 'mark' values attached to the points, and a description of the spatial region or 'window' in which the pattern was observed. Objects of class `"ppp"` can be created using the function `ppp`, converted from other data using the function `as.ppp`, or obtained in a variety of other ways.

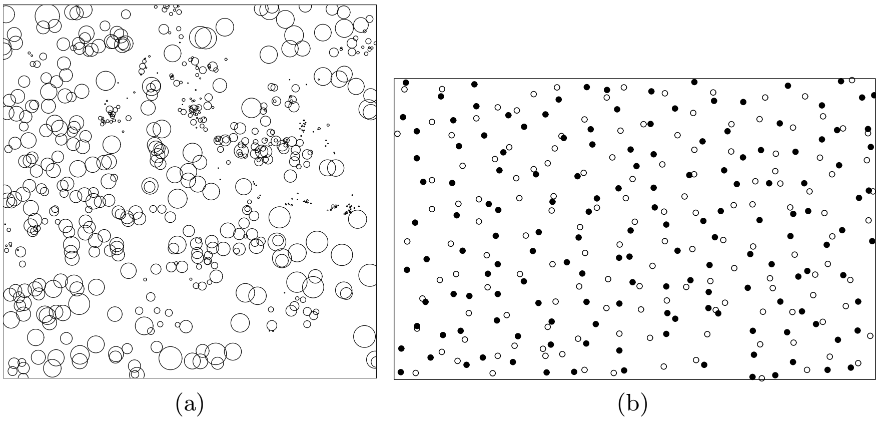
In our current implementation, the mark attached to each point must be a *single* value (which may be numeric, character, complex, logical, or factor). Figure 3(a) shows an example where the mark is a positive real number. A multitype point pattern is represented as a marked point pattern for which the mark is a categorical variable (a "factor" in R). Figure 3(b) shows an example where the mark is a categorical variable with two levels (i.e. a bivariate point pattern).

If `X` is a point pattern object then typing `X` or `print(X)` will print a short description of the point pattern; `summary(X)` will print a longer summary; and `plot(X)` will generate a plot of the point pattern on a correct scale. Numerous facilities are available for manipulating point pattern datasets.

3 Data Analysis in spatstat

3.1 Data Input

Point pattern datasets (objects of class `"ppp"`) can be entered into spatstat in various ways. We may create them from raw data using the function `ppp`,



(a) (b)

Fig. 3. Examples of marked point patterns. (a) continuous marks. Mark values (tree diameters) represented as radii of circles. The Longleaf Pines dataset, available as `longleaf`. (b) categorical marks. Mark values (cell types) represented as different graphical symbols. Hughes' amacrine cell dataset, available as `amacrine`.

convert data from other formats (including other packages) using `as.ppp`, read data from a file using `scanpp`, manipulate existing point pattern objects using a variety of tools, or generate a random pattern using one of the simulation routines.

Suppose, for example, that we have data for a point pattern observed in the rectangle $[0, 10] \times [0, 4]$. Assume the Cartesian coordinates of the points are stored in R as vectors `x` and `y`. Then the command

```
X <- ppp(x, y, c(0,10), c(0,4))
```

creates a point pattern object containing this information.

3.2 Initial Inspection of Data

Chatfield [15] emphasises the importance of careful initial inspection of data. The same principles apply to point pattern data. A point pattern dataset should be inspected for the following: omission of data points; transcription errors; data file format violations; incorrect scaling of the coordinates; flipping of the axes; errors in delimiting the boundary; errors in inclusion/exclusion of points near the boundary; incorrect interpretation of the data type of the marks (e.g. categorical or continuous); inconsistency with plots of the same data in the original source publication; coarse rounding of the Cartesian coordinates; use of values such as 99 or -1 to indicate a missing value; incorrect software translation of the levels of a factor; and duplicated points.

Inspection can be accomplished in `spatstat` mainly with the commands `plot`, `print`, `summary`, `identify`, `hist` (to examine values of the Cartesian coordinates) and `nndist` (to detect duplicated points).

3.3 Exploratory Data Analysis

Before stochastic modelling of a point pattern dataset is attempted, and certainly before any formal hypothesis testing is contemplated, the data should be subjected to exploratory data analysis. General principles of EDA are outlined in [15, 18, 24, 68, 72]. Numerous tools for exploratory analysis of spatial point pattern data are surveyed in [22, 26, 28, 56, 58, 64, 65, 72].

In particular, the assumption of stationarity ('spatial homogeneity') is an essential requirement for many of the classical methods for spatial point pattern analysis [22, 26, 28, 55, 56, 58, 64, 65]. It seems clear that many real point patterns cannot be described as stationary [51, 66], and the use of the classical methods on such data would be invalid. Hence it is extremely important that the homogeneity of a point pattern dataset be critically evaluated. Techniques for analysing nonstationary ('spatially inhomogeneous') patterns are less developed [3, 49, 50, 51, 52, 66].

An exploratory analysis should typically begin with an assessment of spatial inhomogeneity using tools such as the kernel smoothed estimate of intensity [25] (available in `spatstat` as `ksmooth.ppp`), or LISA (Local Indicators of Spatial Association) methods [1, 21, 20]. The dataset could also be partitioned manually using the subset operator `[` or the commands `cut` and `split`.

If a simple form of spatial inhomogeneity (such as a gradient from left to right) is suspected, this trend can be fitted using parametric methods as described in Section 6.

If the data are judged to be spatially homogeneous, the next step would be exploratory analysis using standard summary statistics such as Ripley's K function. A wide choice of summary statistics is now available [22, 28, 58, 65, 64]. In `spatstat` the available choices include `Kest`, which estimates the K function [55, 56],[65, Chapter 15]; `Fest`, estimating the empty space function F [56, 58] also known as the contact distribution function [65, Chapter 15] and point-event distance function [26, Sect. 2.4]; `Gest`, estimating the nearest neighbour distance distribution function G [26, Sect. 2.3],[65, Chapter 15]; `pcf`, the pair correlation function [65, Chapter 15]; `Jest`, the function $J(r) = (1-G(r))/(1-F(r))$ of [70]; `Kmeasure`, the reduced second moment measure [9, 10], [65, pp. 245, 247], [64]; and analogues of these functions for multitype and marked point patterns [71].

However if the data are judged to be spatially inhomogeneous, then at present there is limited scope for further exploratory analysis. One exception is the inhomogeneous K function [3] implemented in `spatstat` as `Kinhom`.

4 Point Process Models

The `spatstat` package can fit parametric models of spatial point processes to point pattern data. This section describes the relevant class of models, and the next section explains how to fit them using `spatstat`.

A typical realisation of a point pattern X in the bounded region $W \subset \mathbb{R}^2$ will be denoted

$$\mathbf{x} = \{x_1, \ldots, x_n\}$$

where $x_i \in W$ are the individual points of the process, and the total number of points $n \geq 0$ is not fixed.

4.1 Formulation of Models

The point process models fitted in spatstat are Gibbs point processes, cf. [5, 48, 69]. The scope of possible models is very wide: they may include spatial trend, dependence on covariates, interpoint interactions of any order (i.e. we are not restricted to pairwise interactions), and dependence on marks.

Each model will be specified in terms of its *conditional intensity* rather than its likelihood. This turns out to be an intuitively appealing way to formulate point process models, as well as being necessary for technical reasons.

The (Papangelou) conditional intensity is a function $\lambda(u, \mathbf{x})$ of spatial location $u \in W$ and of the entire point pattern $\mathbf{x}$. Roughly speaking, if we consider an infinitesimal region around the point u of area $\mathrm{d}u$, then the conditional probability that the point process contains a point in this infinitesimal region, given the position of all points outside this region, is $\lambda(u, \mathbf{x})\,\mathrm{d}u$. See [5, 17] and the excellent surveys by Ripley [58, 59].

For example, the homogeneous Poisson process (complete spatial randomness, CSR) has conditional intensity

$$\lambda(u, \mathbf{x}) = \beta \tag{1}$$

where β is the intensity (expected number of points per unit area). The inhomogeneous Poisson process with local intensity function $\beta(u)$, $u \in \mathbb{R}^2$, has conditional intensity

$$\lambda(u, \mathbf{x}) = \beta(u). \tag{2}$$

The Strauss process, a simple model of dependence between points, has conditional intensity

$$\lambda(u, \mathbf{x}) = \beta \gamma^{t(u, \mathbf{x})} \tag{3}$$

where $t(u, \mathbf{x})$ is the number of points of the pattern $\mathbf{x}$ that lie within a distance r of the location u. Here γ is the interaction parameter, satisfying $0 \leq \gamma \leq 1$, and $r > 0$ is the interaction radius.

The conditional intensity is a useful modelling tool because its functional form has a straightforward interpretation. The simplest form is a constant, $\lambda(u, \mathbf{x}) \equiv \beta$, which corresponds to "complete spatial randomness" (a uniform Poisson process). In most applications, this would be the null model. A conditional intensity $\lambda(u, \mathbf{x})$ which depends only on the location u, say $\lambda(u, \mathbf{x}) = \beta(u)$, corresponds to an inhomogeneous Poisson process with intensity function $\beta(u)$. In this case the functional form of $\beta(u)$ indicates the type of inhomogeneity (or "spatial trend").

A conditional intensity $\lambda(u, \mathbf{x})$ which depends on the point pattern $\mathbf{x}$, as well as on the location u, corresponds to a point process which exhibits stochastic dependence between points. For example, in the Strauss process (3) with $\gamma < 1$, dependence between points is reflected in the fact that the conditional probability of finding a point of the process at the location u is reduced if other points of the process are present within a distance r. In the special case $\gamma = 0$, the conditional probability of finding a point at u is zero if there are any other points of the process within a distance r of this location.

4.2 Scope of Models

Our technique [5] fits any model for which the conditional intensity is of the loglinear form

$$\lambda(u, \mathbf{x}) = \exp(\psi^{\mathsf{T}} B(u) + \varphi^{\mathsf{T}} C(u, \mathbf{x})) \tag{4}$$

where $\theta = (\psi, \varphi)$ are the parameters to be estimated. Both ψ and φ may be vectors of any dimension, corresponding to the dimensions of the vector-value statistics $B(u)$ and $C(u, \mathbf{x})$ respectively.

The term $B(u)$ depends only on the spatial location u, so it represents "spatial trend" or spatial covariate effects. The term $C(u, \mathbf{x})$ represents "stochastic interactions" or dependence between the points of the random point process. For example $C(u, \mathbf{x})$ is absent if the model is a Poisson process.

Gibbs models may require reparametrisation in order to conform to (4). For example, the Strauss process conditional intensity (3) satisfies (4) if we set $B(u) \equiv 1$ and $C(u, \mathbf{x}) = t(u, \mathbf{x})$, and take the parameters to be $\psi = \log \beta$ and $\varphi = \log \gamma$.

In practice there is an additional constraint that the terms $B(u)$ and $C(u, \mathbf{x})$ must be implemented in software. Some point process models which belong to the class of Gibbs processes have a conditional intensity which is difficult to evaluate. Notable examples include Cox processes [8]. For these models, other approaches should be used [48].

4.3 Model-fitting Algorithm

Our software currently fits models by the method of maximum pseudolikelihood (in Besag's sense [13]), using a computational device developed for Poisson models by Berman & Turner [12] which we adapted to pseudolikelihoods of general Gibbs point processes in [5]. Although maximum pseudolikelihood may be statistically inefficient [42, 43], it is adequate in many practical applications [60] and it has the virtue that we can implement it in software with great generality. Future versions of spatstat will implement the Huang-Ogata improvement to maximum pseudolikelihood [39] which is believed to be highly efficient.

Let the point pattern dataset $\mathbf{x}$ consist of n points $x_1, \ldots, x_n$ in a spatial region $W \subseteq \mathbb{R}^d$. Consider a point process model governed by a parameter

θ and having conditional intensity $\lambda_\theta(u, \mathbf{x})$. The *pseudolikelihood* [13] of the model is

$$\mathrm{PL}(\theta; \mathbf{x}) = \prod_{i=1}^{n} \lambda_\theta(x_i; \mathbf{x}) \exp\left(- \int_W \lambda_\theta(u; \mathbf{x}) \, \mathrm{d}u \right) \tag{5}$$

The *maximum pseudolikelihood estimate* of θ is the value which maximises $\mathrm{PL}(\theta; \mathbf{x})$. Now discretise the integral in (5) to obtain

$$\int_W \lambda_\theta(u; \mathbf{x}) \, \mathrm{d}u \approx \sum_{j=1}^{m} \lambda_\theta(u_j; \mathbf{x}) w_j . \tag{6}$$

where $u_j \in W$ are 'quadrature points' and $w_j \geq 0$ the associated 'quadrature weights' for $j = 1, \ldots, m$. The quadrature scheme should be chosen so that (6) is a good approximation.

The Berman-Turner [12] device involves choosing a set of quadrature points $\{u_j\}$ which *includes all the data points* x_i as well as some other ("dummy") points. Let z_j be the indicator which equals 1 if u_j is a data point, and 0 if it is a dummy point. Then the logarithm of the pseudolikelihood can be approximated by

$$\begin{aligned} \log \mathrm{PL}(\theta; \mathbf{x}) &\approx \sum_{j=1}^{m} [z_j \log \lambda_\theta(u_j; \mathbf{x}) - w_j \lambda_\theta(u_j; \mathbf{x})] \\ &= \sum_{j=1}^{m} w_j (y_j \log \lambda_j - \lambda_j) \end{aligned} \tag{7}$$

where $y_j = z_j / w_j$ and $\lambda_j = \lambda_\theta(u_j, \mathbf{x})$. The key to the Berman-Turner device is to recognise that the right hand side of (7) has the same functional form as the log likelihood of m independent Poisson random variables Y_j with means λ_j and responses y_j. This enables us to maximise the pseudolikelihood using standard statistical software for fitting generalised linear models.

Given a point pattern dataset and a model of the form (4), our algorithm constructs a suitable quadrature scheme $\{(u_j, w_j)\}$, evaluates the vector valued sufficient statistic $s_j = (B(u_j), C(u_j, \mathbf{x}))$, forms the indicator variable z_j and the pseudo-response $y_j = z_j / w_j$, then calls standard R software to fit the Poisson loglinear regression model $Y_j \sim \mathsf{Poisson}(\lambda_j)$ where $\log \lambda_j = \theta s_j$. The fitted coefficient vector $\widehat{\theta}$ given by this software is returned as the maximum pseudolikelihood estimate of θ. For further explanation see [5]. Advantages of using existing software to compute the fitted coefficients include its numerical stability, reliability, and most of all, its flexibility.

