Linda Stougaard Nielsen *28th March 2001*

A very short introduction to point processes and maximum likelihood estimation

Point pattern

A point pattern is a finite set of points in some subset of $\mathbb{R}^m$. In the following will we only consider point patterns in the unit square $(0,1)^2$. Thus,

$$x = \{\xi_1, \ldots, \xi_n\}, \quad \xi_i \in (0,1)^2$$

is a point pattern in the unit square.

Let Ω denote the set of all finite point patterns in $(0,1)^2$.

Point process

A number can be interpreted as being a realization of some stochastic variable. Likewise a point pattern can be interpreted as being a realization of a stochastic point process. Thus, the point process X on $(0,1)^2$ has state space Ω.

A density of X can depend on the positions of the points in a given realization, or the number of points or the distance between the points and much more. Let f denote the density, then

$$\int_\Omega f(x)dx = 1,$$

i.e. f integrates to 1 over all possible point patterns.

We are interested in parametric classes of models, therefore the density will be described by a parameter $\psi \in \Psi$: $f(x;\psi)$.

A density for a point process can be separated into a part $h(x;\psi)$ depending on the point pattern, and the rest

$$f(x;\psi) = \frac{h(x;\psi)}{\alpha(\psi)} \quad \text{where} \quad \alpha(\psi) = \int_\Omega h(x;\psi)dx. \tag{1}$$

Typically the integral defining $\alpha(\psi)$ is intractable and the density can not be determined explicitly. Therefore simulation methods are used. For a special class of densities, the Markov densities, Markov chain Monte Carlo (MCMC) simulation can be used to draw realizations from the model.

Example 1 (Strauss point processes)

Let d_2 denote the Euclidean distance and let

$$s_R(x) = \sum_{i<j} 1(d_2(\xi_i, \xi_j) < R)$$

be the number of pairs of points in x that lie closer than R to one another. Let $n(x)$ denote the number of points in x.

The Strauss point process has density proportional to

$$h(x;\beta,\gamma,R)=\beta^{n(x)}\gamma^{s_R(x)}.$$

The Strauss point process is a Markov point process.

Maximum likelihood estimation (MLE)
Given an observed point pattern x and given a point process model with density $f(\cdot;\psi)$, $\psi\in\Psi$, we want to find the value of ψ which makes it most likely that x is a realization from the point process model. Using MLE, this is the value which maximizes the likelihood function $l(\psi;x)=\log f(x;\psi)$ with respect to ψ. This is done by solving the likelihood equation $\frac{d}{d\psi}l(\psi;x)=0$.

MLE for exponential families
An exponential family density is of the form

$$f(x;\psi)=a(\psi)b(x)\mathrm{e}^{\phi(\psi)\cdot t(x)},$$

where $t(x)$ is called the canonical observator. It can be shown, that the likelihood equation for an exponential family takes the form

$$\mathbb{E}_\psi t(X)=t(x).$$

Due to the way the parameter R enters into the density, the Strauss process is **not** an exponential family unless R is fixed. For fixed R, the Strauss model is an exponential family with

$$\psi=(\beta,\gamma),\ a=\alpha,\ b=1,\ \phi(\beta,\gamma)=(\log\beta,\log\gamma)\text{ and }t(x)=(n(x),s_R(x)).$$

Thus, the likelihood equations are

$$\mathbb{E}_{\beta,\gamma}n(X)=n(x)\quad\text{and}\quad\mathbb{E}_{\beta,\gamma}s_R(X)=s_R(x).\tag{2}$$

The left hand-sides can of course not be calculated, so they are instead approximated by

$$\mathbb{E}_{\beta,\gamma}n(X)\approx\frac{1}{m}\sum_{i=1}^{m}n(x_i)\quad\text{and}\quad\mathbb{E}_{\beta,\gamma}s_R(X)\approx\frac{1}{m}\sum_{i=1}^{m}s_R(x_i)\tag{3}$$

where $x_1,\ldots,x_m$ are realizations from the model with parameter γ,β and R.

Profile likelihood MLE
Suppose that $\psi=(\psi_1,\psi_2)$ and that the density in question is an exponential family model for ψ_2 fixed. In this case we will estimate ψ_1 for fixed ψ_2

using the exponential family theory and then maximize w.r.t. ψ_2. Thus, consider the partially maximized likelihood function a.k.a. the profile likelihood function

$$\bar{l}(\psi_2; x) = \max_{\psi_1 \in \Psi_1} l(\psi_1, \psi_2; x),$$

then ψ is estimated by maximizing this function.

In practice we calculate $\bar{l}(\psi_2)/\bar{l}(\psi_{2,0})$ for a grid of ψ_2 values. In this note I will not explain how this is done in detail, see Geyer (1999, 114–116).

