Finite point processes

To define a finite point process $X$ on a bounded window $W$, one may specify

- a discrete probability distribution $(p_n)_{n \in \mathbb{N}_0}$ for the total number of points;

- a family of symmetric joint probability densities $j_n(x_1, \ldots, x_n)$, $n \in \mathbb{N}$, on $(\mathbb{R}^2)^n$ for the locations of the points given that there are $n$ of them.
Density function

The $p_n$ and $j_n$ may be combined in a single function

$$f(\{x_1, \ldots, x_n\}) = e^{|W| n!} p_n j_n(x_1, \ldots, x_n),$$

the density function of $X$.

The factor $n!$ in the right hand side occurs because $f$ is a function of unordered sets, whereas $j_n$ has ordered vectors as its argument.

The constant $e^{|W|}$ is a normalisation.
Recovering $p_n$ and $j_n$

Density function $f$ is defined uniquely in terms of $p_n$ and $j_n$. The reverse is also true. Indeed,

$$p_0 = e^{-|W|} f(\emptyset).$$

For $n \in \mathbb{N}$,

$$p_n = \frac{e^{-|W|}}{n!} \int_W \cdots \int_W f(\{u_1, \ldots, u_n\}) du_1 \cdots du_n$$

and

$$j_n(x_1, \ldots, x_n) = \frac{f(\{x_1, \ldots, x_n\})}{\int_W \cdots \int_W f(\{u_1, \ldots, u_n\}) du_1 \cdots du_n}.$$
Interaction

The presence of a point at location $w \in W$ may influence the likelihood of finding points ‘nearby’, e.g.

- points $v$ for which $||w - v|| \leq R$ for some $R > 0$;

- points $v$ in a zone $Z(w) \subseteq W$ around $w$.

If the zones $Z(\cdot)$ are not balls, the model is **anisotropic**.
Pairwise interaction models

A **pairwise interaction process** $X$ is a point process whose density function is of the form

$$f(x) \propto \prod_{x \in x} \beta(x) \prod_{\{u,v\} \subseteq x} \gamma(u, v)$$

for some function $\beta : W \to \mathbb{R}^+$ and some symmetric function $\gamma : W \times W \to \mathbb{R}^+$.

$\beta$ governs the **heterogeneity** or **trend**, $\gamma$ the **interaction**.
Example: Strauss process

\[ \gamma(u, v) = \begin{cases} \gamma & \text{if} \ ||u - v|| \leq R \\ 1 & \text{if} \ ||u - v|| > R \end{cases} \]

for \( \gamma \in [0, 1] \).

\( \gamma = 0 \) leads to a **hard core process**: no point is allowed to fall within distance \( R \) of another point.

\( \gamma = 1 \) corresponds to a Poisson process.

For intermediate values of \( \gamma \), points tend to avoid lying closer than \( R \) together, the tendency being stronger for smaller values of \( \gamma \).
Strauss process – simulation

strauss0 <- rStrauss(beta=100, gamma=0.0, R=0.1,  
W = square(1), expand = FALSE, nsim=1)
strauss4 <- rStrauss(beta=100, gamma=0.4, R=0.1,  
W = square(1), expand = FALSE, nsim=1)
strauss8 <- rStrauss(beta=100, gamma=0.8, R=0.1,  
W = square(1), expand = FALSE, nsim=1)

If expand = TRUE, the simulation is performed on a larger window and clipped. This is appropriate if \( X \) is the restriction to \( W \) of a point process defined on \( \mathbb{R}^2 \).
Realisations

Left to right: $\gamma = 0, 0.4$ and $0.8$; $R = 0.1$ and $\beta = 100$. 
Multi-step process

Piecewise constant pairwise interaction function

\[ \gamma(u, v) = \begin{cases} 
\gamma_j & \text{if } R_{j-1} \leq ||u - v|| < R_j \\
1 & \text{if } ||u - v|| \geq R_k
\end{cases} \]

for \( 0 = R_0 < R_1 < \cdots < R_k \) and \( \gamma_1, \ldots, \gamma_k \in \mathbb{R} \).