5 Model-fitting in spatstat

5.1 Overview

The model-fitting function is called `ppm` and is strongly analogous to `lm` or `glm`. In simple usage, it is called in the form

```
ppm(X, trend, interaction, ...)
```

where `X` is the point pattern dataset, `trend` describes the spatial trend (the function $B(u)$ in equation (4)) and `interaction` describes the stochastic dependence between points in the pattern (the function $C(u, \mathbf{x})$ in equation (4)). Other arguments to `ppm` may provide covariates, select edge corrections, and control the fitting algorithm.

For example

```
ppm(X, ~1, Strauss(r=0.1), ....)
```

fits the stationary Strauss process (3) with interaction radius $r = 0.1$. The spatial trend formula `~1` is a constant, meaning the process is stationary. The argument `Strauss(r=0.1)` is an object representing the interpoint interaction structure of the Strauss process with interaction radius $r = 0.1$.

Similarly

```
ppm(X, ~x + y, Poisson())
```

fits the non-stationary Poisson process with a *loglinear* intensity of the form

$$\beta(x, y) = \exp(\theta_0 + \theta_1 x + \theta_2 y)$$

where $\theta_0, \theta_1, \theta_2$ are (scalar) parameters to be fitted, and x, y are the Cartesian coordinates.

5.2 Spatial Trend Terms

The `trend` argument of `ppm` describes any spatial trend and covariate effects. It must be a `formula` expression in the R language, and serves a role analogous to the formula for the linear predictor in a generalised linear model. See e.g. [72, sect 6.2].

The right hand side of `trend` specifies the function $B(u)$ in equation (4) following the standard R syntax for a linear predictor. The terms in the formula may include the reserved names `x, y` for the Cartesian coordinates. Spatial covariates may also appear in the trend formula as we explain in Section 7.

Effectively, the function $B(u)$ in (4) is treated as the ‘systematic’ component of the model. Note that the link function is always the logarithm, so the model formula in a `ppm` call is always a description of the **logarithm** of the conditional intensity.

The default trend formula is `~1`, which indicates $B(u) \equiv 1$, corresponding to a process without spatial trend or covariate effects. The formula `~x` indicates the vector statistic $B((x, y)) = (1, x)$ corresponding to a spatial trend of the form $\exp(\psi B((x, y))) = \exp(\alpha + \beta x)$, where α, β are coefficient parameters to be estimated, while `~x + y` indicates $B((x, y)) = (1, x, y)$ corresponding to $\exp(\psi B((x, y))) = \exp(\alpha + \beta x + \gamma y)$.

A wide variety of model terms can easily be constructed from the Cartesian coordinates. For example

```
ppm(X, ~ ifelse(x > 2, 0, 1), Poisson())
```

fits an inhomogeneous Poisson process with different, constant intensities on each side of the line $x = 2$.

spatstat provides a function polynom which generates polynomials in 1 or 2 variables. For example

```
~ polynom(x, y, 2)
```

represents a polynomial of order 2 in the Cartesian coordinates x and y. This would give a "log-quadratic" spatial trend.[3]
Similarly

```
~ harmonic(x, y, 2)
```

represents the most general *harmonic* polynomial of order 2 in x and y.

Other possibilities include B-splines and smoothing splines, fitted with bs and s respectively. These terms introduce smoothing penalties, and thus provide an implementation of "penalised maximum pseudolikelihood" estimation (cf. [30]).

The special term offset can also be used in the trend formula. It has the same role in ppm as it does in other model-fitting functions, namely to add to the linear predictor a term which is not associated with a parameter. For example

```
~ offset(x)
```

will fit the model with log trend $\beta + x$ where β is the only parameter to be estimated.

Observed spatial covariates may also be included in the trend formula; see Section 7 below.

5.3 Interaction Terms

The dependence structure or 'interpoint interaction' in a point process model is determined by the function $C(u, \mathbf{x})$ in (4). This term is specified by the interaction argument of ppm, which is strongly analogous to the family argument to glm. Thus, interpoint interaction is regarded as a "distributional" component of the point process model, analogous to the distribution family in a generalised linear model.

The interaction argument is an object of a special class "interact". The user creates such objects using specialised spatstat functions, similar to those which create the family argument to glm. For example, the command

[3] We caution against using the standard function poly for the same purpose here. For a model formula containing poly, prediction of the fitted model can be erroneous, for reasons which are well-known to R users. The function polynom provided in spatstat does not exhibit this problem.

```
Strauss(r=0.1)
```

will create an object of class `"interact"` representing the interaction function $C(u, \mathbf{x})$ for the Strauss process (3) with interaction radius r. This object is then passed to the model-fitting function `ppm`, usually in the direct form

```
ppm(cells, ~1, Strauss(r=0.1))
```

The following functions are supplied for creating interpoint interaction structures; details of these models can be consulted in [5].

`Poisson`	Poisson process
`Strauss`	Strauss process
`StraussHard`	Strauss process with a hard core
`Softcore`	Pairwise soft core interaction
`PairPiece`	Pairwise interaction, step function potential
`DiggleGratton`	Diggle-Gratton potential
`LennardJones`	Lennard-Jones potential
`Geyer`	Geyer's saturation process
`OrdThresh`	Ord's process, threshold on cell area

Note that `ppm` estimates only the "canonical" parameters of a point process model. These are parameters θ such that the conditional intensity is loglinear in θ, as in equation (4). Other so-called "irregular" parameters (such as the interaction radius r of the Strauss process) cannot be estimated by the Berman-Turner device, and their values must be specified a priori, as arguments to the interaction function. Estimation of irregular parameters is discussed in Section 9.

For more advanced use, the following functions will accept "user-defined potentials" in the form of an arbitrary R language function. They effectively allow arbitrary point process models of these three classes.

`Pairwise`	Pairwise interaction, user-supplied potential
`Ord`	Ord model, user-supplied potential
`Saturated`	Saturated pairwise model, user-supplied potential

6 Fitted Models

The value returned by `ppm` is a "fitted point process model" of class `"ppm"`. It can be stored, inspected, plotted, predicted and updated. The following would be typical usage:

```
fit <- ppm(X, ~1, Strauss(r=0.1), ...)
fit
plot(fit)
pf <- predict(fit)
coef(fit)
```

Methods are provided for the following generic operations applied to `"ppm"` objects:

`print`	Print basic information
`summary`	Print extensive summary information
`coef`	Extract fitted model coefficients
`plot`	Plot fitted intensity
`fitted`	Compute fitted conditional intensity or trend at data points
`predict`	Compute predictions (spatial trend, conditional intensity)
`update`	Update the fit

Printing the fitted object `fit` will produce text output describing the fitted model. Plotting the object will display the spatial trend and the conditional intensity, as perspective plots, contour plots and image plots.

6.1 Interpretation of Fitted Coefficients

The easiest way to interpret a fitted point process model is to print it at the terminal. The `print` method attempts to produce a comprehensible description. For example,

```
> ppm(swedishpines, ~1, Strauss(7))
Stationary Strauss process
First order term:
      beta
0.01823799
Interaction: Strauss process
interaction distance:   7
Fitted interaction parameter gamma:
[1] 0.2472
```

Thus the fitted model is the stationary Strauss process (3) with parameters $\beta = 0.01823799$ and $\gamma = 0.2472$.

Alternatively the coefficients of the fitted model may be extracted using `coef`. These should be interpreted as the canonical parameters $\theta = (\psi, \varphi)$ appearing in (4). For example

```
> u <- ppm(swedishpines, ~1, Strauss(7))
> coef(u)
(Intercept) Interaction
  -4.004248   -1.397759
```

Comparing (4) with (3) we see that the usual parameters β, γ of the Strauss process are $\beta = \exp \psi$ and $\gamma = \exp \varphi$, so typing

```
> exp(coef(u))
(Intercept) Interaction
 0.01823799  0.24715026
```

shows that the fitted parameters are $\beta = 0.0182$ and $\gamma = 0.2472$.

If the model includes a spatial trend, then the fitted *canonical* coefficients of the trend will be presented using the standard R conventions. For example

```
>  ppm(swedishpines, ~x)
Nonstationary Poisson process
Trend formula: ~x
Fitted coefficients for trend formula:
(Intercept)           x
-5.13707319  0.00462614
```

indicates that the fitted model is a Poisson process with intensity function

$$\beta((x, y) = \exp(-5.13707319 + 0.00462614x).$$

In more complex models, the interpretation of the fitted coefficients may depend on the choice of `contrasts` for the coefficients of linear models. For example, if the treatment contrasts [72, sect 6.2] are in force, then a model involving a factor will be printed as follows:

```
> ppm(swedishpines, ~factor(ifelse(x < 50, "left", "right")))
Nonstationary Poisson process
Trend formula: ~factor(ifelse(x < 50, "left", "right"))
Fitted coefficients for trend formula:
                                  (Intercept)
                                   -5.075
factor(ifelse(x < 50, "left", "right"))right
                                    0.331
```

The explanatory variable is a factor with two levels, `left` and `right`. By default the levels are sorted alphabetically. Since we are using the treatment contrasts, the value labelled "(Intercept)" is the fitted coefficient for the first level (`left`), while the value labelled "right" is the estimated treatment contrast (to be *added* to the intercept) for the level `right`. This indicates that the fitted model is a Poisson process with intensity $\exp(-5.075) = 0.00625$ on the left half of the dividing line $x = 50$, and intensity $\exp(-5.075 + 0.331) = 0.00871$ on the right side.

6.2 Invalid Models

For some values of the parameters, a point process model with conditional intensity (4) may be 'invalid' or 'undefined', in the sense that the corresponding probability density is not integrable. For example, the Strauss process (3) is defined only for $0 \leq \gamma \leq 1$ (equivalently for $\varphi \leq 0$); the density is not integrable if $\gamma > 1$, as famously announced in [44].

A point process model fitted by `ppm` may sometimes be invalid in this sense. For example, a fitted Strauss process model may sometimes have a value of γ greater than 1. This happens because, in the Berman-Turner device, the conditional intensity (4) is treated as if it were the mean in a Poisson loglinear regression model. The latter model is well-defined for all values of the linear predictor, so the software does not constrain the values of the canonical parameters ψ, φ in (4).

The spatstat package has internal procedures for deciding whether a fitted model is valid, and for mapping or 'projecting' an invalid model to the nearest valid model. Currently these procedures are invoked only when we simulate a realisation of the fitted model. They are *not* invoked when a model is printed or when it is returned from `ppm`, so that the printed output from `ppm` may represent an invalid model.

6.3 Predicting and Plotting a Fitted Model

The `predict` method for a fitted point process model computes either the fitted spatial trend

$$\tau(u) = \exp(\widehat{\psi} B(u)) \tag{8}$$

or the fitted conditional intensity

$$\lambda_{\widehat{\theta}}(u, \mathbf{x}) = \exp(\widehat{\psi} B(u) + \widehat{\varphi} C(u, \mathbf{x})) \tag{9}$$

at arbitrary locations u. Note that $\mathbf{x}$ is always taken to be the observed data pattern to which the model was fitted.

The default behaviour is to produce a pixel image of both trend and conditional intensity, where these are nontrivial. A typical example is the following:

```
data(cells)
m <- ppm(cells,~polynom(x,y,2),Strauss(0.05), rbord=0.05)
trend <- predict(m,type="trend",ngrid=100)
cif   <- predict(m,type="cif",ngrid=100)
```

The resulting objects `trend` and `cif` are pixel images. One could then plot the resulting surfaces with calls like

```
persp(trend)
persp(cif, theta=-30,phi=40,d=4,ticktype="detailed",zlab="z")
```

We caution again that the result of `predict` may be incorrect if the trend formula of the point process model contains one of the functions `poly`, `bs`, `lo`, or `ns`.

The plot method (`plot.ppm`) will take a fitted point process model and plot the trend and/or the conditional intensity. By default this surface is calculated at a 40 × 40 grid of points on the (enclosing rectangle of) the observation window. The plots may be produced as perspective plots, images, or contour plots. For example

```
plot(fit,cif=FALSE,how="persp")
```

will generate a perspective plot of the fitted trend, where `fit` is the fitted model.

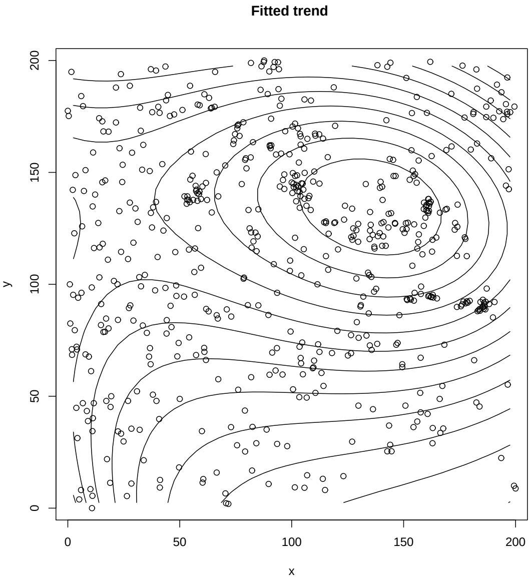


Fig. 4. Contour plot of fitted log-cubic trend for Longleaf Pines data (locations only) obtained using `plot.ppm`.

7 Models with Covariates

In order to model the dependence of a point pattern on a spatial covariate, there are several requirements. First, the covariate must be a quantity $Z(u)$ observable (in principle) at each location u in the window (e.g. altitude, soil pH, or distance to another spatial pattern). There may be several such covariates, and they may be continuous valued or factors. Second, the values $Z(x_i)$ of Z at each point of the data point pattern must be available. Thirdly, the values $Z(u)$ at *some* other points u in the window must be available. The accuracy of the algorithm depends on the number of these additional points and on their spatial arrangement. For a good approximation to the pseudolikelihood, the density of the additional points should be high throughout the window.

The argument `covariates` to the function `ppm` specifies the values of the spatial covariates. It may be either a data frame or a list of pixel images.

(a) If `covariates` is a list of pixel images, then each image is assumed to contain the values of a spatial covariate at a fine grid of spatial locations. The names of the list entries should be the names of the covariates used in the trend formula when you call `ppm`.

(b) If `covariates` is a data frame, then the ith row of the data frame is expected to contain the covariate values for the ith ‘quadrature point’ (see below). The column names of the data frame should be the names of the covariates used in the trend formula when you call `ppm`.

7.1 Covariates in a List of Images

The format (a), in which `covariates` is a list of images, would typically be used when the covariate values are computed from other data.

