Spatial point processes: Theory and practice illustrated with R

Ege Rubak

Department of Mathematical Sciences
Aalborg University

Lecture III, February 24, 2011

About the lecture series (again)

- First, I would like to thank the many people who showed up for lectures I-II last week (if any of them are here today).
- Please remember that Adrian Baddeley should be credited for most of the material presented in the slides.
- To illustrate the usage of R and `spatstat` commands are printed in this style on the slides:

```
> library(spatstat)
```

Contents of lecture III

- Investigating dependence between points using functional summary statistics.
- Simple non-Poisson models.
- Gibbs models.

Fundamentals of spatstat

- The main objects we use in spatstat are: an observation window of class owin, a planar point pattern of class ppp (which contains a window), a tesselation of class tess and an pixel image of class im.
- If have non-ppp data where we know the x and y coordinates and the window of observation we can create the ppp-object (here we choose the coordinates at random and use the unit square):

```
> W <- square(1)
> x <- runif(10)
> y <- runif(10)
> X <- ppp(x, y, win = W)
```

- To convert more complicated data to the ppp-format the package sp can be used in conjunction with other packages as e.g. maptools, shapefiles and rgdal.

Functional summary statistics

Recall that the some of the classical methods to investigate interpoint interaction are through functional summary statistics based on distances such as:

- **pairwise distances** $s_{ij} = ||x_i - x_j||$ between all distinct pairs of points x_i and x_j $(i \neq j)$ in the pattern;
- **nearest neighbour distances** $t_i = \min_{j \neq i} s_{ij}$, the distance from each point x_i to its nearest neighbour;
- **empty space distances** $d(u) = \min_i ||u - x_i||$, the distance from a fixed reference location u in the window to the nearest data point.

Last lecture we specifically looked at the empty space function:

$$F(r) = \mathbb{P}\{d(u, \mathbf{X}) \leq r\} \tag{1}$$

Nearest neighbour distances

Assuming the point process $\mathbf{X}$ is stationary, we can define the cumulative distribution function of the nearest-neighbour distance for a typical point in the pattern,

$$G(r) = \mathbb{P}\{d(u, \mathbf{X} \setminus \{u\}) \leq r \mid u \in \mathbf{X}\} \tag{2}$$

where u is an arbitrary location, and $d(u, \mathbf{X} \setminus \{u\})$ is the shortest distance from u to the point pattern $\mathbf{X}$ excluding u itself. If the process is stationary then this definition does not depend on u.

Nearest neighbour distances

The empirical distribution function of the observed nearest-neighbour distances

$$G^*(r) = \frac{1}{n(\mathbf{x})} \sum_i \mathbf{1}\{d(x_i, \mathbf{X} \setminus \{x_i\}) \leq r\} \tag{3}$$

is a negatively biased estimator of $G(r)$. Typically, a weighted version of (3) is used.
For a homogeneous Poisson point process of intensity λ, the nearest-neighbour distance distribution function is known to be

$$G_p(r) = 1 - \exp(-\lambda \pi r^2). \tag{4}$$

This is identical to the empty space function for the Poisson process.

Pairwise distances

Ripley's K-function is defined for a stationary point process so that $\lambda K(r)$ is the expected number of other points of the process within a distance r of a typical point of the process. Formally

$$K(r) = \frac{1}{\lambda}\mathbb{E}\left[N(\mathbf{X} \cap b(u,r) \setminus \{u\}) \mid u \in \mathbf{X}\right], \tag{5}$$

and for a homogeneous Poisson process

$$K_p(r) = \pi r^2. \tag{6}$$

Most estimators are weighted and renormalised empirical distribution functions of the pairwise distances, of the general form