For an inhibition strength that decreases in interpoint distance, take

\[ \gamma_0 < \gamma_1 < \cdots < \gamma_k < 1. \]

For attraction combined with a hard core, take

\[ \gamma_0 = 0; \gamma_1, \ldots, \gamma_k > 1. \]
Multi-step process – simulation

The script

```r
r <- seq(0.02, 0.1, by=0.02)
gamma <- c(0.0, 0.2, 0.4, 0.6, 0.8)
ms <- list(beta=100, r=r, h=gamma)
mStep <- rmhmodel(cif="lookup", par=ms, w=square(1))

X <- rmh(mStep,
    start=list(n.start=50), control=list(nrep=1e6))
```

generates an approximate realisation of the multi-step process by the Metropolis–Hastings method (lecture 2) starting from a binomial point process with 50 points and run for $10^6$ iterations.
Realisations
Influence zone based interaction

Define an influence function \( \kappa : W \times W \to \mathbb{R}^+ \) supported on \( Z \), i.e.

\[
Z(x) = \{w \in W : \kappa(w, x) > 0\} \subseteq W,
\]

and write

\[
c_x(w) = \sum_{i=1}^{n(x)} \kappa(w, x_i).
\]

Then a shot noise weighted point process on \( W \) with potential function \( V(\cdot) \) is defined by

\[
f(x) \propto \beta^{n(x)} \exp \left[ -\log \gamma \int_W V(c_x(w)) \, dw \right]
\]

where \( \beta, \gamma > 0 \) and \( V : \mathbb{R} \to \mathbb{R} \) with \( V(0) = 0 \).
Example: Area-interaction process

Let \( \kappa(w, x) = 1\{w \in Z(x)\} \). Then

\[
c_x(w) = \sum_{i=1}^{n} 1\{w \in Z(x_i)\}
\]

is the **coverage function** of \( x \). For \( V(x) = 1\{x > 0\} \),

\[
f(x) \propto \beta^n(x) \exp \left[ - \left| \bigcup_{x \in x} Z(x) \right| \log \gamma \right].
\]

For \( \gamma > 1 \), realisations tend to be **clustered** to cover a minimum of space.

For \( \gamma < 1 \), **regular** configurations are favoured.

\( \gamma = 1 \) corresponds to a Poisson process.
Area-interaction process – simulation

```r
iaPar <- list(beta=100, eta=1.5, r=0.1)
iai <- rmhmodel(cif="areaint", par=aiPar, w=square(1))

X1 <- rmh(ai,
    start=list(n.start=50), control=list(nrep=1e6))
```

 generates an approximate realisation of the isotropic model with \( Z(w) = B(w, R), \ R > 0. \)

spatstat uses a parametrisation with

\[
\eta = \gamma \pi R^2
\]

for numerical stability reasons.
Realisations

Left: $\eta = 1.5$; Right: $\eta = 0.5$. In both cases, $R = 0.1$. 
Generalisations

The $l_1$ function $c_x$ may be replaced by the $l_\infty$ function

$$\tilde{c}_x(w) = \max_{x \in x} \kappa(w, x)$$

For example, the multi-step area-interaction model based on the potential $V(x) = x$ and influence function

$$\kappa(u, v) = \begin{cases} \kappa_j & \text{if } R_{j-1} \leq \|u - v\| < R_j \\ 0 & \text{if } \|u - v\| \geq R_k \end{cases}$$

with $1 = \kappa_1 > \kappa_2 > \cdots \kappa_k > 0$ is defined by

$$f(x) \propto \beta^n(x)^\gamma - \int_W \max_{x \in x} \kappa(w, x) dw$$

$$= \beta^n(x)^\gamma - \sum_{j=1}^k \kappa_j |\{w \in W : d(w, x) \in [R_{j-1}, R_j]\}|.$$
Technical remark

When defining a model by its density function $f(\cdot)$, one needs to make sure that

$$\sum_{n=0}^{\infty} \frac{e^{-|W|}}{n!} \int_{W} \cdots \int_{W} f(\{x_1, \ldots, x_n\}) dx_1 \cdots dx_n < \infty.$$ 

A sufficient condition is that $f$ is locally stable: there exists some $\beta > 0$ such that

$$f(\{x_1, \ldots, x_n, x_{n+1}\}) \leq \beta f(\{x_1, \ldots, x_n\})$$

for all $\{x_1, \ldots, x_n\} \subset W$, all $n \in \mathbb{N}$ and all $x_{n+1} \in W$. 