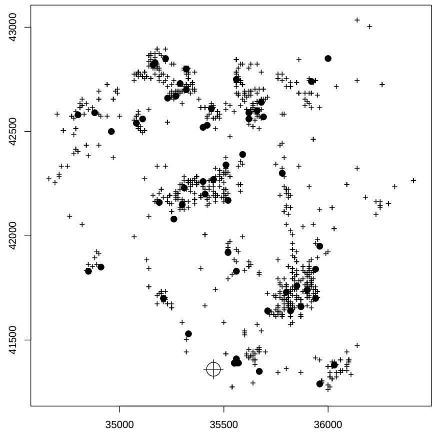


Fig. 5. The Chorley-Ribble data [27]. Cases of cancer of the larynx (●) and lung (+) in the Chorley-Ribble region of Lancashire, England, and the location of a disused industrial incinerator (⊕).

For example, Fig. 5 shows a spatial epidemiological dataset containing a point pattern `X` of disease cases, and another point pattern `Y` of control cases. We want to model `X` as a point process with intensity proportional to the local density ϱ of the susceptible population. We estimate ϱ by taking a kernel-smoothed estimate of the intensity of `Y`. Thus

```
rho.hat <- ksmooth.ppp(Y, sigma=1.2)
ppm(X, ~offset(log(rho)), covariates=list(rho=rho.hat))
```

The first line computes the values of the kernel-smoothed intensity estimate at a fine grid of pixels, and stores them in the pixel image object `rho.hat` (plotted in Fig. 6). The second line fits the Poisson process model with log intensity

$$\log \lambda(u) = \psi + \log \varrho(u) \qquad (10)$$

where ψ is an unknown parameter; that is, it fits the Poisson model with intensity

$$\lambda(u) = \mu \, \varrho(u) \qquad (11)$$

where $\mu = \mathrm{e}^{\psi}$ is the only parameter to be estimated. Note that `covariates` must be a list, even though there is only one covariate. The variable name `rho` in the model formula must match the name `rho` in the list.

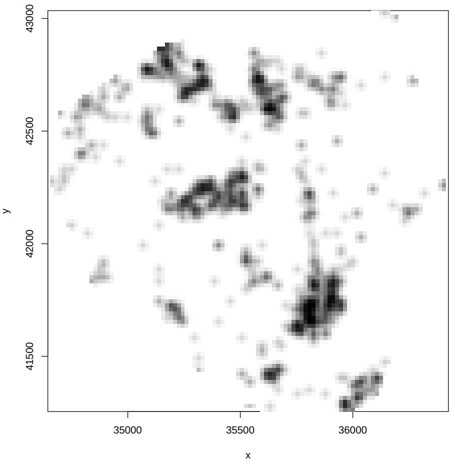


Fig. 6. Kernel smoothed intensity estimate $\widehat{\varrho}(u)$ of the lung cancer data from Fig. 5, which serves as a spatial covariate for modelling the laryngeal cancer data.

Typical output is as follows:

```
> ppm(X, ~offset(log(rho)), Poisson(), data=list(rho=rho.hat))
Nonstationary Poisson process
Trend formula:  ~offset(log(rho))
Fitted coefficients for trend formula:
(Intercept)
  -2.889
```

This indicates that the estimate of the parameter ψ in (10) is $\widehat{\psi} = -2.889$. Equivalently the estimate of μ in (11) is $\widehat{\mu} = \mathrm{e}^{-2.889} = 0.056$.

More complex models may be fitted to explore other effects by adding terms to the trend formula. For example

```
ppm(X, ~ x + offset(log(rho)), data=list(rho=rho.hat))
```

would fit a nonstationary Poisson model with intensity

$$\beta((x, y)) = \mathrm{e}^{\psi + \varphi x} \varrho((x, y)).$$

Covariates represented by pixel images in spatstat may have values that are numerical, complex, logical, or character strings. Unfortunately a pixel image in spatstat cannot have categorical (factor) values, because R refuses to create a `factor`-valued matrix. In order to represent a categorical variate as a pixel image, the categorical values should be encoded as integers (for efficiency's sake) and assigned to an integer-valued pixel image. Then the model formula should invoke the `factor` command on this image. For example if `fim` is an image with integer values which represent levels of a factor, then

```
    ppm(X, ~factor(f), Poisson(), covariates=list(f=fim))
```

fits the nonstationary Poisson process with an intensity that depends on the levels of this factor. Care must be taken to ensure the correct interpretation of the factor levels [72, p. 22 ff.].

7.2 Covariates in a Data Frame

Typically we would use the data frame format (b) if the values of the spatial covariates can only be observed at certain locations. We need to force `ppm` to use these locations to fit the model. That is, these locations must be used as the quadrature points u_j in the Berman-Turner approximation (6).

The function `ppm` may be called in the form

```
    ppm(Q, trend, interaction, ...)
```

where `Q` is a 'quadrature scheme' and the other arguments are unchanged. A quadrature scheme in `spatstat` is an object of a special class `"quad"` which comprises both *'data points'* (the points of the observed point pattern) and *'dummy points'* (some other locations in the window). It is usually created using the function `quadscheme`.

In the present context we will need to create a quadrature scheme based on the spatial locations where the covariate Z has been observed. Then the values of the covariate at these locations are passed to `ppm` through the data frame `covariates`.

For example, suppose that `X` is the observed point pattern and we are trying to model the effect of soil acidity (pH). Suppose we have measured the values of soil pH at the points x_i of the point pattern, and stored them in a vector `XpH`. Suppose we have measured soil pH at some other locations u in the window, and stored the results in a data frame `U` with columns `x, y, pH`. Then do as follows:

```
  Q <- quadscheme(data=X, dummy=list(x=U$x, y=U$y))
  df <- data.frame(pH=c(XpH, U$pH))
```

Then the rows of the data frame `df` correspond to the quadrature points in the quadrature scheme `Q`. To fit just the effect of pH, we type

```
  ppm(Q, ~ pH, Poisson(), covariates=df)
```

where the term `pH` in the formula `pH` agrees with the column label `pH` in the argument `covariates = df`. This will fit an inhomogeneous Poisson process with intensity that is a loglinear function of soil pH. We could can also try (say)

```
ppm(Q, ~ pH, Strauss(r=1), covariates=df)
ppm(Q, ~ factor(pH > 7), Poisson(), covariates=df)
ppm(Q, ~ polynom(x, 2) * factor(pH > 7), covariates=df)
```

8 Fitting Models to Multitype Point Patterns

The function ppm will also fit models to multitype point patterns. A multitype point pattern is a point pattern in which the points are each classified into one of a finite number of possible types (e.g. species, colours, on/off states). In spatstat a multitype point pattern is represented by a "ppp" object X whose marks are a factor. Fig. 3(b) shows an example.

Currently, ppm will not fit models to a marked point pattern if the marks are not a factor.

8.1 Conditional Intensity

A multitype point process in a region $W \subset \mathbb{R}^2$, with a set $\mathcal{M}$ of possible types, may be regarded as a point process in $W \times \mathcal{M}$. Let $\mathbf{y} = \{(x_1, m_1), \ldots, (x_n, m_n)\}$ denote a typical realisation of the process, where $x_i \in W$ are the locations and $m_i \in \mathcal{M}$ the corresponding marks (types).

The conditional intensity is now of the form $\lambda((u, m), \mathbf{y})$, where $u \in W$ and $m \in \mathcal{M}$. It has the interpretation that $\lambda((u, m), \mathbf{y})\, \mathrm{d}u$ is the conditional probability of finding a point *of type* m in an infinitesimal neighbourhood of the point u, given that the rest of the process coincides with $\mathbf{y}$.

This introduces some subtleties. A conditional intensity function which is constant,

$$\lambda((u, m), \mathbf{y}) = \beta \tag{12}$$

corresponds to a process in which **the points of each type** $m \in \mathcal{M}$ constitute a uniform Poisson process with intensity β. By standard properties of the Poisson process [45], this is equivalent to a marked Poisson process of total intensity $M\beta$ (where M is the number of possible types $M = |\mathcal{M}|$) in which the points have independent random marks, with equal probability $1/M$ for each possible type.

A conditional intensity function which depends only on the marks,

$$\lambda((u, m), \mathbf{y}) = \beta_m \tag{13}$$

where β_m, $m \in \mathcal{M}$ are constants, is a marked Poisson process of total intensity $\mu = \sum_m \beta_m$, in which the points have independent random marks, with probability $p_m = \beta_m / \mu$ for type m.

The most general multitype Poisson process has conditional intensity

$$\lambda((u, m), \mathbf{y}) = \beta_m(u) \tag{14}$$

where $\beta_m(u), m \in \mathcal{M}$ are arbitrary nonnegative integrable functions. This process has total intensity $\mu(u) = \sum_m \beta_m(u)$. The marks are independent but not identically distributed: a point at location u has conditional mark distribution $p_m(u) = \beta_m(u)/\mu(u)$ for $m \in \mathcal{M}$.

8.2 Multitype Models

Trend Component

In order to represent the dependence of the trend on the marks, the trend formula passed to `ppm` may involve the reserved name `marks`.

The trend formula `~1` states that the trend is constant and does not depend on the marks, as in (12). The formula `~marks` indicates that there is a separate, constant intensity for each possible mark, as in (13). If a uniform multitype Poisson process is to be fitted to data, the usual intention is to allow for different intensities for each mark, so the appropriate call would be

```
ppm(X, ~ marks, Poisson())
```

The result of fitting this model to the data in Fig. 3(b) yields the following output.

```
Stationary multitype Poisson process
Possible marks: off on
Intensity: Trend formula: ~marks
Fitted intensities:
beta_off  beta_on
88.68302 94.92830
```

This indicates that the fitted model is a multitype Poisson process with intensities 88.7 and 94.9 for the points of type “off” and “on” respectively.

In more elaborate cases, the trend formula may involve both the marks and the spatial locations or spatial covariates. For example the trend formula `~marks + polynom(x,y,2)` signifies that the first order trend is a log-quadratic function of the Cartesian coordinates, multiplied by a constant factor depending on the mark. The formulae

```
~ marks * polynom(x,2)
~ marks + marks:polynom(x,2)
```

both specify that, for each mark, the first order trend is a different log-quadratic function of the Cartesian coordinates. The second form looks “wrong” since it includes a “`marks` by `polynom`” interaction without having `polynom` in the model, but since `polynom` is a covariate rather than a factor this is is allowed, and makes perfectly good sense. As a result the two foregoing models are in fact mathematically equivalent. However, the fitted model objects will give slightly different output.

For example, the first model `~marks * polynom(x,2)` fitted to the data in Fig. 3(b) gives the following output (assuming `options("contrasts")` is set to its default, namely the 'treatment' contrasts):

```
Nonstationary multitype Poisson process
Trend formula: ~marks * polynom(x, 2)
Fitted coefficients for trend formula:
                (Intercept)                      markson
                  4.3127945                    0.2681231
           polynom(x, 2)[x]         polynom(x, 2)[x^2]
                  0.4651860                   -0.2363352
  markson:polynom(x, 2)[x] markson:polynom(x, 2)[x^2]
                 -0.6781045                    0.4023491
```

This form of the model gives two quadratic functions: a "baseline" quadratic

$$P_0(x, y) = 4.3127945 + 0.4651860x - 0.2363352x^2$$

and a quadratic associated with the mark level "`on`",

$$P_{\text{on}}(x, y) = 0.2681231 - 0.6781045x + 0.4023491x^2.$$

The baseline quadratic is the logarithm of the fitted trend for the points of type `off`, since `off` is the first level of the factor `marks`. For points of type `on`, since we are using the treatment contrasts, the log trend is

$$P_0(x, y) + P_{\text{on}}(x, y) = 4.580918 - 0.2129185x + 0.1660139x^2.$$

On the other hand, when the second model `~marks + marks:polynom(x,2))` is fitted to the same dataset, the output is

```
Nonstationary multitype Poisson process
Trend formula: ~marks + marks:polynom(x, 2)
Fitted coefficients for trend formula:
                (Intercept)                        markson
                  4.3127945                      0.2681231
  marksoff:polynom(x, 2)[x]     markson:polynom(x, 2)[x]
                  0.4651860                     -0.2129185
marksoff:polynom(x, 2)[x^2]  markson:polynom(x, 2)[x^2]
                 -0.2363352                      0.1660138
```

This says explicitly that the log trend for points of type `off` is

$$Q_{\text{off}}(x, y) = 4.3127945 + 0.4651860x - 0.2363352x^2$$

while for points of type `on` it is

$$Q_{\text{on}}(x, y) = 4.580918 - 0.2129185x + 0.1660139x^2.$$

Hence the two fitted models are mathematically identical.

Interaction Component

For the interaction component of the model, any of the interactions listed above for unmarked point processes may be used. However these interactions do not depend on the marks. We have additionally defined two interactions which do depend on the marks:

`MultiStrauss`	multitype Strauss process
`MultiStraussHard`	multitype Strauss/hard core

For the multitype Strauss process, a matrix of "interaction radii" must be specified. If there are m distinct levels of the marks, we require a matrix `r` in which `r[i,j]` is the interaction radius r_{ij} between types i and j. For the multitype Strauss/hard core model, a matrix of "hardcore radii" must be supplied as well. These matrices will be of dimension $m \times m$ and must be symmetric.

9 Irregular Parameters

As explained in Section 4.2, our model-fitting technique [5] estimates the parameters θ which appear in loglinear form (4) in the conditional intensity. We call these "regular" parameters, while other model parameters are called "irregular". Most of the familiar point process models have irregular parameters controlling the scale or range of interaction: an example is the interaction radius r of the Strauss process (3). Irregular parameters cannot be estimated directly using our algorithm, and must be given a fixed value in any call to `ppm`.

Very little theory is available about the estimation of irregular parameters. An exception is the case of hard-core radii. For example, consider the classical hard-core process, which is the special case of the Strauss process (3) with $\gamma = 0$. It can easily be shown [61] that the maximum likelihood and maximum pseudolikelihood estimate of r is

$$\widehat{r} = \min_i \min_{j \neq i} ||x_i - x_j||,$$

the minimum interpoint distance in the point pattern $\mathbf{x}$.

Some irregular parameters can be determined from the pair correlation function or the K function [34, 29, 28]. For the Strauss process with $\gamma < 1$, the pair correlation function has a jump at r. This leads to a useful procedure for estimating r called the 'cusp method' [34, 63], [65, p. 333]. These methods are not yet implemented in `spatstat`.