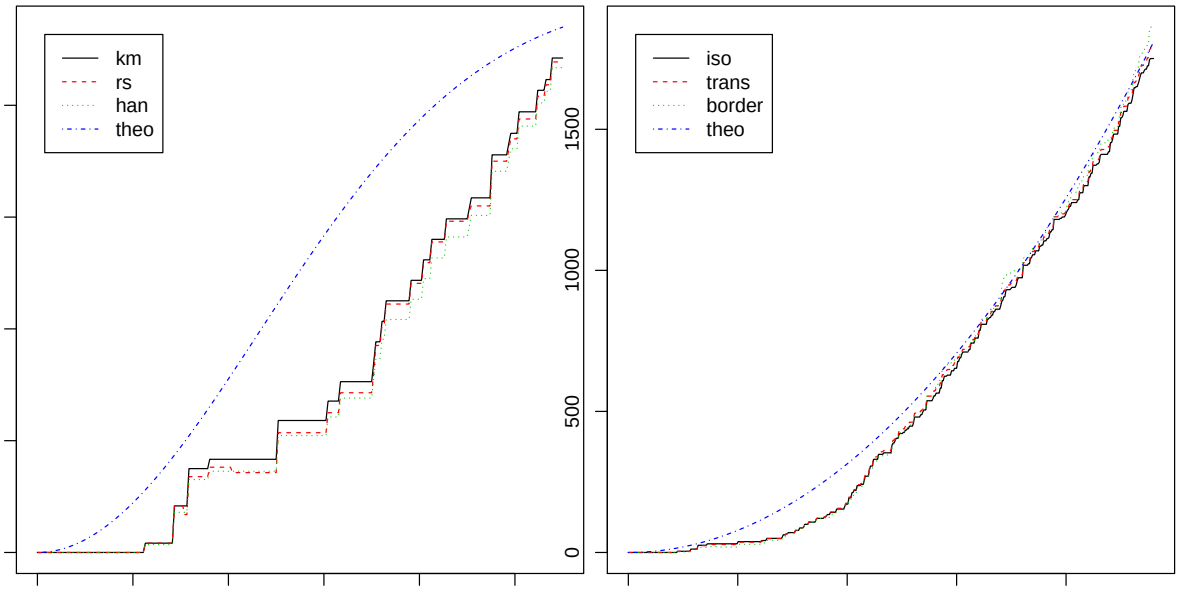
$$\widehat{K}(r) = \frac{1}{\widehat{\lambda}^2 \mathrm{area}(W)} \sum_i \sum_{j \neq i} \mathbf{1}\left\{||x_i - x_j|| \leq r\right\} e(x_i, x_j; r) \tag{7}$$

where $e(u, v, r)$ is an edge correction weight.

Functional summary statistics in spatstat

In spatstat the functions Gest and Kest computes several estimates of respectively the G-function and the K-function.

```
> data(swedishpines)
> G <- Gest(swedishpines)
> K <- Kest(swedishpines)
> plot(G, main = "")
> plot(K, main = "")
```



Other functional summary statistics

Other commonly used functional summary statistics (and their values for the homogeneous Poisson process) are:

- The L-function
$$L(r) = \sqrt{\frac{K(r)}{\pi}},$$
with $L_p(r) = r$, and `spatstat`-command `Lest`.
- The pair-correlation function
$$g(r) = \frac{K'(r)}{2\pi r},$$
with $g_p(r) = 1$, and `spatstat`-command `pcf`.
- The J-function
$$J(r) = \frac{1 - G(r)}{1 - F(r)},$$
with $J_p(r) = 1$, and `spatstat`-command `Jest`.

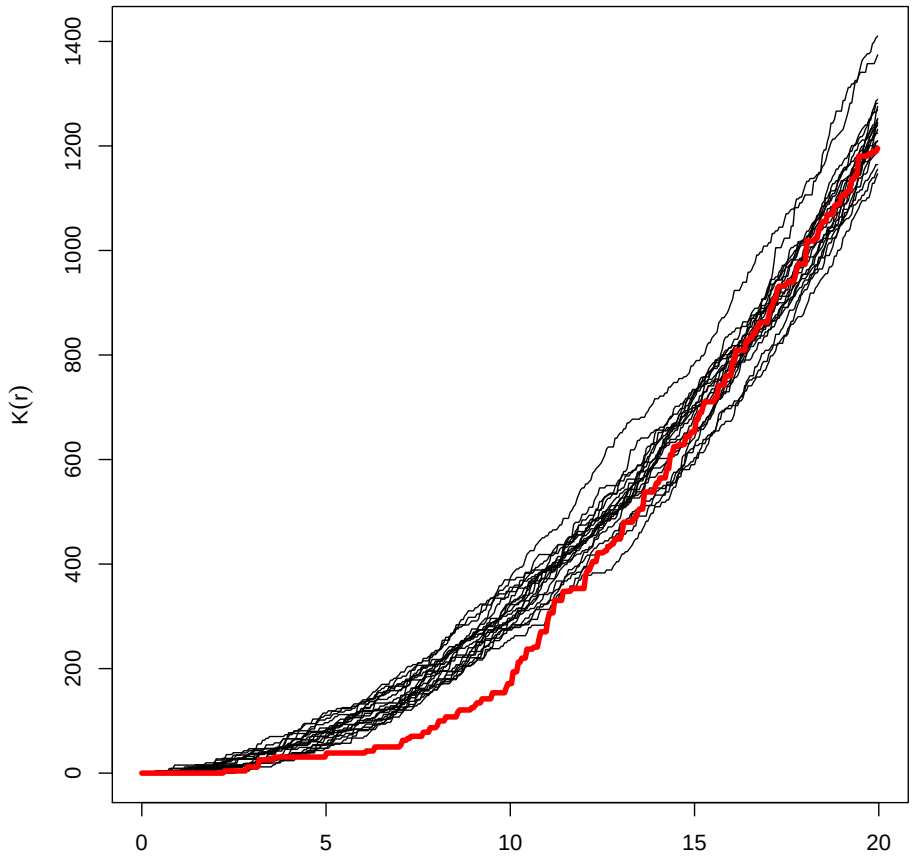
Cautions about functional summary statistics