17
Maximum likelihood estimation

Let \( x \) be a realisation of a Strauss process with parameters \( \beta(\cdot) \equiv \beta > 0 \) and \( \gamma \in [0, 1] \) in window \( W \subset \mathbb{R}^2 \).

Write

\[
S(x) = \sum_{\{u,v\} \subseteq x} 1\{||u - v|| \leq R\}.
\]

Then the log likelihood function becomes

\[
L(\beta, \gamma) = n(x) \log \beta + S(x) \log \gamma - \log Z(\beta, \gamma)
\]

but

\[
Z(\beta, \gamma) = \sum_{n=0}^{\infty} \frac{e^{-|W|}}{n!} \int_{W} \cdots \int_{W} \beta^n \gamma^{S(\{x_1, \ldots, x_n\})} dx_1 \cdots dx_n
\]

depends on the parameters and cannot be evaluated explicitly.
Pseudo-likelihood ideas

Idea: Approximate the log likelihood by that of a Poisson process with intensity function

$$\lambda_\theta(u|x) = \frac{f(x \cup \{u\})}{f(x)}$$

the conditional probability of finding a point at $u \notin x$ given configuration $x$ elsewhere. Here $\lambda_\theta(u|x) = 0$ when $f(x) = 0$.

Advantage: $\lambda_\theta(u|x)$ does not depend on the proportionality constant $Z(\theta)$.

Disadvantage: the approximation may be poor when the interaction is strong.
Maximum pseudo-likelihood estimation

Let $x$ be a realisation of a finite point process defined by a density function $f(x; \theta)$ that depends on a parameter $\theta$.

The log pseudo-likelihood function is defined as

$$PL(\theta) = \sum_{i=1}^{n} \log \lambda_\theta(x_i|x \setminus \{x_i\}) - \int_W \lambda_\theta(w|x) \, dw.$$ 

Optimise numerically over the parameter $\theta$ to obtain the maximum pseudo-likelihood estimate $\hat{\theta}$. 

20
Exponential family models

The models we presented take the form

\[ f_{\theta}(x) = \frac{1}{Z(\theta)} \exp \left[ \sum_{j=1}^{p} \theta_j C_j(x) \right], \]

in other words, form an exponential family with sufficient statistics \( C_j \) and parameters \( \theta_j, j = 1, \ldots, p \).

Hence, for \( u \not\in x \),

\[ \log \lambda_\theta(u|x) = \sum_{j=1}^{p} \theta_j \left[ C_j(x \cup \{u\}) - C_j(x) \right] \]

so \( PL(\theta) \) reads

\[ \sum_{j=1}^{p} \sum_{i=1}^{n} \theta_j \left[ C_j(x) - C_j(x \setminus \{x_i\}) \right] - \int_W e^{\sum_{j=1}^{p} \theta_j \left[ C_j(x \cup \{w\}) - C_j(x) \right]} dw. \]
Maximum pseudo-likelihood estimator

Writing

\[ C_j(u; x) = C_j(x \cup \{u\}) - C_j(x), \]
the score equations are

\[ \int_W C_j(w; x) \lambda_\theta(w|x) du = \sum_{i=1}^{n} C_j(x_i; x \setminus \{x_i\}) \]
for \( j = 1, \ldots, p. \)

The Hessian matrix \( H(\theta) \) of second order partial derivatives has entries

\[ \frac{\partial^2}{\partial \theta_i \partial \theta_j} PL(\theta) = - \int_W C_i(w; x) C_j(w; x) \lambda_\theta(w|x) dw. \]

Note that \( H(\theta) \) does depend on \( x. \)
Maximum pseudo-likelihood estimator – remarks

- In general, the score equations cannot be solved explicitly.

