Spatial point processes: Theory and practice illustrated with R

Ege Rubak

Department of Mathematical Sciences
Aalborg University

Lecture II, February 17, 2011
Contents of lecture II

- Properties for the Poisson point process.
- Maximum likelihood estimation for Poisson point processes.
- Investigating dependence between points using functional summary statistics.
- Relevant commands in spatstat (remember to load the package when a new session is started):
  > library(spatstat)
Homogeneous Poisson processes

The homogeneous Poisson process with intensity $\lambda$ is defined by two properties:

- $N(A)$ is Poisson distributed with mean $\lambda|A|$, for all $A$.
- Conditional on $N(A) = n$ the $n$ points are independent and uniformly distributed in $A$.

The counts $N(A)$ and $N(B)$ are independent for disjoint $A, B$.

We will use the homogeneous Poisson process as the basis for creating more complicated point process models.
A point process $\mathbf{X}$ in the window $\mathcal{W}$ has density $f(\cdot)$ with respect to the unit rate Poisson process if

$$\mathbb{E}[h(\mathbf{X})] = \mathbb{E}[h(\mathbf{Y})f(\mathbf{Y})]$$

(1)

for all functionals $h$, where $\mathbf{Y}$ is a unit rate Poisson process (i.e. $\lambda = 1$).

In particular the homogeneous Poisson process with intensity $\lambda$ has density

$$f(\mathbf{x}) = e^{(1-\lambda)|\mathcal{W}|} \lambda^{|\mathbf{x}|}.$$  

(2)

The maximum likelihood estimate $\hat{\lambda}$ of the intensity $\lambda$ is

$$\hat{\lambda} = \frac{n(\mathbf{x})}{|\mathcal{W}|}.$$
Inhomogeneous Poisson processes

The inhomogeneous Poisson process with intensity function \( \lambda(\cdot) \) is defined by two properties:

- \( N(A) \) is Poisson distributed with mean \( \int_A \lambda(u) \, du \), for all \( A \).
- Conditional on \( N(A) = n \) the \( n \) points are independent and identically distributed in \( A \) with density proportional to \( \lambda(\cdot) \).

The counts \( N(A) \) and \( N(B) \) are still independent for disjoint \( A, B \).

The density of an inhomogeneous Poisson process on \( W \) with intensity \( \lambda_\theta \) parametrised by \( \theta \) is

\[
f(x) = \exp \left( \int_W (1 - \lambda_\theta(u)) \, du \right) \prod_i \lambda_\theta(x_i). \quad (3)
\]

The MLE \( \hat{\theta} \) is not analytically tractable, so it must be computed using numerical algorithms which will be discussed further in next weeks lectures.
For simulation a useful technique is independent thinning: Suppose that $X$ is a Poisson process with intensity function $\beta(\cdot)$, and that each point of $X$ is either deleted or retained, independently of other points. If the retention probability at location $u$ is $p(u)$, then the resulting process of retained points is Poisson, with intensity function $\lambda(u) = \beta(u)p(u)$, for all $u$.

To simulate an inhomogeneous Poisson process with intensity function $\lambda(\cdot)$ we start by simulating a homogeneous Poisson process with intensity $\lambda_{\text{max}} = \max_u \{\lambda(u)\}$. Then a thinning is performed with retention probabilities $p(u) = \lambda(u)/\lambda_{\text{max}}$ to obtain the inhomogeneous process.
Inhomogeneous Poisson processes

To simulate an inhomogeneous Poisson process we can supply the intensity as a function or a pixel image:

```r
> intenfun <- function(x, y) 200 * x
> plot(rpoispp(intenfun, lmax = 200), main = "")
> intenim <- as.im(intenfun, dimyx = c(100, 100),
+ W = unit.square())
> plot(rpoispp(intenim), main = "")
```
Introduction  Inhomogeneous Poisson processes  Fitting models in spatstat  Inter-point dependence

Model-fitting in spatstat

- The spatstat implementation requires the intensity function $\lambda_\theta(u)$ to be loglinear in the parameter $\theta$:

$$\log \lambda_\theta(u) = \theta \cdot S(u)$$  \hspace{1cm} (4)

where $S(u)$ is a real-valued or vector-valued function of location $u$. The form of $S$ is arbitrary so this is not much of a restriction.

