Bayesian estimation of point process intensity using PyMC3

Mathias Weisse

July 1, 2020

After a brief introduction to point processes, a bayesian approach for estimating the intensity of a (nonhomogeneous) Poisson point process is proposed and demonstrated.

1 Point Processes

1.1 Definitions

A point process can be thought of as a random (and at most countable) set of points S_i lying in some set $S \subseteq \mathbb{R}^d$. If we then fix an arbitrary $B \subseteq S$, we can construct a random variable $N(B) := \#\{S_i : S_i \in B\}$ that counts the number of points lying in B and it is straightforward to see that N is a measure in the mathematical sense. A mathematically convenient definition is:

> **Let $(\Omega, \mathscr{F}, P)$ be a probability space and $\mathbb{M}$ a set of counting measures on $S \subseteq \mathbb{R}^d$. A random variable $N : \Omega \to \mathbb{M}$ taking values in $\mathbb{M}$ is then called a point process or random counting measure.**

Often we are interested in the expected number of points that lie in a subset B which is given by $\mu(B) := E[N(B)]$. Using the theorem of monotone convergence we can again check that μ is a measure.

> **The measure μ, defined by $\mu(B) := E[N(B)]$, $B \subseteq \mathbb{R}^d$ is called the intensity measure of N. If μ is absolutely continuous, i.e. of the form $\mu(B) = \int_B \lambda(x)dy$, with an integrable $\lambda(x) \geq 0$, then λ is called the intensity of N.**

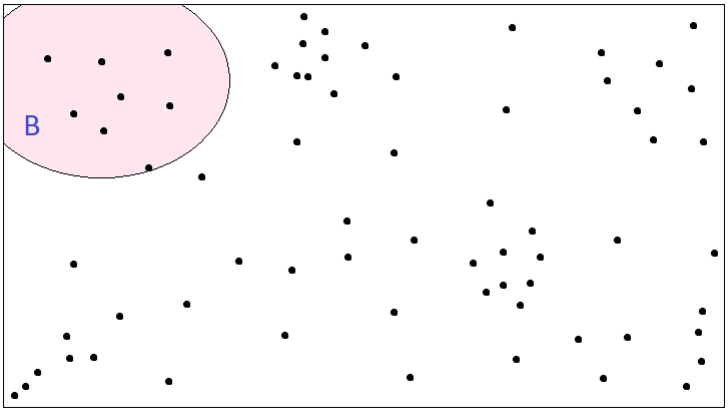

Figure 1: Example of a point process realization. Here, $N(B)$ =8.

The simplest and by far best understood kind of point process is the Poisson point process which models $N(B)$ as a Poisson distributed random variable. The Poisson point process is often understood as representing 'spatial randomness':

> **A Poisson point process with intensity measure μ is a point process N with:**
>
> · $N(B) \sim Poi(\mu(B))$**, i.e.:** $P(N(B) = k) = \frac{\mu(B)^k}{k!} e^{-\mu(B)}$**.**
>
> · $N(B_1), ..., N(B_n)$ **are stochastically independent for disjunct** $B_1, ..., B_n$**.**

1.2 Simulation of Poisson Point Processes

A Poisson point process with intensity measure μ on a set $S \subseteq \mathbb{R}$ can be constructed in the following canonical way: let $(S_i)_{i\geq 1}$ be a sequence of i.i.d. random vectors in S with distribution $\frac{\mu(\cdot)}{\mu(S)}$ and $N(S) \sim Poi(\mu(B))$ a Poisson distributed random variable. Then the counting measure given by:

$$N(B) := \sum_{i=1}^{N(S)} \delta_{S_i}(B)$$

where δ_x is the Dirac measure, is a Poisson process with intensity measure μ. Now, simulation turns out to be easy if:

1) S is a cuboid

2) μ is just a multiple of the Lebesgue measure.