It's important to remember that

1. the functions F, G and K are defined and estimated under the *assumption that the point process is stationary (homogeneous).*
2. these summary functions *do not completely characterise the process.*
3. if the process is not stationary, deviations between the empirical and theoretical functions (e.g. $\widehat{K}$ and K_p) are not necessarily evidence of interpoint interaction, since they may also be attributable to variations in intensity.

Monte Carlo tests using functional summary statistics

We wish to some how judge the significance of departure of an empirical line from a theoretical line based on CSR. Below is the K-function estimated from the `swedishpines` dataset, and also the K-functions of 20 simulated realisations of CSR.



Monte Carlo tests using functional summary statistics

Suppose we have M independent simulations of the null model (CSR) inside the study region W. Compute the estimated K functions for each of these realisations, say $\widehat{K}^{(j)}(r)$ for $j = 1, \ldots, M$. Obtain the pointwise lower and upper envelopes of these simulated curves,

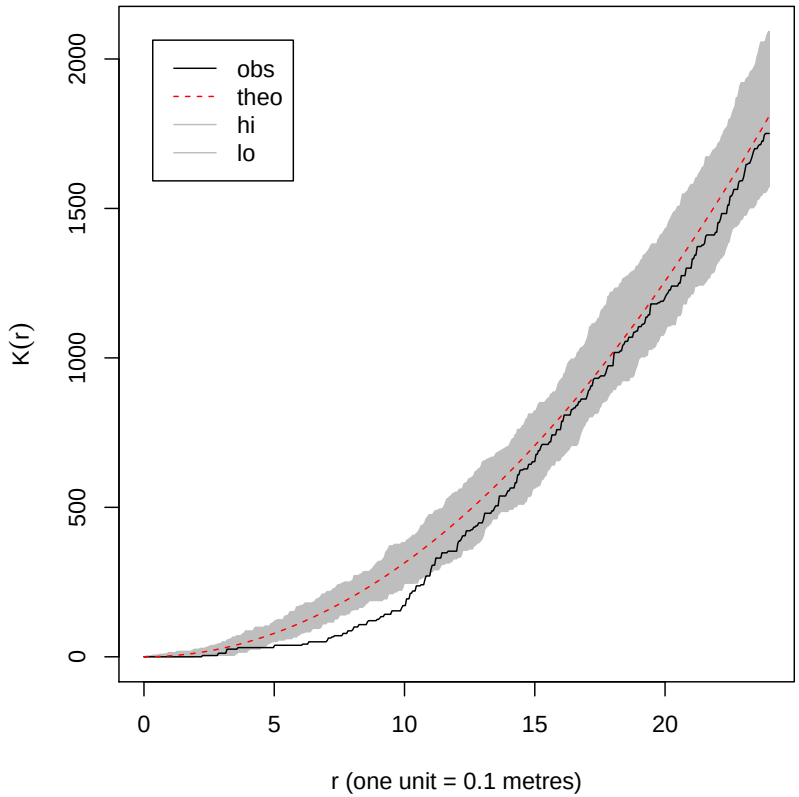
$$L(r) = \min_j \widehat{K}^{(j)}(r) \quad \text{and} \quad U(r) = \max_j \widehat{K}^{(j)}(r).$$

For any fixed value of r, consider the probability that $\widehat{K}(r)$ lies outside the envelope $[L(r), U(r)]$ for the simulated curves. Under the null model $\widehat{K}(r)$ and $\widehat{K}^{(1)}(r), \ldots, \widehat{K}^{(M)}(r)$ are statistically equivalent and independent, so this probability is equal to $2/(M+1)$ by symmetry.

