IMPERIAL

Comparison between two methods of 1D Poisson process simulation

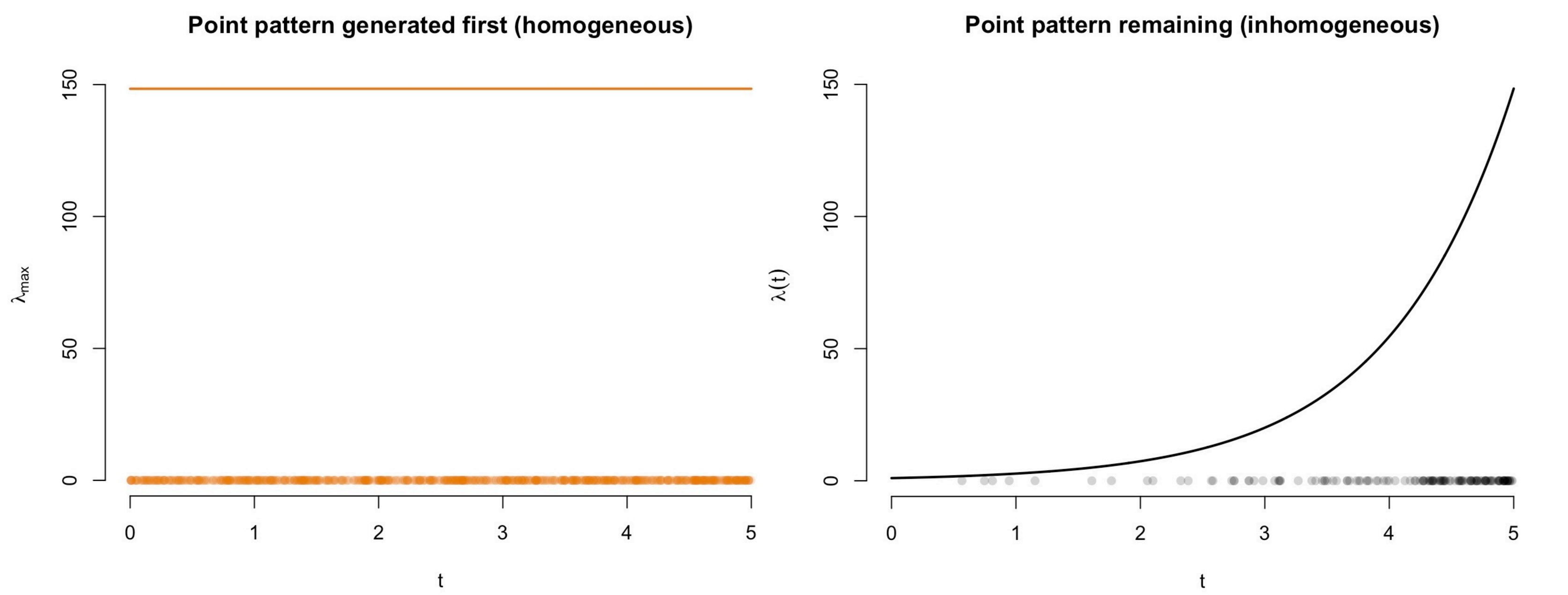
Richard Wu CID: 02381510

Introduction

A Poisson process is a random set of points that follows the two properties: Poisson distribution of point counts and independent scattering[1]. In 1D case, it happens on the real line labelled t, time. Note points can be considered as arrivals for realistic meanning.
A Poisson process takes place in the interval $[t_{min}, t_{max}]$ and has a non-negative intensity function $\lambda(t)$, representing the mean number of points in a set of unit length, showing how dense the points are. For homogeneous Poisson process, $\lambda(t)$ = λ, a constant.
This poster presents and compares two algorithms for simulating an inhomogeneous Poisson process, that is, Poisson process with a non-constant $\lambda(t)$.

Random thinning method

The idea behind random thinning is to generate a stationary process of potential arrivals at the maximum rate, then randomly delete or 'thin' some of the potential arrivals[2].
For a homogeneous Poisson process with intensity λ, let T_i be the time interval between the i th and $i+1$ th event (i.e. $T_i = t_{i+1} - t_i$ for $i \geq 0$, set $t_0 = t_{min}$ for consistency), then we have $T_i \sim exp(\lambda)$[3]. In this way, a sequence of points is generated and that's the pattern.
When simulating an inhomogeneous case, a homogeneous Poisson process with intensity λ_{max} equals the maximum of the function is generated. Then, the probability of keeping the point t is $\frac{\lambda(t)}{\lambda_{max}}$. The points kept give the required inhomogeneous Poisson process.
The following example with $\lambda(t) = e^t$ in the range $[0, 5]$ shows that if λ_{max} is much higher than other parts of the function, loads of points are generated but most of them are deleted. Only a small proportion of them still remains. This is kind of inefficient.