- Any $\hat{\theta}$ that solves the score equations and for which $H(\hat{\theta})$ is negative definite is a local maximum of the log pseudo-likelihood function $PL(\theta)$.

- $PL(\theta)$ involves an integral that must be approximated.

- Little is known about the small sample properties of $\hat{\theta}$. 
Increasing window asymptotics

When the window $W$ grows to $\mathbb{R}^2$,

- the limit distribution of $X \cap W$ may not exist;

- if it does, it may depend on boundary conditions so not be unique (phase transition).

Asymptotic normality of $\hat{\theta}$ was proved under strong ergodicity conditions.

The asymptotic covariance matrix cannot be evaluated explicitly. **Spatstat** has implemented the Cœurjolly and Rubak (2013) estimator.
Example: Barro Colorado data

Two models were fitted, the best one based on a polynomial trend. Consider adding interaction, e.g.,

\[
f(x) \propto \exp \left[ \sum_{j=1}^{15} \eta_j \sum_{x \in x} B_j(x) - \log\gamma \left| \sum_{x \in x} B(x, R) \right| \right],
\]

by

```r
fitbeiAI <- update.ppm(fitbeiXY, interaction=AreaInter(r=5))
```

which yields

Disc radius: 5
Fitted interaction parameter eta: 16.7755
Barro Colorado data: Results

\[
\text{plot(predict(fitbeiAI, type="trend"))}
\]
\[
\text{plot(predict(fitbeiAI, type="cif"))}
\]

Left: trend \(\exp \left[ \sum_{j=1}^{15} \hat{\eta}_j B_j(x) \right] \); Right: cif \(\lambda_{\hat{\theta}}(x|x)\).
Model validation by residuals

A residual analysis is based on

\[ s(x) = h^{-2} \sum_{y \in x} \kappa \left( \frac{x - y}{h} \right) w_h(x, y)^{-1} \]

\[ -h^{-2} \int_W \kappa \left( \frac{x - w}{h} \right) w_h(x, w)^{-1} \lambda(w|x) dw, \]

where \( \kappa \) is a probability density function and \( w_h \) an edge correction factor.

In \texttt{spatstat}, use

```r
a <- diagnose.ppm(fitbeiAI, which="smooth", sigma=100)
> sum(a$smooth$Z)
[1] -0.2828619
```
Barro Colorado data: Smoothed residuals

With interaction (top) and without (integral $-0.11$).
Barro Colorado data: Simulations

simulate(fitbeiAI, nsim=3)
Profile likelihood

So far, we fixed the interaction radius $R$.

It can be estimated by maximising the profile log pseudo-likelihood

$$PPL(R) = \max_{\theta} PL(\theta, R) = PL(\hat{\theta}, R).$$

For the Barro Colorado data,

```r
r <- data.frame(r=seq(1, 10, by=1))
fitbeiProfile <- profilepl(r, AreaInter, bei ~ polynom(x,y,4), aic=FALSE)
yields an optimal value $\hat{R} = 3$.
```
Remarks on model selection

• The likelihood ratio test and AIC rely on the likelihood so do not apply.

• The theory of composite likelihood aka estimating equations provides alternative tools.

(Outside the scope of this course).
Finite marked point processes

To define a finite marked point process $X$ on a bounded window $W$ with marks in $M$ (either $\mathbb{R}$ or $\{1, \ldots, N_M\}$), specify

- a discrete probability distribution $(p_n)_{n \in \mathbb{N}_0}$ for the total number of points;

- a family of symmetric joint probability densities $j_n((x_1, m_1), \ldots, (x_n, m_n))$ for the locations and marks of the points given that there are $n$ of them (with respect to the product of $dx_i d\nu(m_i)$ for some probability density function/mass function $\nu$ on $M$).
Density function

The $p_n$ and $j_n$ may be combined in a single function

$$f(\{(x,m_1), \ldots, (x_n,m_n)\}) =$$

$$e^{-|W|/n!}p_n j_n((x_1,m_1), \ldots, (x_n,m_n))$$

the **density function** of $X$.