Envelopes in spatstat

In spatstat the function envelope computes the pointwise envelopes.

```
> E <- envelope(swedishpines, Kest, nsim = 39, rank = 1)
> plot(E, main = "")
```



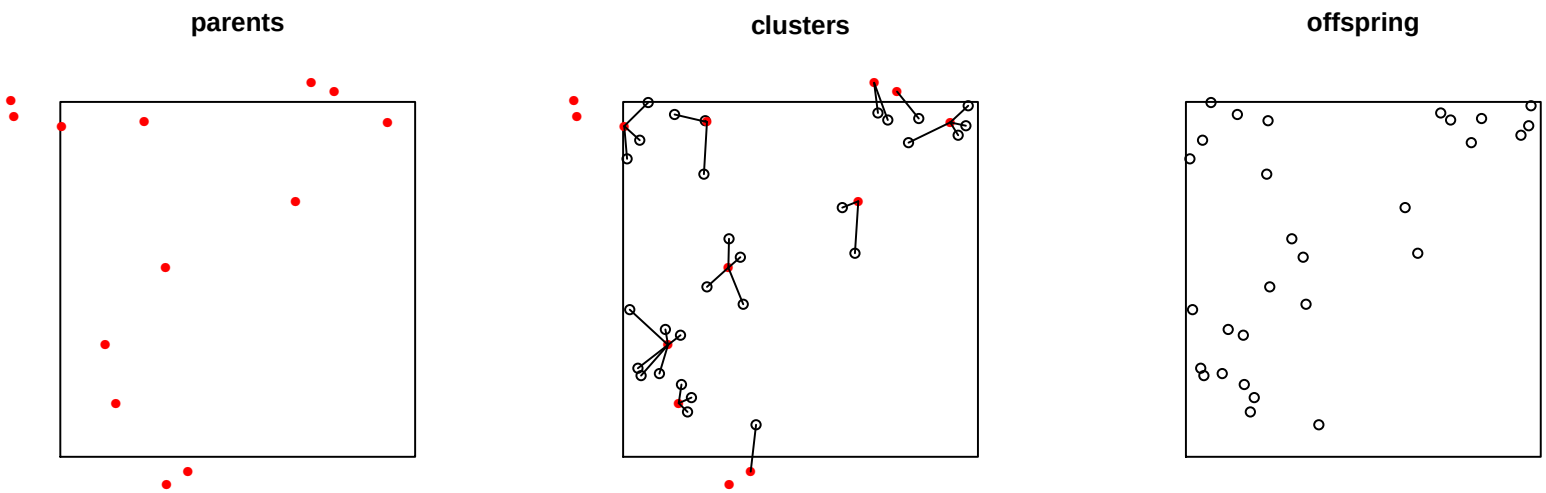
Simple models of non-Poisson patterns

In the next part of the lecture we will introduce a series of non-Poisson spatial point process models. For some of the models we consider simulation and statistical inference. The model classes we consider are

- Poisson cluster processes
- Cox processes
- Thinned processes
- Sequential models

Poisson cluster processes

In a Poisson cluster process, we begin with a Poisson process $\mathbf{Y}$ of 'parent' points. Each 'parent' point $y_i \in \mathbf{Y}$ then gives rise to a finite set Z_i of 'offspring' points according to some stochastic mechanism. The set comprising all the offspring points forms a point process $\mathbf{X}$. Only $\mathbf{X}$ is observed.



Poisson cluster processes

Two well-known examples with homogeneous parent processes are:

1. The *Matérn cluster process* in which each parent has a Poisson(μ) number of offspring, independently and uniformly distributed in a disc of radius r centred around the parent. This process can be generated in `spatstat` using the command `rMatClust`.
2. The *Thomas process*, in which each cluster consists of a Poisson(μ) number of random points, each having an isotropic Gaussian $N(0, \sigma^2 I)$ displacement from its parent. This process can be generated in `spatstat` using the command `rThomas`.

