Point Process Homework1

Description

Thinning algorithm is a prominent method for simulating point process. In order to improve the understanding of the thinning algorithm and point process pattern, this homework is to implement thinning algorithm for multi-dimensional hawkes process simulation.

Background

Thinning Algorithm

Thinning algorithm simulates homogeneous Poisson processes with high intensities and then thin out the points with conditional intensity function.

Algorithm 3: (Ogata, 1981, p.25, Algorithm 2) Simulation of a Univariate Hawkes Poisson with Exponential Kernel $\gamma(u) = \alpha e^{-\beta u}$, on $[0, T]$.

Input: μ, α, β, T
Initialize $\mathcal{T} = \emptyset$, $s = 0$, $n = 0$;
while $s < T$ **do**
    Set $\bar{\lambda} = \lambda(s^+) = \mu + \sum_{\tau \in \mathcal{T}} \alpha e^{-\beta(s-\tau)}$;
    Generate $u \sim$ `uniform(0,1)`;
    Let $w = -\ln u / \bar{\lambda}$; `// so that` $w \sim$ `exponential(`$\bar{\lambda}$`)`
    Set $s = s + w$; `// so that` s `is the next candidate point`
    Generate $D \sim$ `uniform(0,1)`;
    if $D\bar{\lambda} \leq \lambda(s) = \mu + \sum_{\tau \in \mathcal{T}} \alpha e^{-\beta(s-\tau)}$ **then** `// accepting with prob.` $\lambda(s)/\bar{\lambda}$
        $n = n + 1$; `// updating the number of points accepted`
        $t_n = s$; `// naming it` t_n
        $\mathcal{T} = \mathcal{T} \bigcup \{t_n\}$; `// adding` t_n `to the ordered set` $\mathcal{T}$
    end
end
if $t_n \leq T$ **then**
    return $\{t_k\}_{k=1,2,\ldots,n}$
else
    return $\{t_k\}_{k=1,2,\ldots,n-1}$
end

Multi-dimensional Hawkes Process

Multi-dimensional hawkes process is a extention of one-dimensional hawkes process. In addition to considering the event incentives in each dimension, the multi-dimensional Hawkes process also considers the impact of incentives between events in different dimensions. Specifically, we have *Z* Hawkes processes that are coupled with each other: each of the Hawkes processes corresponds to an individual and the influence between individuals are explicitly modeled. Formally, the conditional intensity for the d-th dimension expressed as follows:

$$\lambda_d(t) = \mu_d + \sum_{i:t_i<t} a_{d_i d} \exp\left(-w(t - t_i)\right)$$

The coefficient $a_{d_i d} \geq 0$ captures the mutually-exciting property from the d_i-th to d-th dimension. We collect the parameters into matrix-vector forms $\boldsymbol{\mu} = (\mu_d)$ for the base intensity, and $\boldsymbol{A} = (a_{d_i d})$ for the mutuallyexciting coefficients, called infectivity matrix.

Task

In this homework, $Z = 10, T = 100(optional),\ w = 0.01, \boldsymbol{\mu}$ and $\boldsymbol{A}$ are expressed as follows:

***μ*:**

0	0.001	0.05	0.1	0.025	0.01	0.007	0.03	0.008	0.004

***A*:**

0.1	0.07	0.004	0	0.003	0	0.09	0	0.07	0.025
0	0.05	0.028	0	0.027	0.065	0	0	0.097	0
0.09	0	0.006	0.045	0	0	0.053	0.01	0	0.083
0.02	0.03	0	0.073	0.058	0	0.026	0	0	0
0.05	0.09	0	0	0.066	0	0	0.033	0.006	0
0.07	0	0	0	0	0.075	0.063	0.078	0.085	0.095
0	0.02	0.001	0	0.057	0.091	0.009	0.065	0	0.073
0	0.09	0	0.088	0	0.078	0	0.09	0.068	0
0	0	0.093	0	0.033	0	0.069	0	0.082	0.033
0.001	0	0.089	0	0.008	0	0.007	0	0	0.052

You should simulate 1000 sequences with a maximum length of 200.

In report, you should describe the software language, key libraries, and other things necessary to run the program. Then you should explain what kind of problems are encountered and how to solve them. As for the results of sumulation, you should statistically analyze the sequences simulated. Moreover, you can change the value of $Z, T, w, \boldsymbol{\mu}, \boldsymbol{A}$ and explore the influence to hawkes in order to understand the meaning of these parameters.

Submission

You should submit a .zip file with all source code and the report and not include the sequence data simulated.

Grade

Source code(50%): code norm, program running

Report(50%): well-organized and analytical

Note: implement with C++ can be regraded as a bonus.

Reference

1. Ogata Y. On Lewis' simulation method for point processes[J]. IEEE transactions on information theory, 1981, 27(1): 23-31.
2. Chen Y. Thinning Algorithms for Simulating Point Processes[J]. 2016.

Academic misconduct and handling

We will review all source code and analyze the plagiarism of source code and reports, therefore, don't try to test its reliability. For self-completed student, there will be reward points. For plagiarized student, it will be awarded zero points. If you are convicted of plagiarism, you can ask TAs to make a defense.