As for the unmarked case

$$p_n = \frac{e^{-|W|}}{n!} \int_{W \times M} \cdots \int_{W \times M}$$

$$f(\{(u,l_1), \ldots, (u_n,l_n)\}) du_1 d\nu(l_1) \cdots du_n d\nu(l_n)$$

and $j_n$ is proportional to $f$. 33
Multi-type Poisson process

Let $X$ be a Poisson process on $W$ with intensity $\lambda$ and assign each point a label in $M = \{1, 2\}$ independently with probabilities $\pi_1$ and $\pi_2 = 1 - \pi_1$.

Then

$$p_n = e^{-\lambda|W|}(\lambda|W|)^n/n!$$

and

$$j_n((x_1, m_1), \ldots, (x_n, m_n)) =$$

$$\frac{2^n}{|W|^n} \pi_1^{\#\{m_i : m_i = 1\}} \pi_2^{\#\{m_i : m_i = 2\}}$$

if $\nu(1) = \nu(2) = 1/2$. 
Multi-type Poisson process – marginals

By the Poisson assumptions and the independence of the labels, $N(A_j \times B_j)$, $A_j \subset W$, $B_j \subset \{1, 2\}$, are independent when $A_j \times B_j$ are disjoint.

For $A \subseteq W$, the probability of finding $k$ $m$-labelled points in $A$ is

$$\sum_{n=k}^{\infty} \binom{n}{k} \pi_m^k (1 - \pi_m)^{n-k} = \frac{e^{-\lambda \pi_m |W|}}{k!} (\pi_m \lambda |W|)^k.$$  

**Conclusion:** The points with label $m$ form a Poisson process with intensity $\pi_m \lambda$ and the marginals are independent.
Multi-type Poisson process – simulation

Xlabel <- rmpoispp(c(25,75), types=c("A","B"))
Maximum likelihood estimation

Upon observation of the two components \( x_1 \) and \( x_2 \), the log likelihood function \( L(\theta_1, \theta_2) \) is

\[
L(\theta_1, \theta_2) = n(x_1) \log \theta_1 + n(x_2) \log \theta_2 - (\theta_1 + \theta_2)|W|
\]

where \( \theta_m = \lambda p_m \) are the marginal intensities.

The score equations read

\[
\frac{n(x_m)}{\theta_m} - |W|
\]

so that \( \hat{\theta}_m = n(x_m)/|W| \).

As in the univariate case, the estimators are unbiased and efficient.
Example: Ants data

The locations of nests of two types of ants, Cataglyphis bicolor and Messor wasmanni.
Ants data – maximum likelihood estimates

ppm(ants, ~ marks)

yields

Intensities:

<table>
<thead>
<tr>
<th>beta_Cataglyphis</th>
<th>beta_Messor</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.762949e-05</td>
<td>1.585795e-04</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Estimate</th>
<th>S.E.</th>
<th>CI95.lo</th>
<th>CI95.hi</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>-9.6014665</td>
<td>0.1856953</td>
<td>-9.965423</td>
</tr>
<tr>
<td>marksMessor</td>
<td>0.8522119</td>
<td>0.2217851</td>
<td>0.417521</td>
</tr>
</tbody>
</table>
Heterogeneous multi-type Poisson process

A heterogeneous Poisson model with

$$\log \lambda(x, m) = \mu + \alpha_m + \theta B(x)$$

may be fitted by, e.g.,

\[
\text{ppm(ants, } \sim \text{ marks } + \text{ x, Poisson()})
\]

with default \(\alpha_1 = 0\) (for \(B(\cdot)\) a horizontal trend); the alternative model

$$\log \lambda(x, m) = \mu + \alpha_m + \theta B(x) + \gamma_m B(x)$$

by

\[
\text{ppm(ants, } \sim \text{ marks } * \text{ x, Poisson()})
\]

with default \(\alpha_1 = 0, \gamma_1 = 0\).
Multi-type pairwise interaction models

