Report of simulation

Student ID: 516030910572
Name: Ding Yaoyao

This is the simulation report of multi-dimensional hawkes process . And the report contains three parts: algorithm, result and reference.

Algorithm

In this experiment, thinning algorithm is implemented. This algorithm is proposed by Ogata[1] in 1981. Here is the pseudo of the algorithm:

Algorithm 1: Simulation of an M-variate Hawkes Process with Exponential Kernels $\gamma_{mn}(u) = \alpha_{mn}e^{-\beta_{mn}u}$ for $m, n = 1, 2, \ldots, M$, on $[0, T]$.

Input: $\boldsymbol{\mu}_{M\times 1}$, $\boldsymbol{\alpha}_{M\times M}$, $\boldsymbol{\beta}_{M\times M}$, T

Initialize $\mathcal{T}^1 = \cdots = \mathcal{T}^M = \emptyset$, $n^1 = \cdots = n^M = 0$, $s = 0$;

while $s < T$ **do**

- Set $\bar{\lambda} = \sum_{m=1}^{M} \lambda^m(s^+) = \sum_{m=1}^{M} \left(\mu_m + \sum_{n=1}^{M} \sum_{\tau \in \mathcal{T}^n} \alpha_{mn} e^{-\beta_{mn}(s-\tau)}\right)$;
- Generate $u \sim \texttt{uniform(0,1)}$;
- Let $w = -\ln u/\bar{\lambda}$; // so that $w \sim \texttt{exponential}(\bar{\lambda})$
- Set $s = s + w$; // so that s is the next candidate point
- Generate $D \sim \texttt{uniform(0,1)}$;
- // accepting with probability $\sum_{m=1}^{M} \lambda^m(s)/\bar{\lambda}$
- **if** $D\bar{\lambda} \leq \sum_{m=1}^{M} \lambda^m(s) = \sum_{m=1}^{M} \left(\mu_m + \sum_{n=1}^{M} \sum_{\tau \in \mathcal{T}^n} \alpha_{mn} e^{-\beta_{mn}(s-\tau)}\right)$ **then**
  - $k = 1$;
  - // searching for the first k such that $D\bar{\lambda} \leq \sum_{m=1}^{k} \lambda^m(s)$
  - **while** $D\bar{\lambda} > \sum_{m=1}^{k} \lambda^m(s)$ **do**
    - $k = k + 1$
  - **end**
  - $n^k = n^k + 1$; // updating the number of points in dimension k
  - $t_{n^k}^k = s$; // naming it $t_{n^k}^k$
  - $\mathcal{T}^k = \mathcal{T}^k \bigcup \{t_{n^k}^k\}$; // adding $t_{n^k}^k$ to the ordered set $\mathcal{T}^k$
- **end**

end

if $t_{n^k}^k \leq T$ **then**

- **return** $\mathcal{T}^m$ *for* $m = 1, 2, \ldots, n$;

else

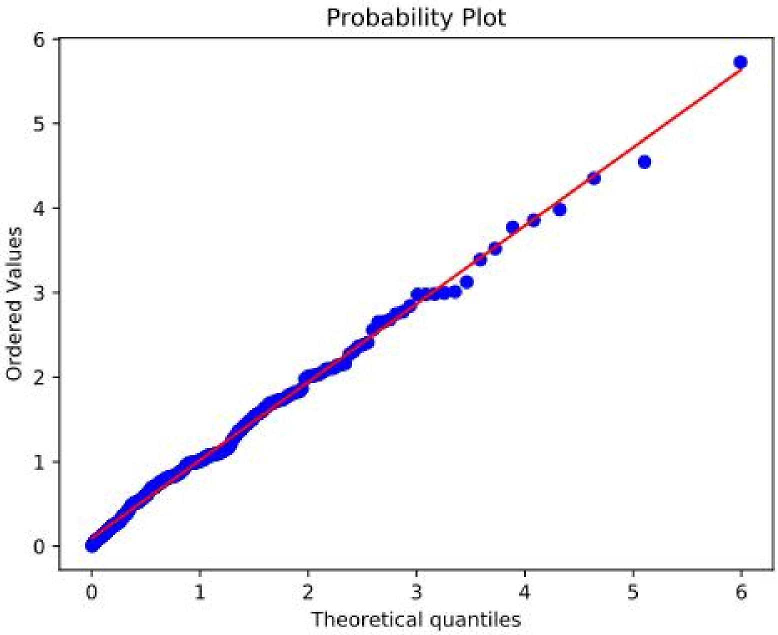
- **return** $\mathcal{T}^1, \ldots, \mathcal{T}^k \backslash \{t_{n^k}^k\}, \ldots, \mathcal{T}^M$;

end

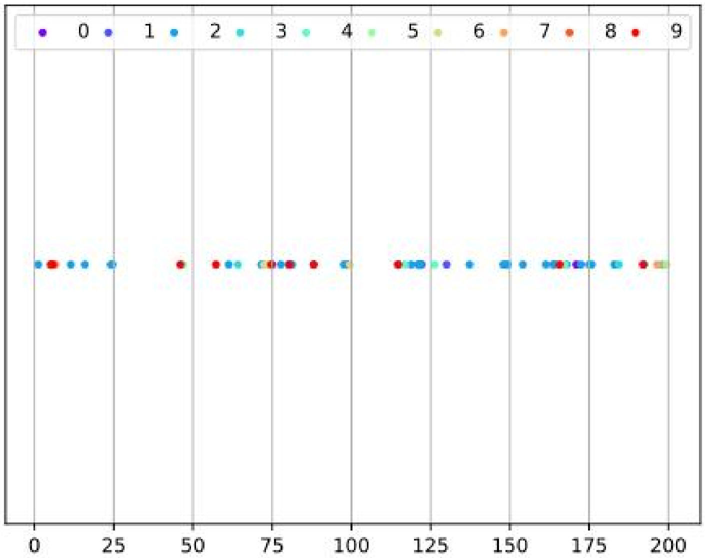
There are two program files main.py and simulation.py. The above algorithm is implemented in simulation.py. And you should run `python main.py` to launch the program. The parameters are stored in parameters.json.

Result

The integral of intensity function between two consecutive events time follows the exponential distribution. And we use the quantile plot to show the similarity of the two distributions:



Almost all the blue points are on the red line, which shows the two distributions are similar.
Here is some events in the simulated data:



Reference

[1] The slides of Point Process course.

[2] The four articles in https://www.math.fsu.edu/~ychen/hawkes.html