Because then $\frac{\mu(\cdot)}{\mu(S)}$ is just the uniform distribution on S and the components S_i^j $j = 1, ..., d$ are independent and uniformly distributed on the respective axis of the cuboid S.

In case we want to simulate an inhomogeneous Poisson point process with intensity $\lambda(x)$, we may proceed by dependent thinning:

1) calculate a bound b on λ: $b \geq \lambda(x)$

2) simulate a point process sample $S_1, ..., S_{N(S)}$ according to a homogeneous Poisson point process with intensity b

3) accept point S_i with probability $\lambda(S_i)/b$.

The remaining points will then constitute a point process sample of the original process[4].

2 Bayesian modelling

If we only consider processes with absolutely continuous intensity measure then a Poisson point process is fully determined by its intensity function. Therefore, we may consider a Poisson point process to be parameterized with one parameter $\lambda \in \Lambda$ which is function-valued. In the following, we will consider intensity functions of the form $\lambda(x) = e^{Y(x)}$, where Y is generated from a Gaussian process.

2.1 Gaussian processes

A stochastic process Y defined on S is called a Gaussian process if for every finite set $x_1, ..., x_n \in S$ the random vector $(Y(x_1), ..., Y(x_n))$ is a multivariate Gaussian. If $E[Y(x)] \equiv const.$ and $Cov[Y(x), Y(x')] = c(x - x')$ the process is said to be stationary.

Examples of popular covariance functions are: the Squared exponential covariance function $c(x - x') := \sigma^2 e^{-|x-x'|^2/(2\beta^2)}$ or the Periodic covariance

function $c(x - x') := e^{-2sin^2(|x-x'|/2)/\beta^2}$. Here, σ^2 is the variance of the process, while β controls the correlation of the process between two points, which increases with β. Hence, realizations of the process tend to be smoother, if β is larger (see figure 2).

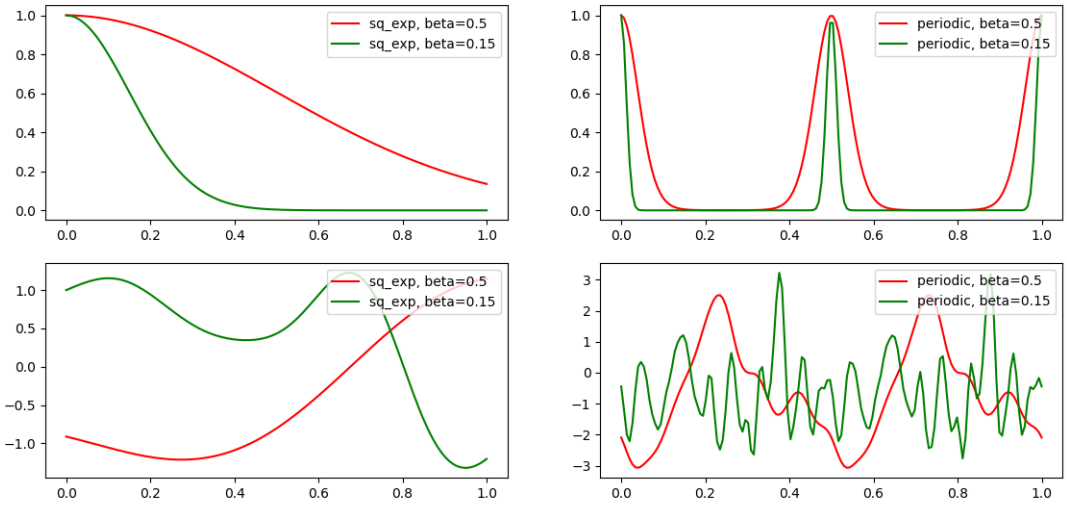

Figure 2: Upper row: Squared exponential- and Periodic covariance functions with $\sigma = 1$. Lower row: sample realizations of the respective processes.