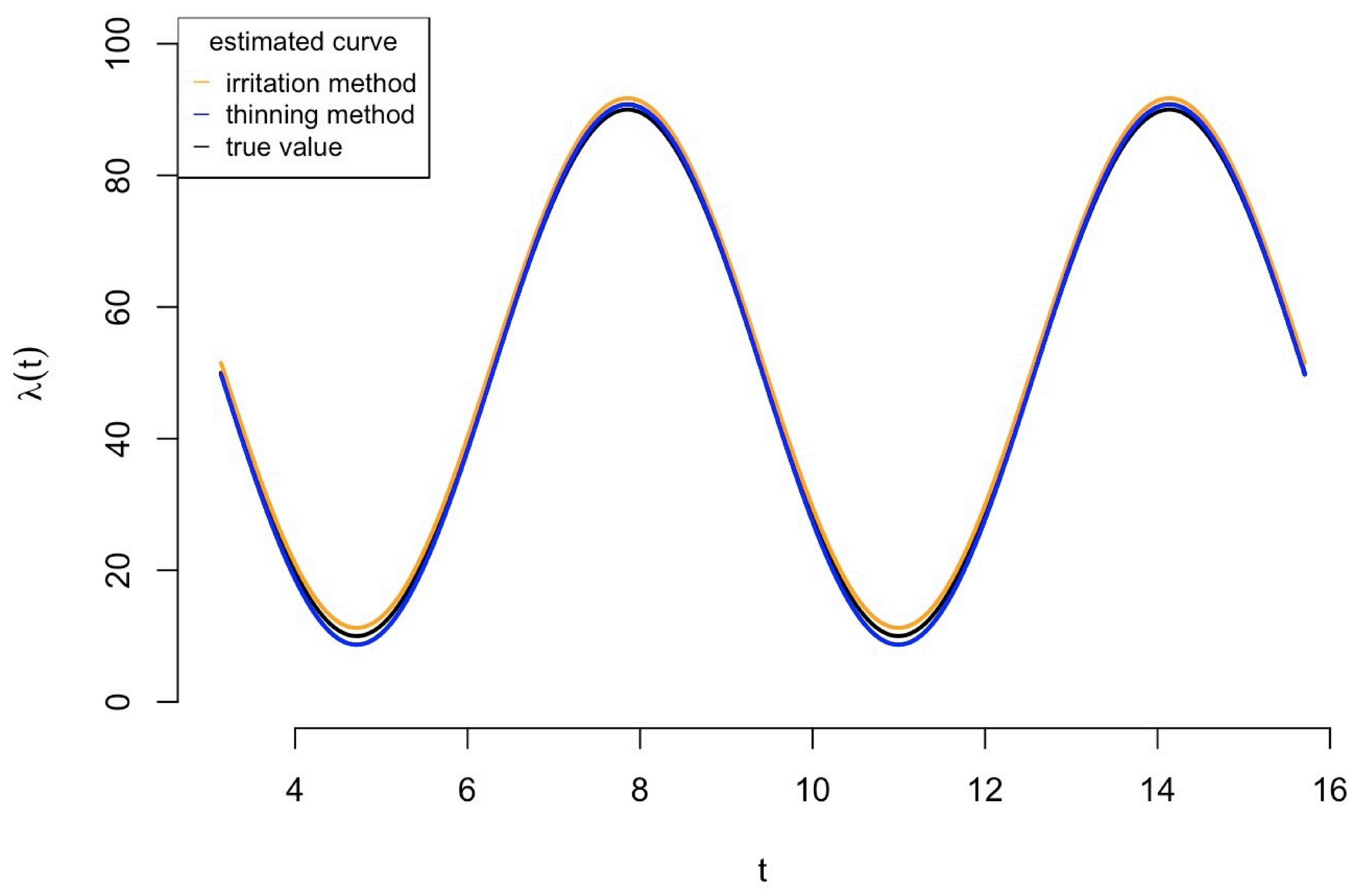
Irritation attempt

Here, we try to simulate inhomogeneous Poisson process in a similar way as we simulate homogeneous cases. $\lambda(t_i)$ is used as an approximation of the rate of exponential distribution in a short time after t_i happens. Hence, we have $t_{i+1} = t_i + T_i$ for $i \geq 0$, where $T_i \sim exp(\lambda(t_i))$. In this way, we are able to generate a series of point events through irritation, and no need to delete any of those. This is our final point pattern.

Comparison

Case 1

We first test a bounded case, with $\lambda(t)$ not that close to 0. We set $\lambda(t) = \theta_1 sin(t) + \theta_2$, with $\theta_1 = 40$ and $\theta_2 = 50$. Two point patterns are generated using the two methods above.
To verify whether the point patterns generated are good realization of $\lambda(t)$, maximum likelihood estimation is used to find θ_1 and θ_2 respectively, using the two point patterns. Then, we sketch $\lambda(t) = \theta_1 sin(t) + \theta_2$ and the result is as follow.