- In practice $S(u)$ could be a function of the spatial coordinates of $u$, or an observed covariate, or a mixture of both.
Model-fitting in spatstat

- The fitting function is called `ppm` (‘point process model’) and is very closely analogous to the model fitting functions in R such as `lm` and `glm`.

- The statistic $S(u)$ is specified by an R language formula, like the formulas used to specify the systematic relationship in a linear model or generalised linear model.

- The basic syntax is:
  
  ```
  > ppm(X, ~trend)
  ```

  where $X$ is the point pattern dataset, and $~trend$ is an R formula with no left-hand side. This should be viewed as a model with log link, so the formula $~trend$ specifies the form of the logarithm of the intensity function.
Model-fitting in spatstat

Fitting a Poisson model with an intensity that is log-linear in the cartesian coordinates, i.e. \( \lambda_\theta((x, y)) = \exp(\theta_0 + \theta_1 x + \theta_2 y) \):

```r
> data(bei)
> ppm(bei, ~x + y)
```

Nonstationary Poisson process

Trend formula: \( \sim x + y \)

Fitted coefficients for trend formula:

(Intercept) x y
-4.7245290274 -0.0008031288 0.0006496090

In this case the fitted intensity function is

\[ \lambda_\theta((x, y)) = \exp(-4.724529 + -0.000803 x + 0.00065 y). \]
Model-fitting in spatstat

To fit an inhomogeneous Poisson model with an intensity that is log-quadratic in the cartesian coordinates, i.e. such that $\log \lambda_\theta((x,y))$ is a quadratic in $x$ and $y$:

\[
\text{> ppm(bei, } \sim \text{polynom}(x, y, 2))
\]

Nonstationary Poisson process

Trend formula: $\sim\text{polynom}(x, y, 2)$

Fitted coefficients for trend formula:

(Intercept)  polynom(x, y, 2)[x]
-4.275762e+00  -1.609187e-03

polynom(x, y, 2)[y]  polynom(x, y, 2)[x^2]
-4.895166e-03  1.625968e-06

polynom(x, y, 2)[x.y]  polynom(x, y, 2)[y^2]
-2.836387e-06  1.331331e-05
Model-fitting in spatstat

To fit a model with constant but unequal intensities on each side of the vertical line $x = 500$, the explanatory variable $S(u)$ should be a factor with two levels, Left and Right say, taking the value Left when the location $u$ is to the left of the line $x = 500$.

```r
> side <- function(z) factor(ifelse(z < 500, "left", + "right"))
> ppm(bei, ~side(x))
```

Nonstationary Poisson process

Trend formula: $\sim$side(x)

Fitted coefficients for trend formula:

(Intercept) side(x)right
-4.8026460 -0.2792705
For the rainforest data we may want to fit the inhomogeneous Poisson model with intensity which is a loglinear function of slope, i.e.

\[ \lambda(u) = \exp(\beta_0 + \beta_1 Z(u)) \]  

where \( \beta_0, \beta_1 \) are parameters and \( Z(u) \) is the slope at location \( u \):

```r
> slope <- bei.extra$grad
> ppm(bei, ~slope, covariates = list(slope = slope))
```

Nonstationary Poisson process

Trend formula: \(~\text{slope}\)

Fitted coefficients for trend formula:

<table>
<thead>
<tr>
<th>(Intercept)</th>
<th>slope</th>
</tr>
</thead>
<tbody>
<tr>
<td>-5.390553</td>
<td>5.022021</td>
</tr>
</tbody>
</table>
Model-fitting in `spatstat`

It might be more appropriate to fit the model

\[ \lambda(u) = \beta Z(u) \]  \hspace{1cm} (6)

where again \( Z(u) \) is the slope at \( u \). Equivalently

\[ \log \lambda(u) = \log \beta + \log Z(u). \]  \hspace{1cm} (7)

There is no coefficient in front of the term \( \log Z(u) \) in (7), so this term is an ‘offset’. To fit this model,

```r
> ppm(bei, ~offset(log(slope)), covariates = list(slope = slope))
```

Nonstationary Poisson process

Trend formula: \( \sim \)offset(log(slope))

Fitted coefficients for trend formula:

(Intercept)

\[-2.427127\]
Predicting the intensity function

The values of the intensity function can be calculated on a grid in the window for a fitted model via the function `predict`. Alternatively, we can simply plot the fitted object which calls the `predict`-function and plots the intensity function:

```
> fit <- ppm(bei, ~x + y)
> plot(fit, how = "image", se = FALSE, pause = FALSE)
```

![Fitted trend](image)
For Poisson models it is possible to calculate the variance-covariance matrix of $\hat{\theta}$:

$$> \text{vcov(fit)}$$

(Intercept)        x          y
(Intercept) 1.854091e-03 -1.491267e-06 -3.528289e-06
x -1.491267e-06   3.437842e-09  1.208410e-14
y -3.528289e-06   1.208410e-14  1.338955e-08

This can be used to give standard deviations on the estimates:

<table>
<thead>
<tr>
<th>$i$</th>
<th>$\theta_i$</th>
<th>$\text{Var}(\hat{\theta}_i)$</th>
<th>standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-4.724529</td>
<td>0.001854091</td>
<td>0.04305915</td>
</tr>
<tr>
<td>1</td>
<td>-0.0008031288</td>
<td>3.437842e-09</td>
<td>5.863311e-05</td>
</tr>
<tr>
<td>2</td>
<td>0.000649609</td>
<td>1.338955e-08</td>
<td>0.0001157132</td>
</tr>
</tbody>
</table>
Covariate effect plot

It is also possible to plot the ‘effect’ of a single covariate in the model. The command `effectfun` computes the intensity of the fitted model as a function of one of its covariates. This is chiefly useful if the model only has one covariate.

```r
> fit <- ppm(bei, ~slope, covariates = list(slope = slope))
> plot(effectfun(fit, "slope"), main = "")
```

![Graph showing the effect of slope on intensity](image-url)
Analysis of deviance for nested Poisson point process models is done by using the function `anova`:

```r
> fitnull <- update(fit, ~1)
> anova(fitnull, fit, test = "Chi")
```

Analysis of Deviance Table

<table>
<thead>
<tr>
<th>Model 1: .mpl.Y ~ 1</th>
<th>Model 2: .mpl.Y ~ slope</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resid. Df</td>
<td>Resid. Dev</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>1</td>
<td>20507</td>
</tr>
<tr>
<td>2</td>
<td>20506</td>
</tr>
</tbody>
</table>

---

Signif. codes: 0 ^ aĂŸ***^ aĂž 0.001 ^ aĂŸ**^ aĂž 0.01 ^ aĂŸ*^ aĂž 0.05 ^ aĂŸ ^ aĂž 1

Clearly, the model including slope is favored.
The homogeneous Poisson process is not nested in the model (6), in which intensity is proportional to slope. One possibility here is to use the Akaike Information Criterion **AIC** for model selection:

```r
> fit <- ppm(bei, ~offset(log(slope)), covariates = list(slope = slope))
> fitnull <- ppm(bei, ~1)
> AIC(fit)
[1] 42496.65
> AIC(fitnull)
[1] 42763.92
```

The smaller AIC favours the model (6) with intensity proportional to slope.
A fitted Poisson model can be simulated automatically using the `simulate` function:

```r
> X <- simulate(fit)
> plot(X, main = "")
```

Simulation 1
Suppose that a point pattern appears to have constant intensity, and we wish to assess whether the pattern is Poisson. The alternative is that the point pattern is either regular or clustered.
The *Fry plot* is a scatterplot of the vector differences \(x_j - x_i\) between all pairs of distinct points in the pattern. It is computed by the command `fryplot`.

![Fry plot examples](image_url)
Classical techniques for investigating interpoint interaction are *distance methods*, based on measuring the distances between points. Specifically we may consider

- **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.

These are available in spatstat as the functions `pairdist`, `nndist` and `distmap` respectively.
Empty space distances

```r
> data(swedishpines)
> emp <- distmap(swedishpines)
> plot(emp, main = "Empty space distances")
> plot(swedishpines, add = TRUE)
```
Edge effects

Confining observations to a window $W$ implies that the observed distance $d(u, x) = d(u, X \cap W)$ to the nearest data point inside $W$, may be greater than the true distance $d(u, X)$ to the nearest point of the complete point process $X$. 
Empty space function \( F \)

- Assuming \( X \) is *stationary* (statistically invariant under translations), we can define the cumulative distribution function of the empty space distance

\[
F(r) = \mathbb{P} \{ d(u, X) \leq r \} \tag{8}
\]

where \( u \) is an arbitrary reference location. If the process is stationary then this definition does not depend on \( u \).