2.2 Model description

From now on, let $S = [0, 1]$. The available data consists of a point process sample $D = \{x_1, ..., x_{N(S)}\} \subseteq S$. This data is transformed to count data $D' = \{k_1, ..., k_m\}$ simply by binning D into m subintervals $B_i := [(i-1)/m, i/m)$ for $i = 1, ..., m-1$ and $B_m := [(m-1)/m, 1]$. Considering the definition of the Poisson point process, we get the likelihood:

$$p(D' \mid Y) = e^{-\sum_{i=1}^{m} \mu(B_i)} \prod_{i=1}^{m} \frac{\mu(B_i)^{k_i}}{k_i!}, \text{ with:}$$

$$\mu(B_i) := \int_{B_i} e^{Y(x)} dx.$$

Instead of putting a Gaussian process prior with covariance function $c(x, x')$ and mean $m(x)$ on Y directly, we choose to approximate Y by $Y'(x) :=$

$\sum_{i=1}^{m} y_i \chi_i(x)$ where $y = (y_1, ..., y_m)$ is drawn from a multivariate Gaussian distribution with covariance matrix $C = (c(\frac{i-1+i}{2m}, \frac{j-1+j}{2m}))_{i,j=1...m}$ and mean $m = (m(\frac{i-1+i}{2m}))_{i=1...m}$ (a similar approach can be found in [1]). This is motivated by the defining property of Gaussian processes. To be more specific, we choose Y to be stationary with mean function $m(x) \equiv 0$. $\mu(B_i)$ then becomes $\frac{e^{y_i}}{m}$.

Furthermore, $\chi_i(x)$ is defined by:

$$\chi_i(x) = \begin{cases} 1, x \in B_i \\ 0, \text{ else} \end{cases}.$$

For the parameter σ^2 of the covariance function, a Half-normal hyperprior with variance 5 has proven to be effective for point process data on $S = [0, 1]$. In contrast, the parameter β is chosen by a data driven approach as described below. The full hierarchical model is:

$$\begin{aligned} &\beta \\ &\sigma^2 \sim \ Half\mathcal{N}(5) \\ &y \sim \mathcal{N}(0, C_{\sigma^2,\beta}) \\ &k_i \sim Poi(\frac{e^{y_i}}{m}). \end{aligned}$$

2.3 Selection of β

Finding an appropriate hyperprior for β proves to be a difficult task. Instead, the following data driven approach is applied here: evaluate the model (i.e. calculate the posterior) on a grid of β's $(\beta_1, ..., \beta_n)$ using only a small number of bins then choose the β (model) that minimizes the WAIC (widely applicable information criterion) and finally fit a model with a larger number of bins. Note that β is inherent to the underlying Gaussian process and thus independent of the bin size. Hence, bin size only indirectly biases β through the approximation of the intensity function. Hoever, figure 4 gives anecdotal evidence that this effect might be neglectable in practice.

Roughly speaking, the WAIC is the negative log-pointwise-predictive-density (lppd):

$$-\sum_{i=1}^{m} log(\int p(k_i \mid \theta) p^{\beta}_{post}(\theta) d\theta),$$

where $\theta = (y, \sigma^2)$ is the parameter vector and p^{β}_{post} the posterior distribution, adjusted by a correction term that accounts for model complexity as well as for the bias introduced by the fact that the lppd is estimated with the same data the model was fitted to. For more information see [5].