A multi-type pairwise interaction process $X$ on a bounded window $W$ is a point process whose density function is of the form

$$f(\{(x_1, m_1), \ldots, x_n, m_n)\}) \propto n \prod_{i=1}^{n} \beta_{m_i}(x_i) \prod_{i<j} \gamma_{m_i, m_j}(x_i, x_j)$$

for $m_i \in \{1, \ldots, N_m\}$, $\beta_m : W \to \mathbb{R}^+$ and $\gamma_{m_i, m_j} : W \times W \to \mathbb{R}^+$. The latter function must be symmetric in the sense that

$$\gamma_{m_i, m_j}(x_i, x_j) = \gamma_{m_j, m_i}(x_j, x_i)$$

and models the interaction.
Example: Two-type Strauss process

\[ \gamma_{m,m'}(x, x') = \begin{cases} 
\gamma_{m,m'} & \text{if } ||x - x'|| \leq R_{m,m'} \\
1 & \text{if } ||x - x'|| > R_{m,m'} 
\end{cases} \]

**Note:** The interaction strengths \( \gamma_{m,m'} = \gamma_{m',m} \in [0, 1] \) and ranges \( R_{m,m'} = R_{m',m} \) are type-dependent.
Multi-type Strauss process – simulation

To simulate from a model in which points of different type cannot come closer than $R = 0.06$ to each other, use the script

```r
beta <- c(50, 50)
r <- 0.03 * matrix(c(1,2,2,1), nrow=2, ncol=2)
gamma <- matrix( c(1,0,0,1), nrow=2, ncol=2)

ms <- list( beta=beta, radii=r, gamma=gamma)
mStrauss <- rmhmodel(cif="straussm", par=ms,
                      w=square(1), types=c("A", "B"))
X <- rmh(mStrauss,
            start=list(n.start=0), control=list(nrep=1e6))
```
Multi-type Strauss process – realisation
Maximum pseudo-likelihood estimation

Let $x = \{(x_1, m_1), \ldots, (x_n, m_n)\}$ be a realisation of a finite point process defined by a density function $f(x; \theta)$ that depends on a parameter $\theta$.

The log pseudo-likelihood function is defined as

$$\sum_{i=1}^{n} \log \lambda_\theta(x_i, m_i | x \setminus \{(x_i, m_i)\}) - \int_W \int_M \lambda_\theta(w, l | x) \, dwd\nu(l).$$

Optimise numerically over the parameter $\theta$ to obtain the maximum pseudo-likelihood estimator $\hat{\theta}$. 
Example: Amacrine data

Locations of displaced amacrine cells in the retina of a rabbit: 152 “on” cells and 142 “off” cells in $[0, 1.6] \times [0, 1]$ with unit $\approx 662$ microns.
Amacrine data – model fitting

\[
r <- 0.03 \times \text{matrix}(c(1,2,2,1), \text{nrow}=2, \text{ncol}=2)
\]

\[
\text{fitAmacrine} <- \text{ppm}(\text{amacrine} \sim \text{marks},
\quad \text{MultiStrauss}(r, \text{c("off","on")}))
\]

yields

First order terms:
\[
\begin{array}{ll}
\text{beta_{off}} & 120.1231 \\
\text{beta_{on}} & 122.3144
\end{array}
\]

Fitted interaction parameters \( \gamma_{ij} \)

\[
\begin{array}{ccc}
\text{off} & \text{on} \\
\text{off} & 0.0654809 & 0.9000259 \\
\text{on} & 0.9000259 & 0.0000000
\end{array}
\]

\textbf{Note:} (almost) hard core within types, (near-) independence between types.
Amacrine data – realisations of the fitted model

plot(simulate(fitAmacrine, nsim=3))
References