Cox processes

- A Cox point process is effectively a Poisson process with a random intensity function.
- Let $\Lambda(u)$ be a random function with non-negative values, defined at all locations $u \in \mathbb{R}^2$. Conditional on Λ, let $\mathbf{X}$ be a Poisson process with intensity function Λ. Then $\mathbf{X}$ is a Cox process, and the intensity is $\lambda(u) = \mathbb{E}[\Lambda(u)]$.
- A particularly interesting and useful class is that of *log-Gaussian Cox processes (LGCP)* in which $\log \Lambda(u)$ is a Gaussian random function.
- The Matérn Cluster process and the Thomas process are both Cox processes, but apart from these examples there are no functions in `spatstat` for generating Cox processes.

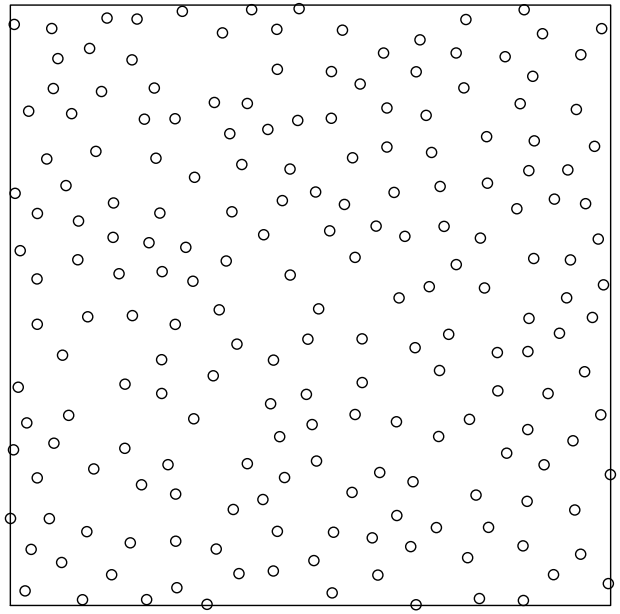
Thinned processes

- *'Thinning'* means deleting some of the points from a point pattern. Under *'independent thinning'* the thinning is done independently of other points, whereas for *dependent thinning* the deletion of a point may depend on the fate of other points.
- In *Matérn's Model I*, a homogeneous Poisson process $\mathbf{Y}$ is first generated. Any point in $\mathbf{Y}$ that lies closer than a distance r from the nearest other point of $\mathbf{Y}$, is deleted. Thus, pairs of close neighbours annihilate each other.
- In *Matérn's Model II*, the points of the homogeneous Poisson process $\mathbf{Y}$ are marked by 'arrival times' t_i which are independent and uniformly distributed in $[0, 1]$. Any point in $\mathbf{Y}$ that lies closer than distance r from another point that has an earlier arrival time, is deleted.
- The `spatstat`-functions are `rMaternI` and `rMaternII`.

Sequential models

In Simple Sequential Inhibition, each new point is generated uniformly in the window and independently of preceding points. If the new point lies closer than r units from an existing point, then it is rejected and another random point is generated. The process terminates when no further points can be added.

```
> plot(rSSI(0.05, 200), main = "")
```



Model-fitting using summary statistics

In a few cases, the K function of a point process is known exactly, as an analytic expression in terms of the model parameters. For example, the K-function of the Thomas process with parameters $\theta = (\kappa, \mu, \sigma)$ is (notice that the K-function does not depend on μ)

$$K_\theta(r) = \pi r^2 + \frac{1}{\kappa}(1 - \exp(-\frac{r^2}{4\sigma^2})).$$

The model can then be fitted by minimising

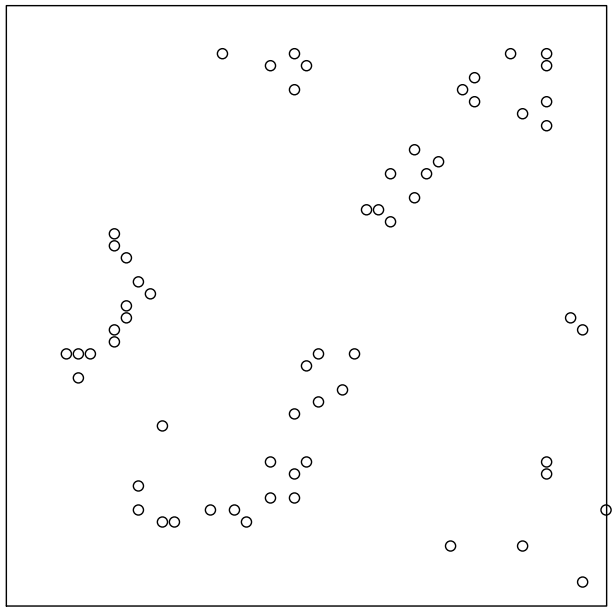
$$D(\theta) = \int_a^b \left| \widehat{K}(r)^q - K_\theta(r)^q \right|^p \, \mathrm{d}r \tag{8}$$