One general strategy available in `spatstat` for estimating irregular parameters is *profile pseudolikelihood* [5, Section 8.2]. Let θ and ψ denote the regular and irregular parameters respectively, and write the pseudolikelihood as $\mathrm{PL}(\theta, \psi; \mathbf{x})$. Define the profile pseudolikelihood for ψ to be

$$\mathrm{PPL}(\psi; \mathbf{x}) = \max_{\theta} \mathrm{PL}(\theta, \psi; \mathbf{x}), \tag{15}$$

the maximum value of pseudolikelihood obtained by maximising over the regular parameters with ψ held fixed. Then the maximum pseudolikelihood estimate of ψ is the value which maximises the profile pseudolikelihood,

$$\widehat{\psi} = \mathrm{argmax}_{\psi}\, \mathrm{PPL}(\psi; \mathbf{x}). \tag{16}$$

In spatstat the profile pseudolikelihood (15) can be evaluated for any given value of ψ by fitting the model with this value of ψ using ppm. The resulting fitted model object has a component named maxlogpl which gives the maximised log pseudolikelihood.

For example, the following code computes the maximum pseudolikelihood estimate of the interaction radius r in the Strauss process model for the cells dataset.

```
data(cells)
rval <- seq(0.01, 0.2, by=0.01)
prof <- numeric(length(rval))
for(i in seq(rval)) {
    fit <- ppm(cells, ~1, Strauss(r=rval[i]),
                      correction="translate")
    prof[i] <- fit$maxlogpl
}
iopt <- min(which(prof == max(prof)))
rhat <- rval[iopt]
```

Note that the same edge correction must be used to fit each model in order that the pseudolikelihood values be comparable. The example above shows the translation edge correction. If the border edge correction is used, the correction distance rbord should be fixed at the maximum interaction radius of all models to be fitted.

For diagnostic purposes the profile pseudolikelihood prof should be plotted against the r argument rval to verify that the function has a unique global maximum. In the example shown above, there is an unambiguous peak in profile likelihood at $\widehat{r} = 0.1$. Section 12.2 gives an example where more care is required.

10 Model Validation

Having fitted a point process model to data, it is important to 'validate' the model, i.e. to check formally or informally that the model is a good fit to the data, and that all terms in the model are appropriate [2, 16, 24, 72], [47, Chap. 12].

10.1 Residuals and Diagnostics

Residuals from the fitted model are an important diagnostic tool in other areas of applied statistics, but in spatial statistics they have only recently been developed [7, 46, 62]. The residuals and diagnostic plots introduced in [7] are available in `spatstat`. The function `diagnose.ppm` is the analogue of plotting the residuals against the covariates in a linear model, while `qqplot.ppm` is the analogue of a Q–Q plot of the residuals in a linear model.

These techniques are particularly well suited to detecting spatial inhomogeneity. For example, Fig. 7 (left) shows a point pattern simulated from the Poisson process with intensity $\lambda(x, y) = 300 \exp(-3|x - 0.5|)$ in the unit square. We then fitted the incorrect model, a *uniform* Poisson process, to these data. The right side of Fig. 7 shows the result of `diagnose.ppm` for this incorrect model. The striking deviations of the plots from their nominal (constant) values indicate clearly that the model is inappropriate, and suggest the form of departure from the model. For detailed information and examples, see [7] and the help files for these functions.

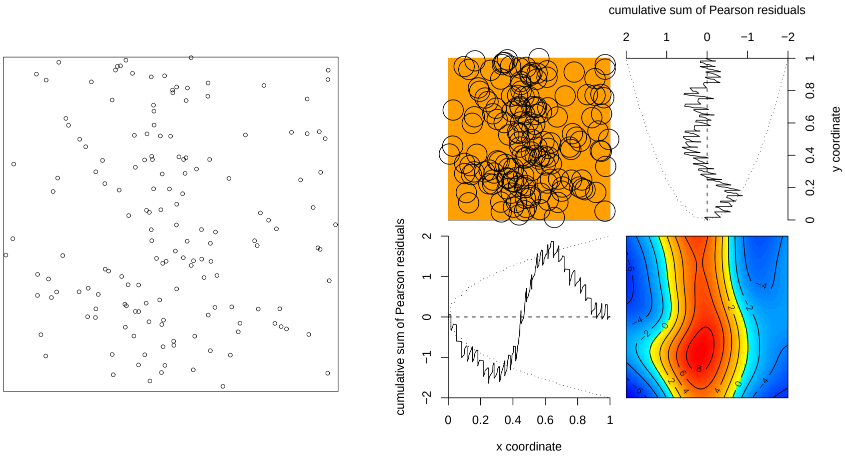


Fig. 7. Demonstration of diagnostic tools in `spatstat`. *Left:* data point pattern, a realisation of an inhomogeneous Poisson process. *Right:* diagnostic plots (generated by `diagnose.ppm`) for an incorrect model, a uniform Poisson process, fitted to the data.

10.2 Formal Inference

Techniques

Formal hypothesis tests are often applied in spatial statistics for the following purposes:

1. to test whether the point pattern is a realisation of a uniform Poisson process (complete spatial randomness or CSR);
2. to assess the goodness-of-fit of a point process model that has been fitted to the point pattern data;
3. to select models (e.g. to decide whether a particular term in a point process model may be omitted).

The first type of test is the most popular in applications, following Ripley [54, 56]. However, this test may often be inappropriate or uninformative: we usually know that the data are not completely random, and a formal confirmation of this statement is not scientifically informative since it does not indicate the kind of departure from complete randomness. Normally the second and third types of tests are more useful in modelling.

Statistical theory of parameter estimation and hypothesis testing for spatial point processes is rather limited. See the recent surveys of Møller and Waagepetersen [48, Chapters 8–10], Diggle [28, Chapter 2] and Van Lieshout [69, Chapter 3].

Techniques available for formal inference depend on the class of models envisaged. For **Poisson processes** (homogeneous or inhomogeneous), much of the classical theory of maximum likelihood is applicable, including the likelihood ratio test. Goodness-of-fit tests based on the χ^2 distribution are also possible after discretisation (binning) of the data. For **Cox processes**, estimation methods include minimum contrast [48, p. 182] and maximum likelihood in special cases [48, Section 10.3]. In the latter case, a likelihood ratio test is applicable. For **Gibbs processes**, we may use Monte Carlo maximum likelihood [48, Sect. 9.1.4] which provides *approximate* maximum likelihood estimates, confidence intervals and likelihood ratio tests. Bayesian inference has also also been developed [48, Sects. 9.3, 10.4]. For **Gibbs processes**, we may also use maximum pseudolikelihood [48, Sect. 9.2]. Maximum pseudolikelihood estimates are known to be consistent and asymptotically normal in some contexts [43, 42], but at the time of writing there is no statistical theory for hypothesis tests based on the pseudolikelihood. For **general point processes** which can be simulated, some elementary *simulation-based inference* is feasible [48, Chapter 8]. The canonical example is the Monte Carlo test [37, 14, 56]. Monte Carlo tests based on envelopes of simulations of the K function (and other summary functions) are very popular [28].

It is not known whether maximum likelihood estimation is optimally efficient [23, Sect. 8.5.8], [57]. Little is known about the power of the various tests mentioned above [28, p. 28]. The distributional information required for statistical inference can often be obtained only by using Monte Carlo methods. For example, simple formulae for the variance of estimators are available only for the uniform Poisson process [58, Sect. 3.3], but the Fisher information matrix for a general Gibbs process can be estimated by MCMC methods [32, 33], [69, page 103].

Implementation in spatstat

Spatstat includes the following support for formal inference.

For Poisson processes, since maximum pseudolikelihood is equivalent to maximum likelihood, models fitted using ppm can be compared using the likelihood ratio test. The function anova.ppm performs analysis of deviance and reports p-values for the likelihood ratio test.

For example, suppose we wish to test the null hypothesis that the point process is a homogeneous Poisson process, against the alternative that it is an inhomogeneous Poisson process with intensity of the form

$$\lambda(x, y) = \exp(a + bx) \; .$$

The likelihood ratio test is performed as follows, assuming X is the point pattern dataset.

```
fit0 <- ppm(X, ~1, Poisson())
fit1 <- ppm(X, ~x, Poisson())
anova(fit0,fit1)
```

For Gibbs processes, spatstat provides a simulation algorithm rmh, an implementation of the Metropolis-Hastings algorithm. This implementation will simulate a wide range of models, including models fitted to data by ppm. The algorithm handles arbitrary first-order trends, using a renormalisation technique [6, Sect. 10.4]. Trends may be specified as symbolic functions, as pixel images, or using a fitted model object. However, due to the high computational load, interpoint interaction terms in the conditional intensity are calculated in Fortran. This restricts the range of models that can be simulated. Currently the available interaction terms include Poisson, Strauss, Strauss/hard core, soft core, Geyer saturation process, multitype Strauss, multitype Strauss/hard-core, and the general stationary pairwise interaction with step-function potential.

Here are two examples of the use of rmh(); see the help file in spatstat for a plethora of other examples.

```
m <- list(cif="strauss",par=c(beta=2,gamma=0.2,r=0.7),
          w=c(0,10,0,10))
X1 <- rmh(model=m,start=list(n.start=80),
          control=list(nrep=5e6,nverb=1e5))
fit <- ppm(cells, ~1, Strauss(0.1))
X2 <- rmh(fit,start=list(n.start=200),
          control=list(nrep=1e5,nverb=5000))
```

The user may exploit rmh to perform simulation-based inference. Currently, inferential techniques must be implemented by hand: an example is given in Section 11.4. Future extensions of the package will include basic support for simulation-based inference.

Pseudolikelihood Ratio and Monte Carlo Tests

A reasonable substitute for the likelihood ratio test statistic for a general Gibbs process is based on the log *pseudolikelihood* ratio. Consider a null hypothesis H_0 and an alternative H_1, and suppose H_0 is contained in H_1. Denote the point pattern dataset by $\mathbf{x}$. Let $\widehat{\theta}_0 = \widehat{\theta}_0(\mathbf{x})$ be the estimate of the canonical parameters (by maximum pseudolikelihood) under H_0, and $\widehat{\theta}_1 = \widehat{\theta}_1(\mathbf{x})$ the estimate under H_1. The test statistic will be twice the log pseudolikelihood ratio

$$\Delta = \Delta(\mathbf{x}) = 2\left(\log \mathrm{PL}(\widehat{\theta}_1(\mathbf{x}); \mathbf{x}) - \log \mathrm{PL}(\widehat{\theta}_0(\mathbf{x}); \mathbf{x})\right) \tag{17}$$

analogous to the deviance in likelihood theory. The following simple function calculates the quantity Δ from two fitted model objects in spatstat.

```
delta <- function(model0, model1)
   2 * (model1$maxlogpl - model0$maxlogpl)
```

However we emphasise again that there is no statistical theory available to support inferential interpretations of Δ. We explore a distributional approximation for Δ in Sect. 11. Alternatively one may simply use Δ as the test statistic in a Monte Carlo test. Suppose for example that H_0 is a simple hypothesis (i.e. in which θ_0 is fixed). Generate m independent realisations $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}$ from the null hypothesis. Compute the corresponding values of the test statistic, say $\Delta_i = \Delta(\mathbf{x}^{(i)})$ for $i = 1, \ldots, m$. Compute the rank of Δ in the set of values $\{\Delta_1, \ldots, \Delta_m\} \cup \{\Delta\}$, that is, $R = 1 + \sum_{i=1}^{m} \mathbf{1}\{\Delta_i > \Delta\}$. Then under H_0, the rank R is uniformly distributed on $\{1, 2, \ldots, m+1\}$, assuming there are no ties. Hence, the test which rejects H_0 when $R \leq k$ has size $\alpha = k/(m+1)$ exactly, if H_0 is simple. The associated p-value is

$$p = \frac{R}{m+1}. \tag{18}$$

Gamma Approximation to Distribution of Pseudolikelihood Ratio

Another possibility is to approximate the null distribution of the log pseudolikelihood ratio statistic Δ by a Gamma distribution. The Gamma family is chosen simply because it is a flexible class of distributions, and because it includes the χ^2 distribution, which is the asymptotic null distribution of the likelihood ratio test statistic.

Given some realisations from the null distribution of Δ, we fit a Gamma distribution using the method of moments, then calculate a critical value or p-value for the observed Δ statistic based on this fitted Gamma distribution. The p-value so obtained will be called the “gamma p-value”, in contrast to the “Monte Carlo p-value” given by (18). The gamma approximation offers a substantial economy in the number of replicates used in the simulations. Of course this economy comes at the cost of placing trust in the approximation.

Some minimal experimentation indicates that the fit is generally good in the upper tail. See Fig. 8.

11 Harkness-Isham Ants' Nests Data

11.1 Description of Data

Figure 9 shows a point pattern data set recorded by Professor R.D. Harkness at a site in northern Greece, and described and analysed in [35]. The points record the locations of two species of ants: 68 nests of *Messor wasmanni* and 29 nests of *Cataglyphis bicolor*, in an irregular region 425 feet in diameter. Covariate information is also provided: the bold diagonal line in the Fig. 9 indicates a boundary between vegetation types, 'field' and 'scrub', while the two closely-spaced parallel lines delimit a foot track.

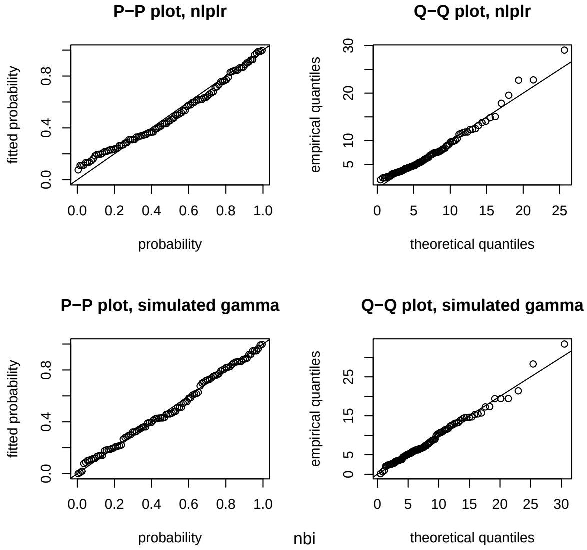


Fig. 8. Assessment of the gamma distribution approximation to the null distribution of the log pseudolikelihood ratio statistic Δ. Calculated for the test for between-species interaction described in Section 11. *Top:* P–P and Q–Q plots comparing the empirical null distribution of Δ (from simulations of H_0) with the gamma distribution. *Bottom:* analogous plots for i.i.d. random Gamma variates.

Interest in these data focuses on whether there is evidence of spatial inhibition between *Messor* nests, and of a tendency for *Cataglyphis* nests to be situated close to *Messor* nests. Harkness and Isham suggested that the two species have a relationship similar to that of predator and prey. *Messor* is a harvester which collects seeds for food and builds nests composed mainly of seed husks. *Cataglyphis* is a forager which eats dead insects and other arthropods, and, while not preying upon the *Messor* ants, feeds upon dead *Messor*s which have been killed by a predatory spider.