2.4 Practical Inference

The model is of course not analytically tractable. Therefore, inference was done using PyMC3 which uses NUTS (No-U-Turn-Sampler), a variant of the Hamiltonian Monte Carlo algorithm (HMC) to generate samples from the posterior distribution. For more information on HMC, see [2].
Implementation of the model itself is fairly easy using PyMC3:

```python
import pymc3 as pm
import numpy as np

def intensityLogGauss(sample,bins,beta):
    """
    sample: 1d-numpy array representing point process sample
    bins: number of bins
    beta: length parameter of gaussian process prior
    """
    width=1/bins
    data,edges=np.histogram(sample,bins=bins,range=(0,1))
    distMat=width*np.array([[np.abs(i-j) for i in range(bins)] for j in range(bins)])

    model=pm.Model()
    with model:
        #building the model
        beta=beta
        sigmaSq=pm.HalfNormal('sigmaSq',5)
        chol=np.sqrt(sigmaSq)*cholesky(pm.math.exp(-1.*distMat**2/(2.*beta**2))+1e-6*np.eye(bins))
        y=pm.Normal('gaussfield',mu=0,sigma=1,shape=bins)
        lam=pm.Deterministic('intensity',width*pm.math.exp(pm.math.dot(chol,y)))
        k=pm.Poisson('points',mu=lam,observed=data)
        #sampling
        trace=pm.sample(draws=1000, tune=500)

    return trace
```

Note, that the multivariate Gaussian is implemented as a non-centered parameterization (see [3]).

3 Example

We will now have a look on the models capability of recovering the intensity function from a simulated point process sample. The intensity function used in this example is:

$$\lambda(x) = 50e^{sin(2\pi x)}$$

and the covariance of the underlying Gaussian process prior is modelled with the Squared exponential covariance function. In each MCMC-simulation, 1500 samples were drawn from the posterior distribution but the first 500 were only used for tuning.

3.1 β-selection

Figure 3 shows a 20 bins-model for four different values for β. The red dots are the posterior (sample-) mean, the red lines indicate a 90%-confidence interval and the blue line is the true intensity function. The point process

sample is depicted on the x-axes. The respective WAIC-values are (top to bottom-left to right): 86,62; 77,02; 76,32; 80,56. Hence, the WAIC chooses β =0,35 in this case which is a visually quite pleasing choice as well.

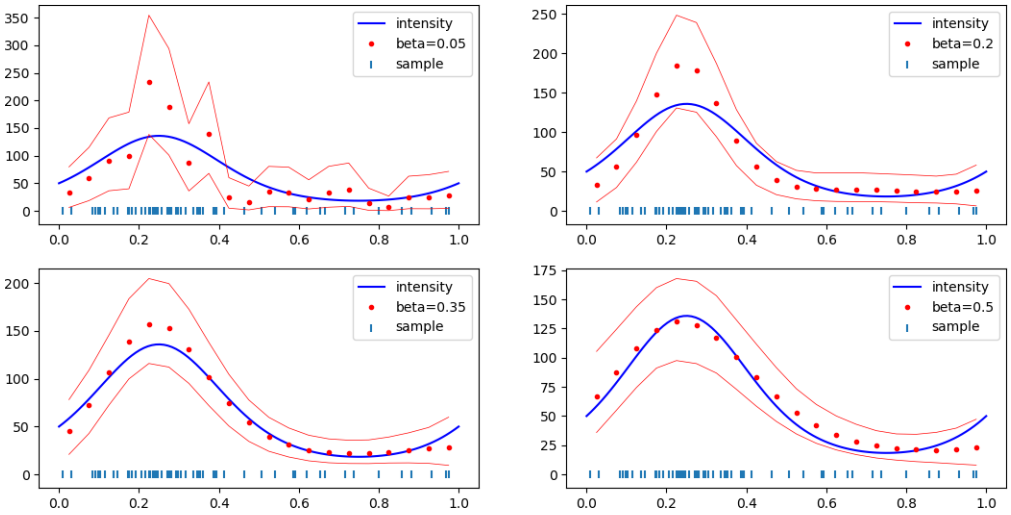

Figure 3: Model evaluated with four different values for β

Figure 4 shows the optimal β chosen from ten potential values (ranging from 0,05 to 0,5; linearly spaced) by the WAIC. The figure suggests that the number of bins does not affect the the choice of β too much.