where $0 \leq a < b$, and where $p, q > 0$ are exponents. This method is known as the *method of minimum contrast*. (The remaining parameter μ is inferred from the estimated intensity).

Model-fitting using summary statistics

An example of a clustered point pattern is the `redwood` data.

```
> data(redwood)
> plot(redwood, main = "")
```



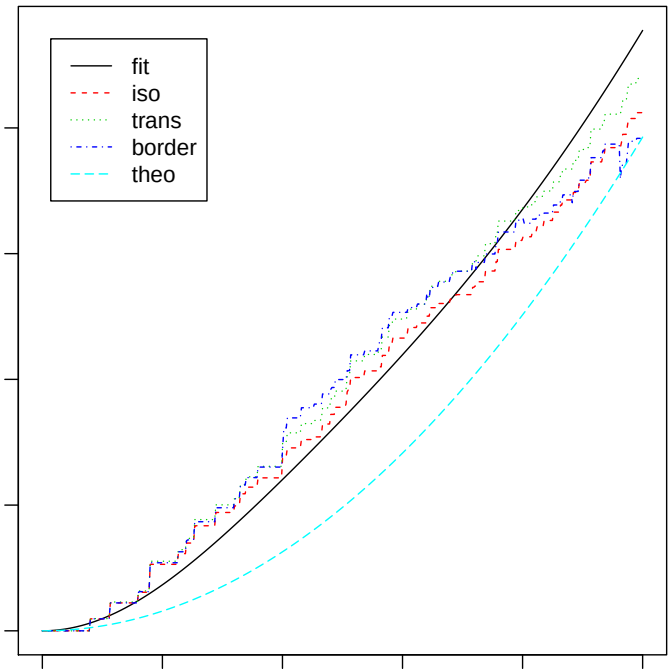
To fit the Thomas model to this data:

```
> fit <- kppm(redwood, ~1, "Thomas")
```

Model-fitting using summary statistics

The fitted model, `fit`, is an object of class `kppm`. There are methods for printing and plotting objects of this class. The plot shows the K-function for the fitted fitted model with empirical estimates $\hat{K}$ and the theoretical Poisson curve.