Rectangular subsets of the data were analysed in [35, 41, 67, 36, 5] and [60, Sect. 5.3]. Most of these analyses have used the dashed rectangles labelled

A and B in Fig. 9, which were defined by Harkness and Isham. The current analysis is, to our knowledge, the first to treat the full data set in its original (polygonal) window.

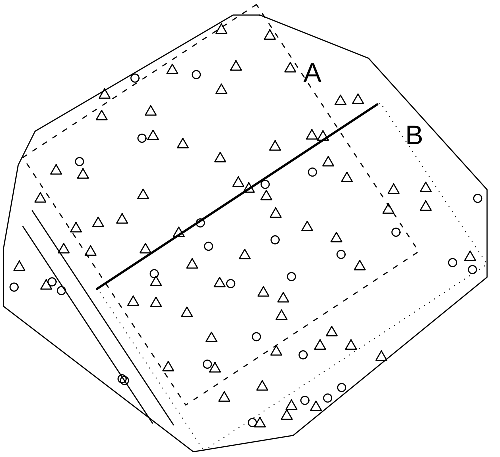


Fig. 9. Harkness-Isham ants' nests data [35, Fig. 1]. Locations of nests of two species of ants, *Messor wasmanni* (△) and *Cataglyphis bicolor* (○) in an irregular region 425 feet in diameter. Additional markings explained in the text. North at top of figure. Data reproduced by kind permission of Profs. R.D. Harkness and V. Isham.

The nest locations (in units of half-feet) were kindly provided by Professor V. Isham. The polygonal window and the extra features (foot track, field-scrub boundary, rectangles A and B) were digitized by the first author from Fig. 1 in [35]. The full dataset is now available in spatstat as `ants`. Dr A. Särkkä also kindly provided a version of the subset in rectangle *A* which was analysed in her work [60].

Harkness and Isham [35] concluded from their analysis of rectangular subsets *A* and *B* that there is spatial dependence in the location of the nests, both within and between species. Results for subsets *A* and *B* were similar, suggesting that the field-scrub boundary has no effect. Särkkä [60] concluded from an analysis of subset *A* that there was strong inhibition among *Cataglyphis* nests, but obtained conflicting conclusions (depending on the choice of technique) about any dependence between species.

11.2 Exploratory Analysis

Our analysis of the full dataset in spatstat begins with exploratory methods (Section 3.3). Kernel-smoothed intensity estimates for the nests of each species (see Fig. 10) are plotted by the code at the top of the next page.

```
smoothants <- lapply(split(ants), ksmooth.ppp)
plot(smoothants$Cataglyphis, main="Cataglyphis nests")
plot(smoothants$Messor, main="Messor nests")
```

The results suggest some slightly nonstationary trends in nest abundance. Messor nests are absent close to the track, and close to the eastern corner of the polygon. Cataglyphis nests are absent from the northeast border. We may also plot, for example, the ratio of these intensity estimates by

```
cata <- smoothants$Cataglyphis
mess <- smoothants$Messor
ratio <- im(cata$v/mess$v, cata$xcol, cata$yrow)
plot(ratio, main="Cataglyphis-to-Messor ratio")
```

A plot of this ratio appears roughly constant and suggests that the same slight inhomogeneity may affect both species equally.

The next step might be to assess within- and between-species interaction by computing exploratory summary functions. Spatstat provides multitype versions of the standard summary functions F, G, K, and J and the pair correlation function. In the notation of [71], for a stationary multitype point process, F_i denotes the empty space function F for the pattern consisting solely of points of type i, while $F_\bullet$ is the ordinary empty space function of the process of all points regardless of type. G_{ij} is the distribution function of the distance from a typical point of type i to the nearest point of type j, while $G_{i\bullet}$ is the distribution function of the distance from a typical point of type i to the nearest point regardless of type. Similarly K_{ij} is the K-function based on distances from points of type i to points of type j only, while $K_{i\bullet}$ is the K-function for distances from points of type i to points of any type. The pair correlation function ϱ_{ij} is defined by $\varrho_{ij}(t) = [(d/dt)K_{ij}(t)]/(2\pi t)$, analogously to the univariate case, and similarly for $\varrho_{i\bullet}$. Finally the J functions are defined [71] by $J_{ij}(t) = (1 - G_{ij}(t))/(1 - F_i(t))$ and $J_{i\bullet}(t) = (1 - G_{i\bullet}(t))/(1 - F_i(t))$. Diagnostic interpretation of these functions is described in [71, 65].

The spatstat function `alltypes` will compute these statistics and return an array of functions. For example

```
antsF <- alltypes(ants, "F")
plot(antsF)
```

computes the functions F_i (for i = Cataglyphis and i = Messor) and plots them. Similarly

```
antsG <- alltypes(ants, "G")
plot(antsG)
```

computes the functions G_{ij} for each i, j and plots them as a 2×2 array of panels, shown in Fig. 11. Similarly for the functions K_{ij} and J_{ij}. Algebraic transformations of these functions can be plotted easily using the R syntax for formulas.

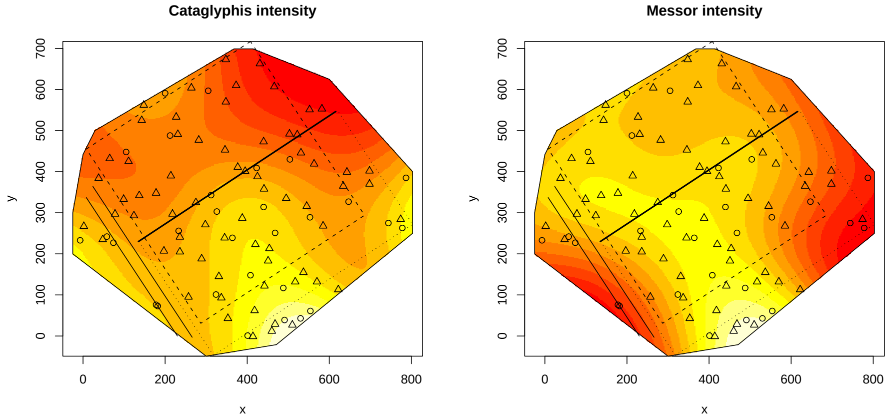


Fig. 10. Kernel smoothed intensity estimates for the two species of ants' nests

For example the corresponding L functions $L_{ij}(t) = \sqrt{K_{ij}(t)/\pi}$ may be plotted by

```
antsK <- alltypes(ants, "K")
plot(antsK, sqrt(trans/pi) ~ r)
```

The last line invokes `plot.fasp`. The second argument to `plot.fasp` is a model formula representing the variables which should be plotted. Here `trans` refers to the translation-correction estimate of K_{ij}. These plots all appear to evince some indication of between species attraction and of within species repulsion, at least over certain distance ranges. Plots based on the rectangular subset used by Särkkä are reasonably consistent in their appearance with those plots based on the full data set.

The pair correlation functions ϱ_{ij} are obtained from the K_{ij} estimates using `pcf`:

```
antsK <- alltypes(ants, "K")
antspcf <- pcf(antsK)
plot(antspcf)
```

This plot, shown in Fig. 12, tells a somewhat different story. It suggests that there is strong inhibition between *Messor* nests at all scales, while there is inhibition between *Cataglyphis* and *Messor* nests up to 10 half-feet and no interaction at longer distances. Between *Cataglyphis* nests there is a suggestion of short-scale inhibition and medium-scale attraction. For comparison we also show in Fig. 13 the pair correlation plot for rectangular subset A. This suggests inhibition for all combinations of nests.

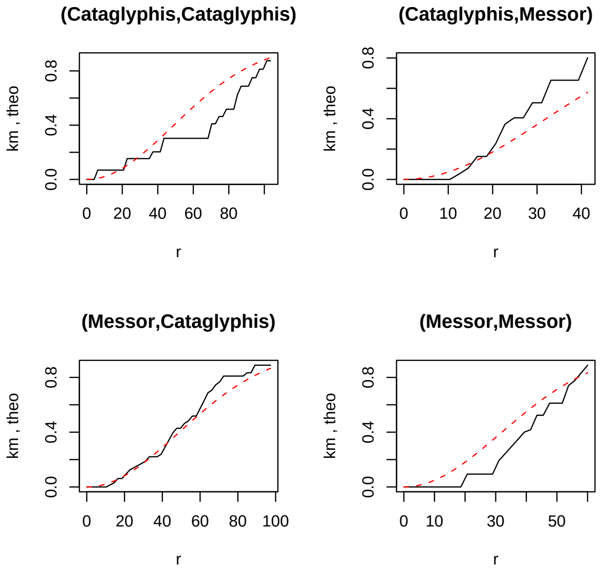


Fig. 11. Estimates of the cross-type nearest neighbour functions G_{ij} for the full ants' nests data.

11.3 Modelling

Takacs & Fiksel [67] and Särkkä [60, Sect. 5.3] fitted a Strauss/hard core model (amongst other models) to assess the evidence of within- and between-species dependence. Let $\mathbf{y} = \{(x_1, m_1), \dots, (x_n, m_n)\}$ denote a typical realisation of the process, where $x_i \in W$ are the locations and $m_i \in \mathcal{M}$ the corresponding marks (types). The Strauss/hard-core model has conditional intensity

$$\lambda((u,k),\mathbf{y}) = \beta_m \prod_i g(k, m_i, ||u - x_i||) \tag{19}$$

where

$$g(k,m,d) = \begin{cases} 0 & \text{if } d < h_{km} \\ \gamma_{mk} & \text{if } h_{km} \le d \le r_{km} \\ 1 & \text{if } d > r_{km} \end{cases}$$

Here $\beta_m > 0$ are parameters influencing the intensity of the process, and $\gamma_{km} > 0$ are interaction parameters similar to the Strauss interaction parameter γ and satisfying $\gamma_{mk} = \gamma_{km}$. The parameters $h_{km} > 0$ are 'hard-core distances' satisfying $h_{mk} = h_{km}$, while $r_{km} > 0$ are 'interaction distances' analogous to the Strauss interaction radius r, and satisfying $r_{mk} = r_{km}$ and $r_{km} > h_{km}$. The process is well-defined and integrable provided either that $h_{mm} > 0$ for all m, or that $\gamma_{km} \le 1$ for all k, m.

This model is chosen for its simplicity and flexibility in allowing for both negative and positive association within and between species. It is certainly a tentative model, and indeed pairwise interaction models such as the

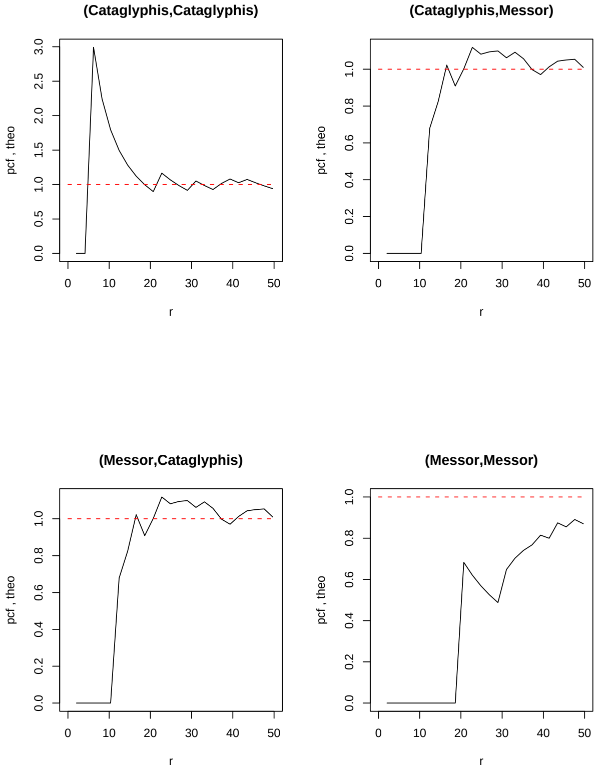


Fig. 12. Estimates of the cross-type pair correlation functions ϱ_{ij} for the full ants' nests data.

Strauss/hard core model are sometimes regarded as inadequate for describing clustering.

Denoting the two types Cataglyphis and Messor by `C` and `M` respectively, the model has 5 regular parameters (the intensities β_C, β_M and interaction parameters $\gamma_{CC}, \gamma_{MM}, \gamma_{CM}$) and 6 irregular parameters (the hard core distances h_{ij} and interaction distances r_{ij}).

To reduce the computational load we estimate the hard core distances by their maximum likelihood (and maximum pseudolikelihood) estimates, which are the corresponding minimum interpoint distances, obtained by

```
d <- pairdist(ants)
mks <- ants$marks
tapply(d, list(mks[row(d)], mks[col(d)]), min)
```

Note that if rounding is performed, then these values must be rounded downward, to ensure that the model still has nonzero likelihood. The resulting values are $\widehat{h}_{MM} = 18.7$, $\widehat{h}_{CC} = 4.9$ and $\widehat{h}_{CM} = 12.2$ (in half-feet). For the

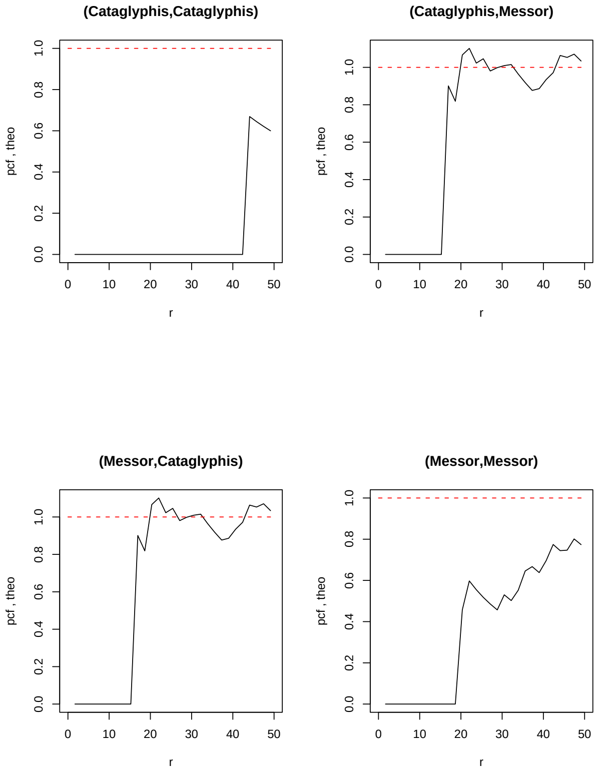


Fig. 13. Estimates of the pair correlation functions for Särkkä's version of the ants' nests data.

values of the Strauss interaction radii we adopted the same values as Takacs & Fiksel [67] and Särkkä [60], namely $r_{MM} = r_{CC} = r_{CM} = 90$ half-feet.