- The empirical cumulative distribution function (ecdf) of the observed empty space distances on a grid of locations \( u_j, j = 1, \ldots, m \),

\[
F^*(r) = \frac{1}{m} \sum_j 1 \{ d(u_j, x) \leq r \} \tag{9}
\]

is a negatively biased estimator of \( F(r) \), as explained above.
Typically edge-corrected estimates of $F$ are weighted versions of the ecdf,

$$\hat{F}(r) = \sum_{j} e(u_j, r) 1 \{d(u_j, x) \leq r\}$$

(10)

where $e(u, r)$ is an edge correction weight designed so that $\hat{F}(r)$ is unbiased.

The edge effect problem can also be regarded as a form of censoring (analogous to right-censoring in survival data), and a counterpart of the Kaplan-Meier estimator is available.

Thus, *assuming that the point process is homogeneous*, we are able to compute an unbiased and reasonably accurate estimate of the empty space function $F$ defined by (8).
Empty space function: Poisson case

- Notice that $d(u, X) > r$ if and only if there are no points of $X$ in the disc $b(u, r)$ of radius $r$ centred on $u$. For a homogeneous Poisson process with intensity $\lambda$ the probability that there are no points in this region is $\exp(-\lambda \pi r^2)$. Hence for a Poisson process

\[
F_p(r) = 1 - \exp(-\lambda \pi r^2). \tag{11}
\]

- Typically we compare $\hat{F}(r)$ with the value of $F_p(r)$ obtained by plugging in the estimated intensity $\hat{\lambda} = n(x)/\text{area}(W)$. Values $\hat{F}(r) > F_p(r)$ suggest that empty space distances in the point pattern are shorter than for a Poisson process, suggesting a regularly space pattern; while values $\hat{F}(r) < F_p(r)$ suggest a clustered pattern.
The empty space function in spatstat

The function Fest computes estimates of $F(r)$ using several edge corrections, and the benchmark value for the Poisson process.

```r
> Fswed <- Fest(swedishpines)
> Fswed

Function value object (class `fv`)
for the function $r \rightarrow F(r)$
Entries:

<table>
<thead>
<tr>
<th>id</th>
<th>label</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>distance</td>
<td>argument $r$</td>
</tr>
<tr>
<td>theo</td>
<td>$F<a href="r">\text{pois}</a>$</td>
<td>theoretical Poisson $F(r)$</td>
</tr>
<tr>
<td>cs</td>
<td>$F<a href="r">\text{cs}</a>$</td>
<td>Chiu-Stoyan estimate of $F(r)$</td>
</tr>
<tr>
<td>rs</td>
<td>$F<a href="r">\text{bord}</a>$</td>
<td>border corrected estimate of $F(r)$</td>
</tr>
<tr>
<td>km</td>
<td>$F<a href="r">\text{km}</a>$</td>
<td>Kaplan-Meier estimate of $F(r)$</td>
</tr>
<tr>
<td>hazard</td>
<td>hazard$(r)$</td>
<td>Kaplan-Meier estimate of hazard function</td>
</tr>
</tbody>
</table>
```

Default plot formula: 
```
. ~ r
```

Recommended range of argument $r$: [0, 7.68]
Available range of argument $r$: [0, 22.32]
Unit of length: 0.1 metres
Introduction
Inhomogeneous Poisson processes
Fitting models in spatstat
Inter-point dependence

Plotting the empty space function

> plot(Fswed)
> plot(Fswed, cbind(km, rs, theo) ~ theo)
Plotting the empty space function

\[ \text{plot}(F_{\text{swed}}, \cdot \sim \text{theo}) \]
\[ \text{plot}(F_{\text{swed}}, \cdot - \text{theo} \sim r) \]

\[ F_{\text{swed}}(r) \]
\[ F(r) - F_{\text{pois}}(r) \]
Thank your for your attention.