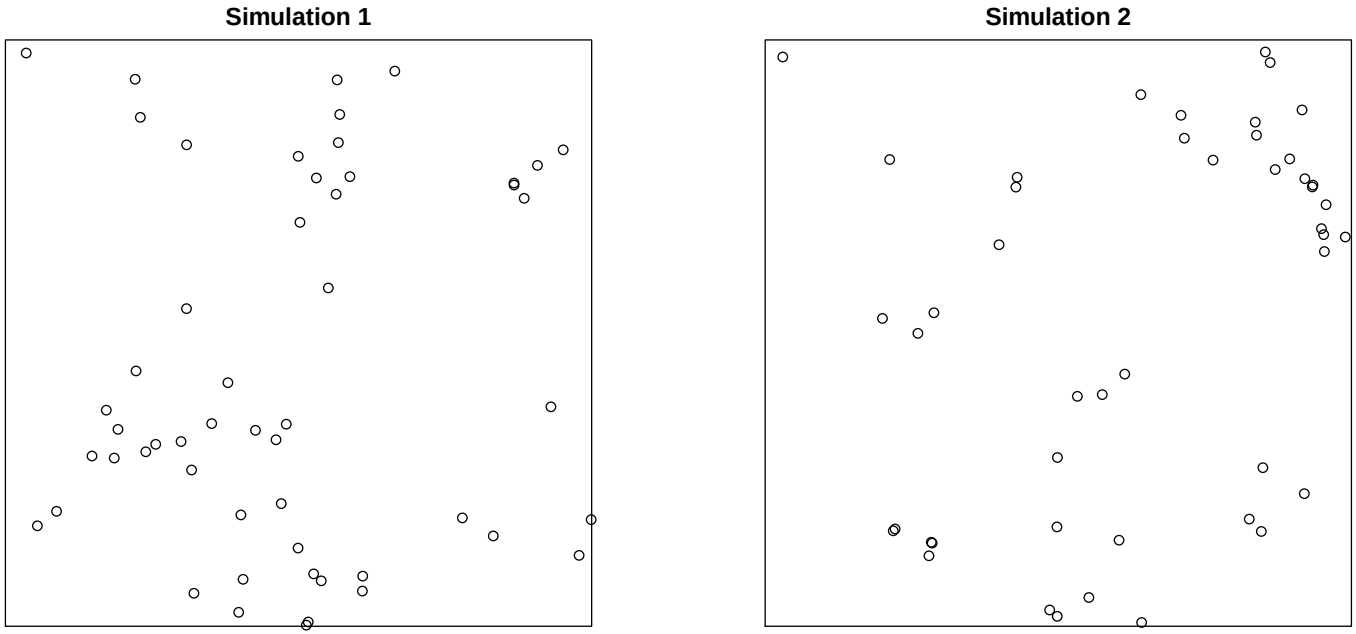
```
> plot(fit, main = "")
```



Model-fitting using summary statistics

A fitted model returned by `kppm` can be simulated immediately:

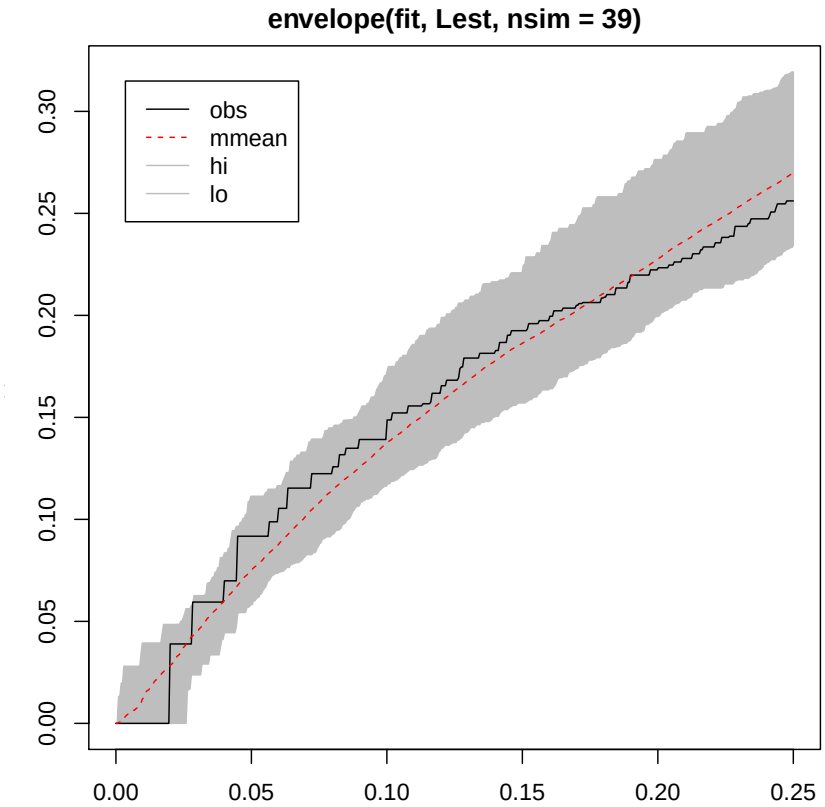
```
> sim <- simulate(fit, nsim = 2)
> plot(sim, main = "")
```



Model-fitting using summary statistics

Simulation envelopes are also available, using the function `envelope`.

```
> plot(envelope(fit, Lest, nsim = 39))
```



Fitting log-Gaussian Cox processes

The K-function is also known as a function of the model parameters for *log-Gaussian Cox processes*. To fit a log-Gaussian Cox process with constant mean μ and exponential covariance function

$$c(r) = \sigma^2 * \exp(-r/\alpha)$$

to the `redwood` data, simply run:

```
> fit <- lgcp.estK(redwood, c(sigma2 = 0.1, alpha = 1))
```

The second argument to `lgcp.estK` gives initial values for the model parameters σ^2 and α. The result of `lgcp.estK` is an object of class `minconfit` (representing a 'minimum contrast fit').