The model was fitted to the full dataset as follows.

```
rr   <- matrix(90,2,2)
hh   <- matrix(c(5.0,12.2,12.2,18.7),2,2)
types <- levels(ants$marks)
Int <- MultiStraussHard(types, rr, hh)
fit <- ppm(ants, ~marks, Int, correction="border", rbord=90)
```

Note that the trend formula must be `~marks` in order to allow different intensity values β_M, β_C for the two species.

Printing the fitted model object `fit` shows the fitted values of all parameters. It is necessary to select a value for the `correction` argument specifying the edge correction for the pseudolikelihood [5]. Here we fitted the model using the 'border' correction. Alternative choices of edge correction yield different fitted parameter values, as shown in the following table.

Edge correction	$\beta_C \times 10^4$	$\beta_M \times 10^4$	γ_{CC}	γ_{MM}	γ_{CM}
Border	0.92	8.69	0.50	0.68	1.12
Translation	0.72	2.35	0.75	0.90	1.11
None	0.71	2.33	0.72	0.88	1.13

Evidence for between-species dependence is quantified by the Strauss interaction parameter γ_{CM}, which is slightly greater than unity, suggesting moderate positive association. The estimates of within-species interaction γ_{CC}, γ_{MM} are below unity, suggesting inhibition.

These conclusions should be compared to those of Särkkä's [60] analysis of the subset of data in the rectangle labelled A in Fig. 9. She obtained $\hat{\gamma}_{CM} = 0.88$, which would indicate *inhibition* between species. For the sake of direct comparison, we also fitted the Strauss/hard core model to the data used by Dr Särkkä and kindly supplied by her. Several small differences can be observed between Särkkä's dataset and the subset of our data indicated by rectangle A. These may be attributed to slight differences in digitising Fig. 1 of [35]. In Särkkä's dataset the minimum interpoint distance between *Cataglyphis* and *Messor* nests is 11.2 half-feet rather than 12.2.

Changing the inter-species hard core distance to $h_{CM} = 11.2$, we fitted the Strauss-hard core model to Särkkä's version of the data in rectangle A. This also allows us to compare four different edge corrections for the pseudolikelihood [5] which are implemented for rectangular windows, namely the border, periodic and translation edge corrections, and Ripley's isotropic correction.

The choice of edge correction appeared to have a substantial impact upon the results. (As a matter of convenience Särkkä used a periodic edge correction in her analysis, and a stochastic approximation to the pseudolikelihood.) Our estimates of γ_{CM}, based upon Särkkä's data, are 1.37 (border correction), 0.99 (periodic edge correction), 1.20 (translation correction) and 1.00 (Ripley isotropic correction). These estimates are larger than Särkkä's value of 0.88, and two of them are larger than unity, consistent with between-species *attraction*.

We also fitted the same model (i.e. with smaller $h_{CM} = 11.2$) to the complete data set, in its polygonal window, resulting in γ_{CM} estimates of 1.33 (border correction) and 1.12 (translation correction). In this case both estimates are greater than 1, perhaps substantially greater. The evidence at this point is thus somewhat contradictory. The exploratory summary functions F, G, K, J suggest interspecies attraction, while the pair correlation function exhibits no sign of between-species interaction. Four of the six estimates of γ_{CM} are larger than unity, again suggesting interspecies attraction. Formal methods may be useful at this point.

11.4 Formal Inference

We conducted formal hypothesis tests for the presence of inter-species interaction using the methods described in Section 10.2. The null hypothesis of no inter-species interaction can be formulated as

$$H_0: \quad \gamma_{CM} = 1, \quad h_{CM} = 0$$

which implies that the nests of the two species are independent point processes of Strauss-hard core type.

The Strauss interaction radii were held fixed at 90 half-feet in all instances. Under the null hypothesis, the cross-species interaction radius r_{CM} is not identifiable, since it plays no role in the model when $\gamma_{CM} = 1$. Hence r_{CM} should not be estimated from simulations of H_0. The within-species interaction parameters γ_{CC}, γ_{MM} were held fixed to reduce computational load, but they could have been estimated instead.

Under the null model, there are no interaction terms between nests of different species. In spatstat, assigning a value of NA to an irregular parameter will cause the interpoint interaction term associated with this parameter to be omitted from the analysis. Thus our null model is represented by assigning NA values to every off-diagonal entry in the matrices of hard core distances and of Strauss interaction distances. The following code fits the null and alternative hypotheses to the data and evaluates the log pseudolikelihood ratio statistic Δ:

```
Str1 <- matrix(c(90, 90, 90, 90), 2,2)
Str0 <- matrix(c(90, NA, NA, 90), 2,2)
Hard1 <- matrix(c(5.0, 12.2, 12.2, 18.7), 2,2)
Hard0 <- matrix(c(5.0, NA,   NA,   18.7), 2,2)
Int0 <- MultiStraussHard(types, Str0, Hard0)
Int1 <- MultiStraussHard(types, Str1, Hard1)
fit0 <- ppm(ants, ~marks, Int0, correction="translate")
fit1 <- ppm(ants, ~marks, Int1, correction="translate")
dobs <- 2 * (fit1$maxlogpl - fit0$maxlogpl)
```

To generate 99 realisations from the null distribution of Δ we proceed as follows:

```
dvalues <- numeric(99)
for(i in 1:99) {
   Xsim <- rmh(fit0)
   hc1 <- nnd(Xsim)
   hc0 <- matrix(NA, 2, 2)
   diag(hc0) <- diag(hc1)
   Int0sim <- MultiStraussHard(types, Str0, hc0)
   Int1sim <- MultiStraussHard(types, Str1, hc1)
   fit0sim <- ppm(Xsim, ~marks, Int0sim,
                        correction="translate")
   fit1sim <- ppm(Xsim, ~marks, Int1sim,
                        correction="translate")
   dvalues[i] <-  2 * (fit1sim$maxlogpl - fit0sim$maxlogpl)
}
```

where nnd is a small function to compute the minimum nearest-neighbour distances between each pair of types:

```
nnd <- function(X) {
   mks <- X$marks
   d <- pairdist(X)
   tapply(d, list(mks[row(d)], mks[col(d)]), min)
}
```

The resulting Monte Carlo test for between-species interaction gave a p-value (18) of 0.18, based on 99 simulations from the null model as above. Using the gamma approximation (Sect. 10.2) based on a separate set of only 30 simulated realisations from the null model, the approximate p-value obtained was 0.1885. (Validity of this approximation is confirmed by Fig. 8 in Section 10.2.) Thus there appears to be no evidence of between-species interaction.

We then checked whether there was evidence of any interaction at all. In this case the null model simply consists of two independent Poisson processes, of different intensities. This is fitted by calling ppm with the trend given as ~marks and the interaction as Poisson. The alternative model was taken to be the full model, including both between and within species interactions. We obtained a Monte Carlo p-value of 0.03, and a gamma approximation p-value of 0.0402, thus providing evidence that some sort of interaction is present.

If we eliminate between-species interaction from the model, we can test for within-species interaction either for both species simultaneously, or in the context of univariate models fitted to each species separately. The p-values for the simultaneous test were 0.03 (Monte Carlo) and 0.0021 (gamma) indicating some evidence of within species interaction. The test based on univariate models gave p-values for the Messor ants of 0 (Monte Carlo) and 0.004 (gamma) and for the Cataglyphis ants of 0.64 (Monte Carlo) and 0.6279 (gamma), suggesting that there is within-species interaction among the Messor ants, but not among the Cataglyphis ants.

Finally, as a check on the absence of between species interaction, we performed a test in terms of a univariate model fitted to the Messor ants *conditional upon* the Cataglyphis ants. This model used the Strauss/hard core interaction as before, but added a trend term, the trend being a log-linear function of distance to the nearest Cataglyphis nest. The null model was formed simply by omitting the trend term. The empirical p-values for this test were 0.18 (Monte Carlo) and 0.1108 (gamma), which are again consistent with the hypothesis of no between-species interaction.

11.5 Incorporation of Covariates

In addition to recording the locations of the ants' nests, Harkness [35] noted a boundary between "field" and "scrub" crossing the middle of the study region, and a foot track running close to the perimeter. The relevance of these geographical features to the ants' nests pattern can easily be assessed using spatstat. Here we demonstrate the use of the modelling software to formulate and fit point process models which depend on covariates (Section 7).

A very simple model for “field/scrub” effect is one in which the intensity of the process is a different, constant value on each side of the field/scrub boundary. In more complicated models, the intensity might also depend on distance from the field/scrub boundary.

It is convenient to use the function `fsdistance(x,y)` which is displayed on page 38. This function computes the *signed* distance from any location (x, y) to the field/scrub boundary. A point `(x,y)` belongs to the field region if `fsdistance(x,y) > 0`.

The simplest sensible model, in which intensity of each species is a different, constant value on each side of the field/scrub boundary, can be fitted by including a covariate which is a two-level factor, indicating whether the point in question is in field or scrub. One way to do this is by means of the function `fsfac(x,y)` shown below.

```
fsdistance <- function(x,y) {
  ends <- ants.extra$fieldscrub
  para <- c(diff(ends$x),diff(ends$y))
  perp <- c(para[2], -para[1])
  unit <- perp/sqrt(sum(perp^2))
  cbind(x,y) %*% unit - (ends$x[1] * unit[1] +
                                    ends$y[1] * unit[2])
}

fsfac <- function(x,y) {
  factor(ifelse(fsdistance(x,y) > 0,  "field", "scrub"))
}
```

The desired model can then be fitted via:

```
ppm(ants, ~ marks * fsfac(x,y), Poisson())
```

Note carefully that the variable names `x` and `y` in the call to `ppm` above, are reserved names which refer to the Cartesian coordinates in the quadrature scheme. The code above exploits the fact that the chosen covariate can be expressed as a function of the Cartesian coordinates. If this is not true, then the covariates must be supplied either as pixel images or as columns in a data frame, as explained in Section 7.

The fitted model output (after rounding) is

```
Nonstationary multitype Poisson process
Trend formula: ~marks * fs(x, y)
Fitted coefficients
(Intercept)   marksMessor  fsfac(x,y)scrub
-9.35         0.52         -0.77
              marksMessor:fsfac(x,y)scrub
                           0.97
```

Since `field` is the first level of the factor `fsfac(x,y)`, this output indicates that the fitted intensities are as follows:

	Cataglyphis	*Messor*
FIELD	$\exp(-9.35)$ $= 0.9 \times 10^{-4}$	$\exp(-9.35 + 0.52)$ $= 1.5 \times 10^{-4}$
SCRUB	$\exp(-9.35 - 0.77)$ $= 0.4 \times 10^{-4}$	$\exp(-9.35 + 0.52 - 0.77 + 0.97)$ $= 1.7 \times 10^{-4}$

These values could also have been obtained by geometrically dividing the study region into two subregions and counting the numbers of nests of each species in each subregion. They show that *Cataglyphis* has a marked preference for the field region, while *Messor* nests have approximately equal intensity in field and scrub regions. This finding was reported by Harkness & Isham [35].

However, these differences are not significant according to the (asymptotic) likelihood ratio test for a field/scrub effect. Typing

```
fit1 <- ppm(ants, ~ marks * fsfac(x,y), Poisson())
fit0 <- ppm(ants, ~ marks,              Poisson())
anova(fit0, fit1, test="Chi")
```

yields a p-value of 0.12 (with reference to the χ^2_2 distribution).

Interpoint interaction may be incorporated, and probably should be incorporated, even into the simplest model. For example, we may fit

```
ppm(ants, ~ marks * fsfac(x,y), Int1)
```

where `Int1` is the interaction object representing the multitype Strauss/hard core model, constructed in the previous section using `MultiStraussHard`. The fitted intensity parameters β are as follows.

	Cataglyphis	*Messor*
FIELD	1.0×10^{-4}	2.0×10^{-4}
SCRUB	0.3×10^{-4}	2.0×10^{-4}

This strengthens the earlier suggestion that *Cataglyphis* nests have an affinity for field over scrub while *Messor* nests are indifferent.

Extending the model further, we might fit a trend (in either or both of the types) depending on the *distance* from the field/scrub boundary, as well as on the distinction between field and scrub. Assuming that the dependence on distance is loglinear, the model can be fitted by

```
fsdist <- function(x,y) { abs(fsdistance(x,y)) }
ppm(ants, ~ marks * fsdist(x,y) * fsfac(x,y), Int1)
```

This trend is essentially the simplest which can be fitted and which makes full use of all the variables of interest. It is admittedly arbitrary, but should have a reasonable chance of revealing a trend dependent upon the field/scrub dichotomy if such a trend exists.

We tested for a trend of the specified form, first in terms of a model allowing for both between and within species interactions, and then in terms of the model which appears most appropriate in the light of the tests previously conducted, namely a model in which there is interaction within the Messor species only. Testing for the trend yielded empirical p-values of 0.32 (Monte Carlo) and 0.3692 (gamma) when the full multivariate Strauss/hard core interaction was used, and of 0.50 (Monte Carlo) and 0.5709 (gamma) when the within-Messor-only interaction was used. Thus there is no evidence of a field/scrub effect, at least as described by a model of this form. Models involving smooth intensity functions can also be fitted in the same style.

12 The Queensland Copper Data

12.1 Data and Previous Analyses

The Queensland copper data, shown in Fig. 14, were introduced and analysed by Berman [11]. They consist of a point pattern of 67 copper ore deposits, and a line segment pattern of 146 geological features, called 'lineaments', obtained from an intensive geological survey of a 70 × 158 km region in central Queensland, Australia. It is of interest to find any association between the copper deposits and the lineaments. Since the lineaments are visible on satellite images, they might be used to guide the search for copper deposits, by predicting regions of high intensity for the copper points.

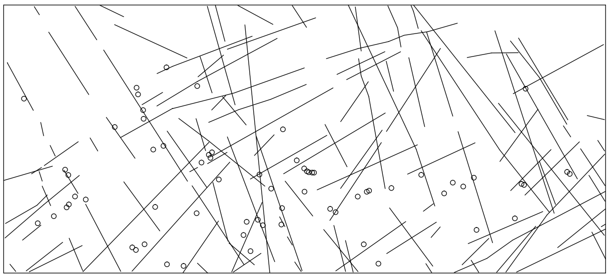


Fig. 14. Copper ore deposits (○) and lineaments (—) in a region of central Queensland. North at top of frame. Reproduced by kind permission of Dr A Green, Dr J Huntington, Dr M Berman and the Royal Statistical Society.