Finally, we use t-test with 100 samples to show we do not reject that θ_1 and θ_2 estimated has an average 40 and 50 respectively, using both methods. This means both methods are satisfying for now, but note runtime of the thinning method is twice that of the irritation method.

Case 2

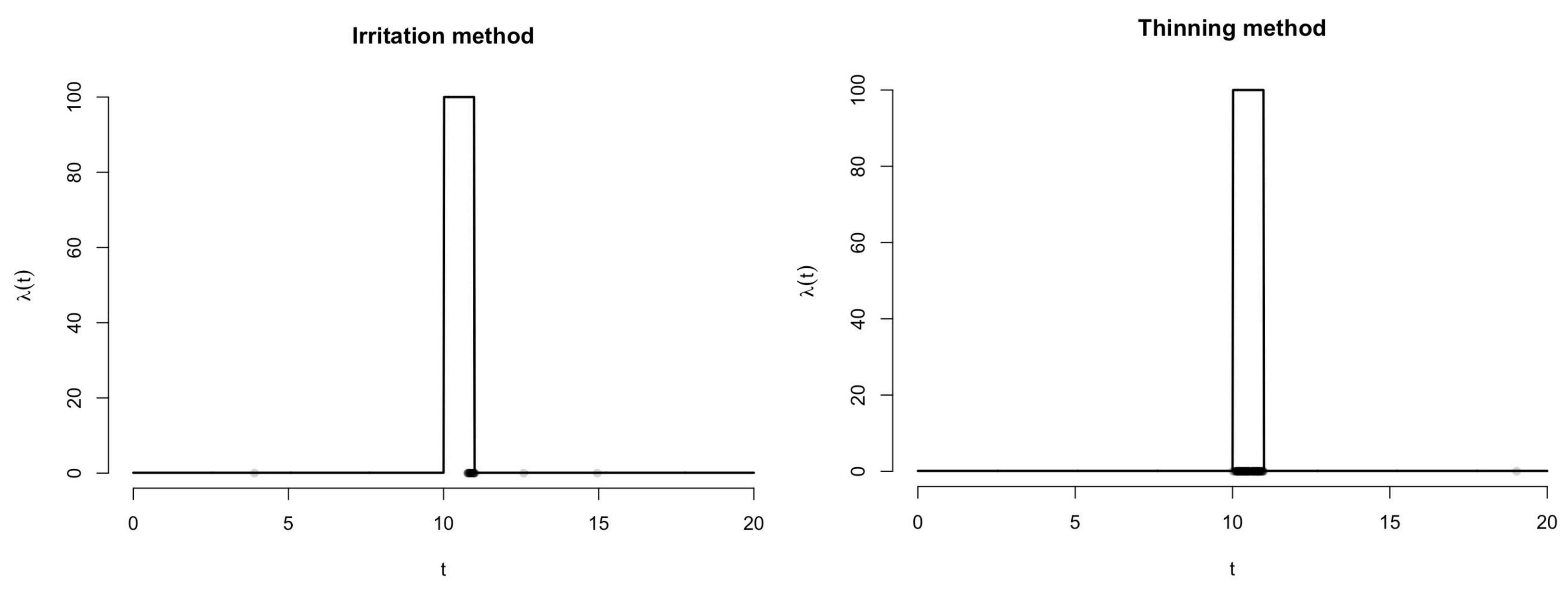
Both method cannot handle function with a vertical asymptote (e.g. $\lambda(t) = \frac{1}{t}$), because irritation diverges and we cannot find a λ_{max} for thinning. However, we can still change t_{min} and t_{max} so that we get a bounded function(e.g. set $\lambda_{min} = 0.01$ in the above example).

Case 3

Consider the following function, with $\lambda(t)$ small except in $(10, 11)$:

$$\lambda(t) = \begin{cases} 100 \; 10 < t < 11 \\ 0.1 \; otherwise \end{cases}$$

Here are the point patterns generated using two methods.



This time, irritation method is not good enough. With a small $\lambda(t_i)$, T_i is likely to be very large, which means the next point we generate may already skip the part with high intensity!

Conclusion

To conclude, sometimes only the thinning method can give a correct point pattern. However, when both methods work, the irritation method is technically more efficient.

References

[1] Sung Nok Chiu et al. Stochastic geometry and its applications. John Wiley & Sons, 2013.

[2] Barry Nelson et al. Foundations and methods of stochastic simulation. Springer, 2021.

[3] Zak Varty. Exploring Stochastic Point Processes. 2024. URL: `https://m1r.zakvarty.com/`.