Extensions

- It is also possible to base estimation on the pair correlation function (implemented as `lgcp.estpcf` for log-Gaussian Cox Processes).
- More generally minimum contrast can be used for any function where the theoretical function can be computed exactly from the model parameters. This is implemented in the generic fitting algorithm `mincontrast`.
- It is possible to adjust for inhomogeneity in some functional summary statistics as the K-function and the pair correlation function. These intensity adjusted versions are also implemented in `spatstat`.
- Cluster models can be adjusted to include inhomogeneity as well, and inhomogeneous models can aslo be fitted by minimum contrast.

Introduction to Gibbs models

- The density (with respect to the unit rate Poisson process) of a Poisson process on W with intensity λ_θ parametrised by θ is

$$f(\mathbf{x}) = \underbrace{\exp\left(\int_W (1 - \lambda_\theta(u))\,\mathrm{d}u\right)}_{\alpha} \prod_i \lambda_\theta(x_i), \tag{9}$$

  where α is the normalizing constant only depending on the parameters and not on the data.
- Notice that the density is a product of terms associated with individual points x_i. This reflects the conditional independence property of the Poisson process.
- In the following we will construct point process models with interaction between points by specifying the density of the model. The models we create in this way are all so-called Gibbs models.

Pairwise interaction models

Pairwise interaction models have probability densities of the form

$$f(\mathbf{x}) = \alpha \left[\prod_{i=1}^{n(\mathbf{x})} b(x_i) \right] \left[\prod_{i<j} c(x_i, x_j) \right] \tag{10}$$

where α is a normalising constant, $b(u),\ u \in W$ is the 'first order' term, and $c(u, v),\ u, v \in W$ is the 'second order' or 'pairwise interaction' term, which must be symmetric, $c(u, v) = c(v, u)$. In principle we are free to choose any functions b and c, provided the resulting density is integrable with respect to the unit rate Poisson process.

Hard core process

If we take $b(u) \equiv \beta$ and

$$c(u,v) = \begin{cases} 1 & \text{if } ||u-v|| > r \\ 0 & \text{if } ||u-v|| \leq r \end{cases} \tag{11}$$

where $||u-v||$ denotes the distance between u and v, and $r > 0$ is a fixed distance, then the density becomes

$$f(\mathbf{x}) = \begin{cases} \alpha\beta^{n(\mathbf{x})} & \text{if } ||x_i - x_j|| > r \text{ for all } i \neq j \\ 0 & \text{otherwise} \end{cases}$$

This is known as the *hard core process*.

Strauss process

Generalising the hard core process, suppose we take $b(u) \equiv \beta$ and

$$c(u,v) = \begin{cases} 1 & \text{if } ||u-v|| > r \\ \gamma & \text{if } ||u-v|| \leq r \end{cases} \tag{12}$$

where γ is a parameter. Then the density becomes

$$f(\mathbf{x}) = \alpha \beta^{n(\mathbf{x})} \gamma^{s(\mathbf{x})} \tag{13}$$

where $s(\mathbf{x})$ is the number of pairs of distinct points in $\mathbf{x}$ that lie closer than r units apart.
The parameter γ controls the 'strength' of interaction between points. If $\gamma = 1$ the model reduces to a Poisson process with intensity β. If $\gamma = 0$ the model is a hard core process. For values $0 < \gamma < 1$, the process exhibits inhibition (negative association) between points.

Other models

- A more general model is the pairwise interaction model with *piecewise constant* interaction in which $c(||u - v||)$ is a step function of $||u - v||$.
- An example of a model with interactions of all orders is the *area-interaction* or Widom-Rowlinson process with probability density

$$f(\mathbf{x}) = \alpha\beta^{n(\mathbf{x})}\gamma^{-A(\mathbf{x})} \tag{14}$$

where α is the normalising constant, $\beta > 0$ is an intensity parameter, and $\gamma > 0$ is an interaction parameter. Here $A(\mathbf{x})$ denotes the area of the region obtained by drawing a disc of radius r centred at each point x_i, and taking the union of these discs. The value $\gamma = 1$ again corresponds to a Poisson process, while $\gamma < 1$ produces a regular process and $\gamma > 1$ a clustered process.

Time for a break!