Berman [11] developed formal tests for dependence of the points upon the lineaments, based on measuring the distance from each point to the nearest lineament. The points are assumed to constitute an inhomogeneous Poisson process, with an intensity that depends on distance to the nearest lineament. The null hypothesis is that the intensity is constant.

Let X denote the copper point process and L the lineament process. All analysis will be performed conditionally upon L. In [11] it is assumed that X is conditionally Poisson given L, with intensity function of the form

$$\lambda_{X|L}(u) = \varrho(d(u, L)) \tag{20}$$

where $d(u, L)$ denotes the shortest distance from the point u to the nearest lineament, and ϱ is an unknown function. Under this assumption, the *observed* distances $d_i = d(x_i, L)$ for all points $x_i \in X$ are i.i.d. The null hypothesis, that ϱ is constant, corresponds to assuming a known distribution for the d_i (determined by the geometry of L) and hence can be tested using the Kolmogorov-Smirnov or other tests of goodness-of-fit. For details see [11].

For geological reasons, lineaments lying in different spatial orientations have typically been created at different epochs. Hence Berman [11] also considered the possibility that the intensity of the points depends only upon distance to lineaments lying in a particular subset of orientations. This subset consists of those lineaments having an angle (measured in the anticlockwise direction from the horizontal, with 0° pointing east) between 120° and 160°. He also considered the subset whose angles lie between 10° and 40°, but found the results from this latter set not to differ from the results for all lineaments.

Berman concluded that there is some evidence of dependence of the intensity of points upon the lineaments, when the entire window is considered, but speculated that this dependence might be a spurious artifact due to the scarcity of points in the northern half of the window. When he restricted attention to the southern half of the window (shown in Fig. 15) he found no evidence of association between points and lineaments.

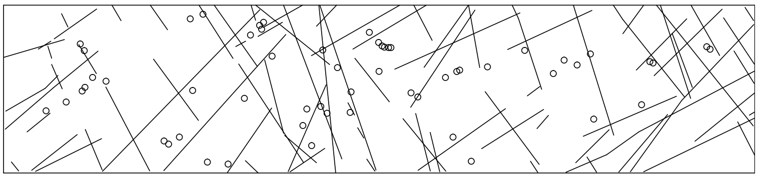


Fig. 15. Southern half window of the Queensland copper data.

The data in the southern half window were re-analysed in [7, 31]. Both analyses concluded that there is no evidence of dependence.

12.2 Analysis

In this work we re-visit these data, making use of the spatstat package. The convenient model-fitting and simulation facilities of spatstat make it easy to conduct tests of association between the points and lineaments, and to explore other aspects of the nature of these data. In particular we investigate the assumption that the points are conditionally Poisson. Attention is mainly restricted to the southern half window, but a further analysis of the entire window is discussed briefly.

We test for dependence of the points on the lineaments, using three simple parametric loglinear models for the intensity of the points given the lines:

$$\lambda(u) = \beta \exp\{\alpha d\} \tag{21}$$

$$\lambda(u) = \beta \exp\{\alpha_1 d + \alpha_2 d \cos\theta + \alpha_3 d \sin\theta\} \tag{22}$$

$$\lambda(u) = \exp\{\alpha_{0,i} + \alpha_{1,i} d\},\ i = 1, \ldots, 3\ . \tag{23}$$

These models are expressed in terms of $d = d(u, L)$, the distance from a point u to the nearest lineament, and $\theta = \theta(u, L)$, the spatial orientation of the lineament closest to u (measured as an anticlockwise angle from the east-pointing direction). Model (23) is obtained by dividing the angle θ into classes, with breakpoints determined by the lineament subsets investigated by Berman in [11]. These breakpoints are 10°, 40°, 120°,and 160°. There were no lineaments in the intervals $[0°, 10°]$ nor $[160°, 180°]$ so there are effectively three classes. The index $i = 1, \ldots, 3$ is determined by the class in which the angle θ falls.

Under the assumption that the points, given the lineaments, are a realization of an (inhomogeneous) Poisson processes we may apply the likelihood ratio test. The three models (and the null model comprising a constant intensity Poisson process) are fitted as follows. First we construct a data frame `Cov` containing the desired covariates: it has columns `d` (the distance to the nearest lineament), `angle` (the angle made by this nearest lineament with the horizontal) and `cat.ang` (the categorical variable or factor resulting from classifying `angle` into three groups).

```
data(copper)
attach(copper)
Q   <- quadscheme(SouthPoints,nd=c(24,106))
UQ  <- union.quad(Q)
Cov <- makecov(UQ, SouthLines)
```

The function `makecov` is a one-off utility which performs the analytic geometry of computing distances between points and line segments. The implementation of such calculations will change shortly. Interested readers should contact the authors for further information.

The three models (21)–(23) can then be fitted, along with the null model, as follows

```
F0  <- ppm(Q)
F1  <- ppm(Q, ~d, covariates=Cov)
F2  <- ppm(Q, ~d + I(d * sin(angle)) + I(d * cos(angle)),
           covariates=Cov)
F3  <- ppm(Q, ~d * cat.ang, covariates=Cov)
```

These models are Poisson by default. Note that expressions like `d*sin(angle)` must be protected by `I()` within a call to `ppm()` to ensure that `*` is interpreted as multiplication.

The likelihood ratio test for each successive pair of models can now be performed using `anova.ppm`, or by hand as indicated in Sect. 10. The resulting p-values are as follows.

Model	Statistic	p-value
1	0.5613	0.4538
2	6.0080	0.1112
3	3.7561	0.5850

Another subdivision of the lineament orientations with breakpoints of 0°, 60°, 120° and 180°, resulted in a likelihood ratio statistic of 7.7873 with p-value 0.1684.

There is thus no evidence of dependence of the points upon the lineaments, at least in the forms of these models. Thus our conclusions here are in agreement with the previous analyses [7, 11, 31].

However, the foregoing analyses assume that the copper points form an (inhomogeneous) Poisson process given the lineaments. This assumption should be validated. One possibility is to use the residual plots described in Section 10.1; an analysis of these data is reported in [7].

Alternatively we may use the inhomogeneous version of the K-function, $K_{\mathrm{inhom}}(r)$ introduced in [3]. This requires the intensity function of the copper point process, evaluated at the data points. We estimated the intensity function in four ways: from the three foregoing parametric models, and also non-parametrically.

The parametric estimates are straightforward. The fitted intensity at the data points is provided by `fitted.ppm`. Thus for example

```
lambda <- fitted(F1)
K1 <- Kinhom(SouthPoints, lambda)
plot(K1)
```

computes the inhomogeneous K-function based on the intensity function λ estimated under the model (21), and plots the result.

For the non-parametric estimate of the intensity function, we assume (20) holds, where the form of the function ϱ is not specified. We make use of the following relationship [31]. Suppose X and L are jointly stationary. Let F_L be the empty space function for the L process, that is, $F_L(t)$ is the cumulative distribution function of the distance from an arbitrary point in the plane to the nearest lineament in L. Let G_{XL} be the cumulative distribution function of the distance from a typical point of the process X to the nearest line segment in L. Then if (20) holds, we have

$$G_{XL}(t) = \frac{\int_0^t \varrho(s) dF_L(s)}{\int_0^\infty \varrho(s) dF_L(s)} \tag{24}$$

From this it follows that

$$\varrho(t) = \mu \frac{dG_{XL}(t)/dt}{dF_L(t)/dt} \tag{25}$$

where

$$\mu = \int_0^\infty \varrho(s) dF_L(s) = \mathbb{E}(\varrho(d(u, L))$$

is the intensity of X. The moment estimator of μ is $\hat{\mu} = n(X)/|W|$.

To estimate $dG_{XL}(t)/dt$ and $dF(t)/dt$ we can compute empirical estimates of F and G, fit smoothing splines, and take the derivatives of the splines. Estimates of F and G can be computed using standard methods. At the time of writing, these methods must be implemented by hand for line segment patterns. Future extensions of spatstat will include support for these calculations. The graph of $\widehat{\varrho}(t)$ is shown in Fig. 16.

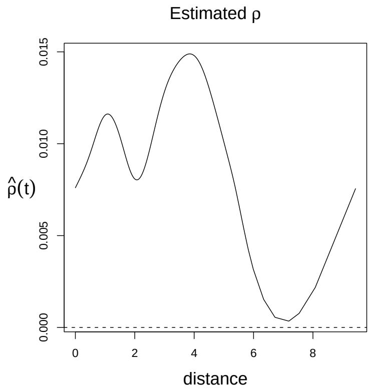


Fig. 16. Estimate of the function $\varrho(t)$ in (20) obtained using by substituting spline estimates in (25).

There were very substantial differences in the appearance of the intensity surfaces computed by different parametric methods, and by the nonparametric methods. Despite this, the four estimates of the inhomogeneous K function turned out to be virtually identical to each other, and to the estimated conventional K function, for each window. Plots of one of the inhomogeneous K function estimates and of the conventional K function estimate are shown in Fig. 17. The explanation is that the inhomogeneous K function depends only on the estimated intensity values at the points of the point pattern. These intensity values were approximately constant for these data.

The K functions suggest that there is positive association between the copper deposits, conditioned on the lineaments. Again, we would like to be able

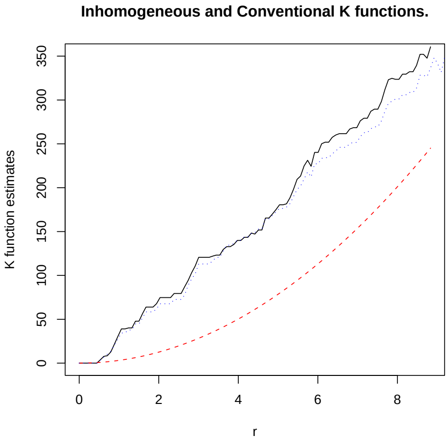


Fig. 17. Inhomogeneous K function (solid line), conventional K function (dotted line) and theoretical K function under CSR (dashed line) for the Queensland copper data. Inhomogeneous version computed using parametric model (22).

to test this formally, and this requires a class of models which allow for positive association or clustering. One convenient choice is the Geyer saturation model [32].

In order to fit the Geyer model we need to estimate the "irregular" parameters of the model, namely the interaction radius $r > 0$ and the saturation number s. Rough estimates may be found by searching over a small set of integer values for s (1 to 5 inclusive) using profile pseudolikelihood. Note that the maximum over r of the log pseudolikelihood (for a fixed value of s) must occur at one of the interpoint distances of the observed pattern.

The values obtained for the estimates of the irregular parameters were $\widehat{r} = 1.18$ and $\widehat{s} = 2$ respectively. Similar estimates ($\widehat{r} = 1.05$ and $\widehat{s} = 2$) were obtained when we also included in the model a trend of the form (22) along with the Geyer interaction. Sample plots of the profiles, for the trend-included setting, are shown in Fig. 18. The profile over r for $s = 5$ is very similar to that for $s = 4$ and is omitted to save space.

The profile log pseudolikelihood in Fig. 18 is shown only for $r \leq 10$ km. Localized sharp peaks occur for some larger values of r, and in fact the overall unconstrained maximum occurs at $r = 10.62$ and $s = 1$. However, this value of r is not credible. The associated estimate of γ is 94.272, which would cause immensely strong clustering. Plots of the estimated G and K functions and the pair correlation function suggest an interaction range between 1 and 3 km. It seems plausible that the value of $r = 10.62$ is a numerical artifact, since it is just slightly larger than the *maximum* nearest neighbour distance in the data, and the observation window has a width of only 35 km. We therefore decided to dismiss the profile peaks for $r > 10$ as anomalies. This example illustrates the delicacy of estimating irregular parameters and the need to check the results of a maximisation algorithm. For interaction radii it is probably sensible to restrict the search range to the interval from 0 to the maximum nearest neighbour distance.

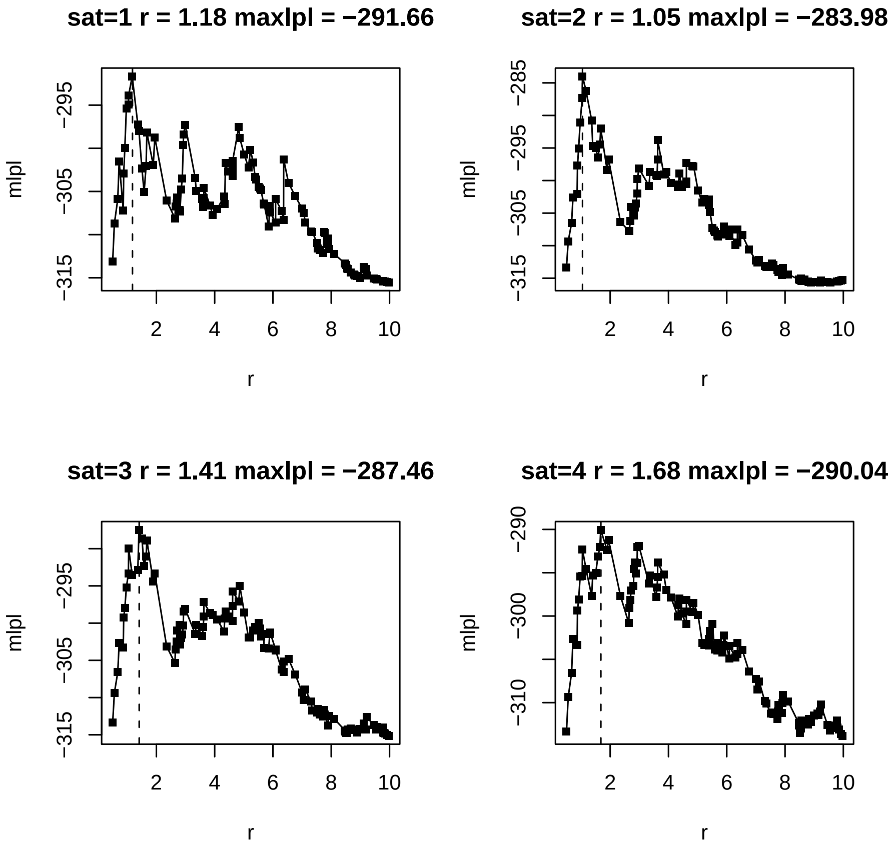


Fig. 18. Profile log pseudolikelihood for a Geyer model, as a function of interaction radius r, for several values of the saturation parameter s. Southern half window. Trend from model (22) included in the fit.

We tested the model with trend given by model (22) and interaction given by `Geyer(1.05,2)` against the null model with trend only. We also tested a stationary model with `Geyer(1.18,2)` interaction against a completely null (i.e. constant intensity Poisson) model.