Problem formulation for parallel programming

Goal

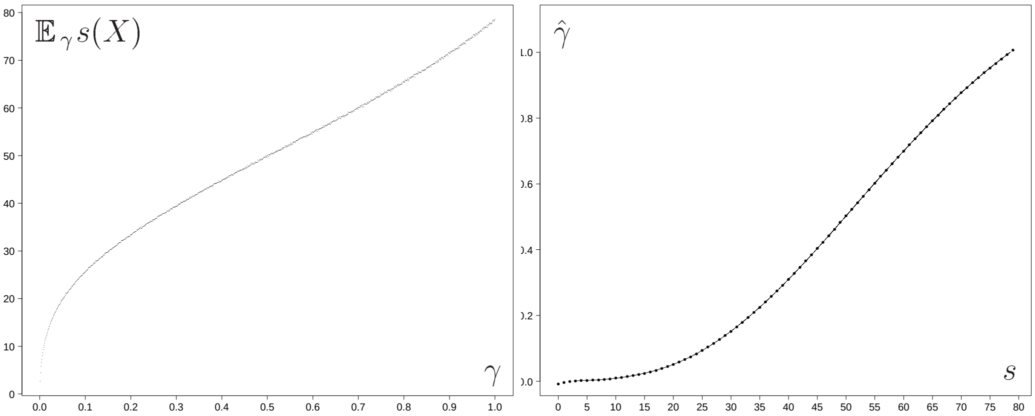
We will only consider the Strauss process and are interested in making a table

$$(\beta, \gamma, R) \rightarrow (\mathbb{E}_{\beta,\gamma} n(X), \mathbb{E}_{\beta,\gamma} s_R(X)), \tag{4}$$

such that given a point pattern and an R, the MLE can be found quickly by the inverse mapping. Notice, that not knowing the value of R, we get a whole set of MLE's, one for each value of R,

$$\{(\hat{\beta}(R), \hat{\gamma}(R), R) : R \geq 0\}. \tag{5}$$

As discussed in previous section, the final MLE is found by profile likelihood. This is however very complicated so we will at time being only concentrate on tabulating (4) and thereby finding (5).



(a) Simulated values

(b) Predicted values

Figure 1: The model is the Strauss process on the interval $(0; 2\pi)$ with circular neighbour relation (the point 0 equals the point 2π). $n = 100$ and $R = 0.05$ are fixed. For a dense grid of γ-values, $\sum s(x_i)/m$ have been calculated by simulation (a). The predicted values of γ have been calculated for each value of $s \in \mathbb{N}$ (b) on basis of the inverse of (a).

In Figure 1 the procedure have been illustrated in a very simple example where the number of points is fixed (β then falls away) and R is fixed. We therefore have only one parameter γ. In the left hand-side plot $\mathbb{E}_\gamma s(X)$ have been calculated for a very dense grid of $\gamma \in (0;1]$. The inverse mapping have been estimated using standard statistical methods (fitted by a 12th degree polynomium) and for each value of $s(x) \in \mathbb{N}$ we then have the MLE $\hat{\gamma}$.

Programming

We will tabulate the mapping (4) for a large grid of (β, γ, R) of dimension $n_\beta \times n_\gamma \times n_R$, say. For each grid value, a large number m of realizations have to be drawn from the Strauss process with the given parameters, c.f. (3). In order to draw one realization, we have to run a large number n_{burn} of steps in a MCMC algorithm.

Hence, there are three levels of iterations to be done. It does not make sense to parallelize the n_{burn} steps of the MCMC algorithm since these steps are very dependent on each other and use all the available information. The m realizations have to be independent of each other and therefore it would be evident to parallelize this step. However, for MCMC algorithms, the time consuming part is for the chain to reach equilibrium (after the n_{burn} steps). Once equilibrium has been reached, the chain can be run a little longer n_{spac} and draw a new realization which of course is dependent of the first drawn realization, but not very much. (I have used $n_{\text{burn}} = 50000$, $n_{\text{spac}} = 1000$ and $m = 2000$, so instead of $50000 \cdot 2000 = 10^8$ iterations this method only uses $50000 + 2000 \cdot 1000 = 2.05 \cdot 10^6$ iterations).

The last level of iterations is the (β, γ, R) grid, and here the operations between grid values are totally independent since the parameters of the model are changed. So this part can with great advantage be made parallel (embarrassing parallel).

We have to take into account that the actual value of (β, γ, R) have a large impact on the run-time, since a larger β value gives a higher number of points in the realizations and therefore each step of the Markov chain will take longer time. Similarly, large R values give a smaller number of points. Therefore the CPU's have to work independently of each other, and we have to reserve one CPU to keep track of which grid values that has been calculated and tell the other CPU's which is the next non-treated grid value. Alternatively we can try to split the grid up in portions of similar time consumption, but this might be very difficult, in particular because we work with randomness.

In Figure 1 (a) I used $n_\gamma = 1000$. Such a fine grid is not necessary and the grid does not need to be regular. Notice that a fine grid was needed for γ close to 0, whereas for all other values a coarser grid could have been used.

Vizualisation
Suppose that we have the table (4). In applications we have observed a point pattern and want to find the MLE's under the Strauss model. Then we calculate the canonical statistics $n = n(x)$ and $s(R) = s_R(x)$. Notice that the first one is constant for all R. We now plot the surface $(\beta, \gamma) \to \mathbb{E}_{\beta,\gamma} n(X)$ based on the values from the table, and intersect this by the constant plane n. The intersecting set (curves) is then possible MLE's. We do the same for $\mathbb{E}_{\beta,\gamma} s_R(X)$, but here both the surface and the intersecting constant plane $s(R)$ changes in R. This could be made as an animation with R changing in time. Then by intersecting the "n" curves with the moving "s" curves we get the set (5). See Figure 2.

Figure 2: Moving surface from simulations and intersecting constant plane from the given observation gives a solution set for the MLE.

References

Geyer, C. J. (1999). Likelihood inference for spatial point processes. In Barndorff-Nielsen, O. E., Kendall, W. S., and van Lieshout, M. N. M., editors, *Stochastic Geometry: Likelihood and Computation*, pages 79–140, London. Chapman and Hall/CRC.