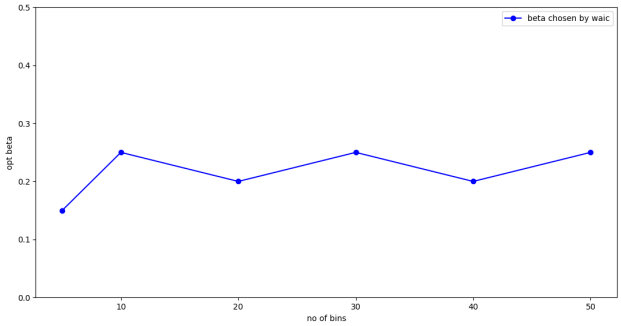

Figure 4: Optimal β for varying number of bins according to WAIC.

3.2 Final model

Figure 5 shows a 40-bins model. Here, β was chosen from the same grid as above, using a model with 7 bins. Again, the choice of β seems to transfer quite well to the more complex model.

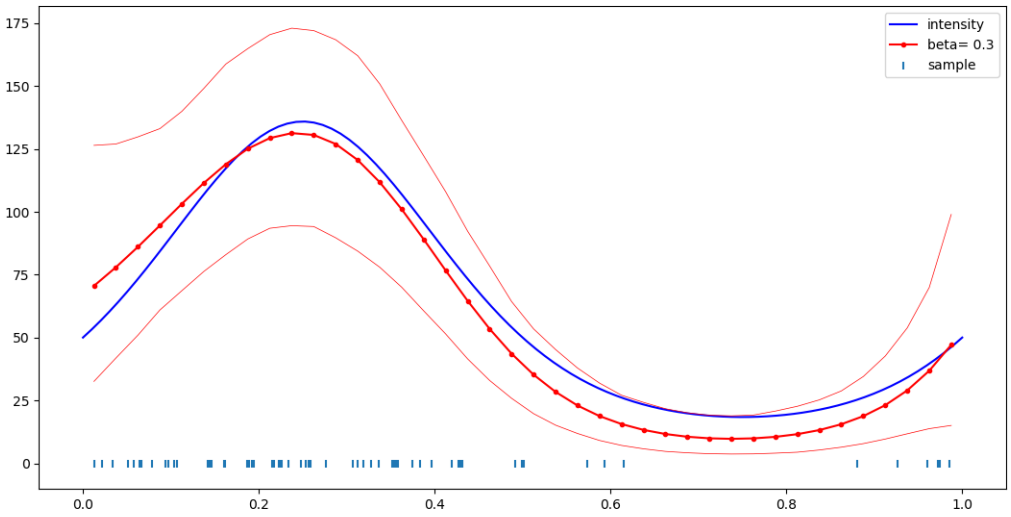

Figure 5: 40-bins model.

4 Outlook

The next step is to generalize the model to the two dimensional (spatial) case. This means that the number of bins will roughly square, resulting in an increase in computational complexity. Hence, a more sophisticated approach for the selection of β and appropriate subdivision of the domain S into bins is required.

Potential applications of spatial point processes include forestry, epidemiology, risk assessment of natural hazards or real estate market-analysis.

References

[1] Moller J. et al, *Log Gaussian Cox processes*, (1998), http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.71.6732&rep=rep1&type=pdf

[2] Betancourt M., *A Conceptual Introduction to Hamiltonian Monte Carlo*, (2018), https://arxiv.org/pdf/1701.02434.pdf

[3] Wiecki T., *Why hierarchical models are awesome, tricky, and Bayesian*, (2017), https://twiecki.io/blog/2017/02/08/bayesian-hierchical-non-centered/

[4] Chen Y., *Thinning Algorithms for Simulating Point Processes*, (2016), https://www.math.fsu.edu/ ychen/research/Thinning%20algorithm.pdf

[5] Gelman A. et al, *Understanding predictive information criteria for Bayesian models*, (2013), http://www.stat.columbia.edu/ gelman/research/published/waic_understand3.pdf