In the first case we obtained a log pseudolikelihood ratio statistic $\Delta =$ 57.35, a Monte Carlo p-value of 0.01, and a gamma approximation p-value of

4.6×10^{-6}. In the second case $\varDelta = 59.36$, with Monte Carlo p-value 0.01, and gamma p-value of 4.8×10^{-11}.

These tests appear to confirm the impression given by the K function plots that there is positive association between the points. Given that there is attraction, the tests for dependence of the points upon the lineaments, based upon inhomogeneous Poisson models, cannot be considered valid. However we can now conduct tests allowing for the apparent interaction. The test of trend (given by model (22)) plus `Geyer(1.05,2)` against Geyer interaction only gave $\varDelta = 4.75$, with a Monte Carlo p-value of 0.12, and a gamma p-value of 0.144. Thus when interaction is allowed for, the evidence of dependence of the points upon the lines is still "insignificant" and is in fact slightly weaker than if we assume the points to arise from an (inhomogeneous) Poisson process.

12.3 North-South Effect

We now briefly consider the complete data set rather than the southern half window. In particular we focus on Berman's conjecture that the apparent dependence of points on lineaments, when the entire window is considered, might be a spurious artifact due to the scarcity of points in the northern half of the window. If this is indeed the case, then it may be possible to adjust for the low intensity in the northern half window by introducing a trend depending upon the spatial covariates x and y.

One convenient class of models uses the smoothing term `s` in the trend formula. For example

```
Q  <- quadscheme(copper$points, nd=c(34,75))
F0 <- ppm(Q,~s(y),use.gam=TRUE)
```

fits a Poisson model with a smooth trend in the y coordinate (Northing) only. We may test this null model against more elaborate models such as

```
UQ  <- union.quad(Q)
Cov <- makecov(UQ, copper$lines)
F1  <- ppm(Q, ~s(y) + d + I(d * sin(angle)) +
           I(d * cos(angle)), covariates=Cov, use.gam=TRUE)
F2  <- ppm(Q, ~s(y) + d * cat.ang, covariates=Cov,
           use.gam=TRUE)
```

The likelihood ratio test of models `F1` and `F2` against model `F0` turned out to have values of 18.22 and 9.96 on 3 and 9 degrees of freedom respectively. (Note that for the full window, all five angle categories are non-empty.) The corresponding p-values are 0.0003 and 0.3537. Thus there appears to remain an indication of dependence of the points on the lineaments via model (22) (although not via model (23)) for the full data set, even after a spatial trend (depending on the x and y coordinates) is allowed for.

The foregoing likelihood ratio test may be criticised since we had already demonstrated an interpoint interaction in the southern half window. Instead we should be conducting Monte Carlo tests involving an interpoint interaction term. The Geyer model irregular parameters may be estimated for the full window by profiling, yielding $\widehat{r} = 1.18$ and $\widehat{s} = 2$ (as for the southern half window when no trend is included. The estimates of the irregular parameters were the same for the full window whether a lineaments-dependent trend was allowed for or not.) We might thus set out to test the null model

```
F0 <- ppm(Q, ~s(y), Geyer(1.18,2), covariates=Cov,
          use.gam=TRUE)
```

against (for instance)

```
F1 <- ppm(Q,~ s(y) + d * cat.ang,  Geyer(1.18,2),
          covariates=Cov, use.gam=TRUE)
```

Notice that here, as elsewhere, we propose to conduct inference *conditionally* on the fitted values of the irregular parameters. This is done mainly to save computational time. A definitive formal analysis should also look at the effect of estimating the irregular parameters.

References

1. L. Anselin. Local indicators of spatial association – LISA. *Geographical Analysis*, 27:93–115, 1995.
2. A.C. Atkinson. *Plots, Transformations and Regression.* Number 1 in Oxford Statistical Science Series. Oxford University Press/ Clarendon, 1985.
3. A. Baddeley, J. Møller, and R. Waagepetersen. Non- and semiparametric estimation of interaction in inhomogeneous point patterns. *Statistica Neerlandica*, 54(3):329–350, November 2000.
4. Adrian Baddeley and Rolf Turner. spatstat: an R package for analyzing spatial point patterns. *Journal of Statistical Software*, 12(6):1–42, 2005. URL: http://www.jstatsoft.org.
5. A. Baddeley and R. Turner. Practical maximum pseudolikelihood for spatial point patterns (with discussion). *Australian and New Zealand Journal of Statistics*, 42(3):283–322, 2000.
6. A. Baddeley and R. Turner. Spatstat: an R package for analyzing spatial point patterns. Research Report 2004/13, School of Mathematics and Statistics, University of Western Australia, September 2004.
7. A. Baddeley, R. Turner, J. Møller, and M. Hazelton. Residual analysis for spatial point processes. Provisionally accepted for publication as a read paper in the *Journal of the Royal Statistical Society (series B).*
8. A.J. Baddeley, M.N.M. van Lieshout, and J. Møller. Markov properties of cluster processes. *Advances in Applied Probability*, 28:346–355, 1996.
9. M.S. Bartlett. The spectral analysis of two-dimensional point processes. *Biometrika*, 51:299–311, 1964.
10. M.S. Bartlett. *The statistical analysis of spatial pattern.* Chapman and Hall, London, 1975.
11. M. Berman. Testing for spatial association between a point process and another stochastic process. *Applied Statistics*, 35:54–62, 1986.

12. M. Berman and T.R. Turner. Approximating point process likelihoods with GLIM. *Applied Statistics*, 41:31–38, 1992.
13. J. Besag. Statistical analysis of non-lattice data. *The Statistician*, 24:179–195, 1975.
14. J. Besag and P.J. Diggle. Simple Monte Carlo tests for spatial pattern. *Applied Statistics*, 26:327–333, 1977.
15. C. Chatfield. *Problem solving: a statistician's guide*. Chapman and Hall, 1988.
16. D. Collett. *Modelling Binary Data*. Chapman and Hall, London, 1991.
17. D. R. Cox and V. Isham. *Point processes*. Chapman and Hall, London, 1980.
18. D.R. Cox and E.J. Snell. *Applied Statistics: principles and examples*. Chapman and Hall, 1981.
19. The Comprehensive R Archive Network.
    URL `http://www.cran.r-project.org`.
20. N. Cressie and L.B. Collins. Analysis of spatial point patterns using bundles of product density LISA functions. *Journal of Agricultural, Biological and Environmental Statistics*, 6:118–135, 2001.
21. N. Cressie and L.B. Collins. Patterns in spatial point locations: local indicators of spatial association in a minefield with clutter. *Naval Research Logistics*, 48:333–347, 2001.
22. N.A.C. Cressie. *Statistics for Spatial Data*. John Wiley and Sons, New York, 1991.
23. N.A.C. Cressie. *Statistics for Spatial Data*. John Wiley and Sons, New York, 1993. Revised edition.
24. A.C. Davison and E.J. Snell. Residuals and diagnostics. In D.V. Hinkley, N. Reid, and E.J. Snell, editors, *Statistical theory and modelling (in honour of Sir David Cox FRS)*, chapter 4, pages 83–106. Chapman and Hall, London, 1991.
25. P. J. Diggle. A kernel method for smoothing point process data. *Journal of the Royal Statistical Society, series C (Applied Statistics)*, 34:138–147, 1985.
26. P.J. Diggle. *Statistical analysis of spatial point patterns*. Academic Press, London, 1983.
27. P.J. Diggle. A point process modelling approach to raised incidence of a rare phenomenon in the vicinity of a prespecified point. *Journal of the Royal Statistical Society, series A*, 153:349–362, 1990.
28. P.J. Diggle. *Statistical Analysis of Spatial Point Patterns*. Arnold, second edition, 2003.
29. P.J. Diggle, D.J. Gates, and A. Stibbard. A nonparametric estimator for pairwise-interaction point processes. *Biometrika*, 74:763–770, 1987.
30. F. Divino, A. Frigessi, and P.J. Green. Penalised pseudolikelihood estimation in Markov random field models. *Scandinavian Journal of Statistics*, 27(3):445–458, 2000.
31. R. Foxall and A. Baddeley. Nonparametric measures of association between a spatial point process and a random set, with geological applications. *Applied Statistics*, 51(2):165–182, 2002.
32. C.J. Geyer. Likelihood inference for spatial point processes. In O.E. Barndorff-Nielsen, W.S. Kendall, and M.N.M. van Lieshout, editors, *Stochastic Geometry: Likelihood and Computation*, number 80 in Monographs on Statistics and Applied Probability, chapter 3, pages 79–140. Chapman and Hall / CRC, Boca Raton, Florida, 1999.

33. C.J. Geyer and J. Møller. Simulation procedures and likelihood inference for spatial point processes. *Scandinavian Journal of Statistics*, 21(4):359–373, 1994.
34. K.-H. Hanisch and D. Stoyan. Remarks on statistical inference and prediction for a hard-core clustering model. *Statistics*, 14:559–567, 1983.
35. R.D. Harkness and V. Isham. A bivariate spatial point pattern of ants' nests. *Applied Statistics*, 32:293–303, 1983.
36. H. Högmander and A. Särkkä. Multitype spatial point patterns with hierarchical interactions. *Biometrics*, 55:1051–1058, 1999.
37. A.C.A. Hope. A simplified Monte Carlo significance test procedure. *Journal of the Royal Statistical Society, series B*, 30:582–598, 1968.
38. K. Hornik. The R FAQ: Frequently asked questions on R. URL `http://www.ci.tuwien.ac.at/~hornik/R/`. ISBN 3-901167-51-X.
39. F. Huang and Y. Ogata. Improvements of the maximum pseudo-likelihood estimators in various spatial statistical models. *Journal of Computational and Graphical Statistics*, 8(3):510–530, 1999.
40. Ross Ihaka and Robert Gentleman. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3):299–314, 1996.
41. V.S. Isham. Multitype Markov point processes: some approximations. *Proceedings of the Royal Society of London, Series A*, 391:39–53, 1984.
42. J.L. Jensen and H.R. Künsch. On asymptotic normality of pseudo likelihood estimates for pairwise interaction processes. *Annals of the Institute of Statistical Mathematics*, 46:475–486, 1994.
43. J.L. Jensen and J. Møller. Pseudolikelihood for exponential family models of spatial point processes. *Annals of Applied Probability*, 1:445–461, 1991.
44. F.P. Kelly and B.D. Ripley. On Strauss's model for clustering. *Biometrika*, 63:357–360, 1976.
45. J.F.C. Kingman. *Poisson Processes.* Oxford University Press, 1993.
46. A.B. Lawson. A deviance residual for heterogeneous spatial Poisson processes. *Biometrics*, 49:889–897, 1993.
47. P. McCullagh and J.A. Nelder. *Generalized Linear Models.* Chapman and Hall, second edition, 1989.
48. J. Møller and R.P. Waagepetersen. *Statistical Inference and Simulation for Spatial Point Processes.* Chapman and Hall/CRC, Boca Raton, 2003.
49. Y. Ogata and M. Tanemura. Estimation of interaction potentials of spatial point patterns through the maximum likelihood procedure. *Annals of the Institute of Statistical Mathematics*, B 33:315–338, 1981.
50. Y. Ogata and M. Tanemura. Likelihood analysis of spatial point patterns. *Journal of the Royal Statistical Society, series B*, 46:496–518, 1984.
51. Y. Ogata and M. Tanemura. Likelihood estimation of interaction potentials and external fields of inhomogeneous spatial point patterns. In I.S. Francis, B.J.F. Manly, and F.C. Lam, editors, *Pacific Statistical Congress*, pages 150–154. Elsevier, 1986.
52. A. Penttinen. *Modelling Interaction in Spatial Point Patterns: Parameter Estimation by the Maximum Likelihood Method.* Number 7 in Jyväskylä Studies in Computer Science, Economics and Statistics. University of Jyväskylä, 1984.
53. R Development Core Team. *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria, 2004. ISBN 3-900051-00-3.
54. B.D. Ripley. The second-order analysis of stationary point processes. *Journal of Applied Probability*, 13:255–266, 1976.

55. B.D. Ripley. Modelling spatial patterns (with discussion). *Journal of the Royal Statistical Society, Series B*, 39:172–212, 1977.
56. B.D. Ripley. *Spatial Statistics.* John Wiley and Sons, New York, 1981.
57. B.D. Ripley. Spatial statistics: developments 1980–3. *International Statistical Review*, 52:141–150, 1984.
58. B.D. Ripley. *Statistical Inference for Spatial Processes.* Cambridge University Press, 1988.
59. B.D. Ripley. Gibbsian interaction models. In D.A. Griffiths, editor, *Spatial Statistics: Past, Present and Future*, pages 1–19. Image, New York, 1989.
60. A. Särkkä. *Pseudo-likelihood approach for pair potential estimation of Gibbs processes.* Number 22 in Jyväskylä Studies in Computer Science, Economics and Statistics. University of Jyväskylä, 1993.
61. B. W. Silverman and T. C. Brown. Short distances, flat triangles and poisson limits. *Journal of Applied Probability*, 15:815–825, 1978.
62. D. Stoyan and P. Grabarnik. Second-order characteristics for stochastic structures connected with Gibbs point processes. *Mathematische Nachrichten*, 151:95–100, 1991.
63. D. Stoyan and P. Grabarnik. Statistics for the stationary Strauss model by the cusp point method. *Statistics*, 22:283–289, 1991.
64. D. Stoyan, W.S. Kendall, and J. Mecke. *Stochastic Geometry and its Applications.* John Wiley and Sons, Chichester, second edition, 1995.
65. D. Stoyan and H. Stoyan. *Fractals, random shapes and point fields.* Wiley, 1995.
66. D. Stoyan and H. Stoyan. Non-homogeneous Gibbs process models for forestry — a case study. *Biometrical Journal*, 40:521–531, 1998.
67. R. Takacs and T. Fiksel. Interaction pair-potentials for a system of ants' nests. *Biometrical Journal*, 28:1007–1013, 1986.
68. J. Tukey. *Exploratory Data Analysis.* Addison-Wesley, Reading, Mass., 1977.
69. M.N.M. van Lieshout. *Markov Point Processes and their Applications.* Imperial College Press, 2000.
70. M.N.M. van Lieshout and A.J. Baddeley. A nonparametric measure of spatial interaction in point patterns. *Statistica Neerlandica*, 50:344–361, 1996.
71. M.N.M. van Lieshout and A.J. Baddeley. Indices of dependence between types in multivariate point patterns. *Scandinavian Journal of Statistics*, 26:511–532, 1999.
72. W.N. Venables and B.D. Ripley. *Modern Applied Statistics with S-Plus.* Springer, second edition, 1997.