VIETNAM GENERAL CONFEDERATION OF LABOUR
TON DUC THANG UNIVERSITY
FACULTY OF MATHEMATICS AND STATISTICS

HAWKES PROCESSES

by

Nguyen Le Thao Trang

Le Thi Minh Phuong

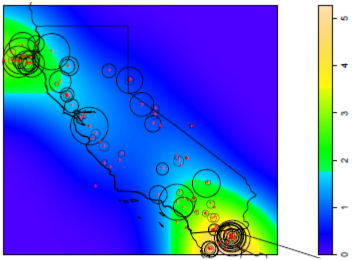
advised by

Dr. Nguyen Chi Thien

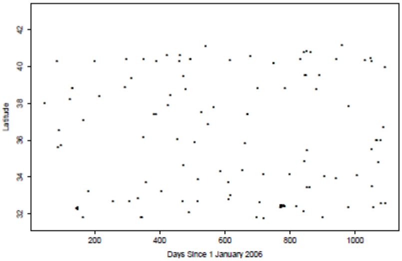
Ho Chi Minh City, Jun 2019

Outline

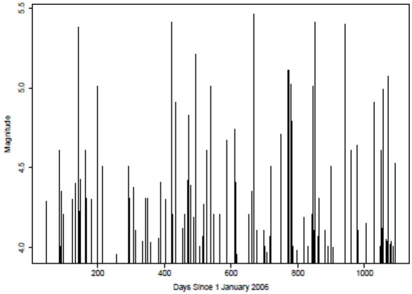
Introduction



(a)



(b)



Background

Definition 1

(Point process) Let $\{T_i, i \in \mathbb{N}\}$ be a sequence of non-negative random variables such that $\forall i \in \mathbb{N}$, $T_i < T_{i+1}$. Then $\{T_i, i \in \mathbb{N}\}$ is a (simple) point process.

Background

Definition 2

(Counting process) A counting process is a stochastic process $(N(t) : t \geq 0)$ taking values in $\mathbb{N}_0$ that satisfies $N(0) = 0$, is almost surely finite, and is a right-continuous step function with increments of size $+1$. Denote by $(\mathcal{H}(u) : u \geq 0)$ the history of the arrivals up to time u.

Background

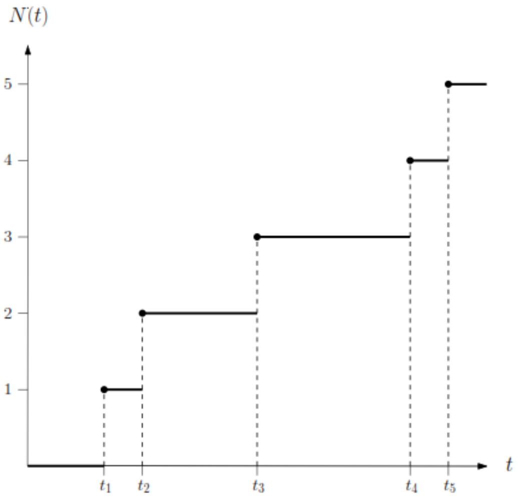


Figure 1: Point process $\{t_1, t_2, ...\}$ and corresponding counting process $N(t)$.

Background

Definition 3

(Inhomogeneous Poisson Process) Consider $(N(t) : t \geq 0)$ a counting process and that satisfies

$$\mathbb{P}(N(t+h) - N(t) = m|N(t)) = \begin{cases} \lambda(t)h, & \text{if } m = 1 \\ o(h), & \text{if } m > 1 \\ 1 - \lambda(t)h + o(h), & \text{if } m = 0 \end{cases}$$

Then $N(t)$ is called a inhomogeneous Poisson process with $\lambda : \mathbb{R}^+ \to \mathbb{R}^+$.

Background

Definition 4

(Hawkes process) Consider $(N(t) : t \geq 0)$ a counting process, with associated history $\mathcal{H}(t) : t \geq 0$, that satisfies

$$\mathbb{P}(N(t+h) - N(t) = m|\mathcal{H}(t)) = \begin{cases} \lambda^*(t)h + o(h), & \text{if } m = 1 \\ o(h), & \text{if } m > 1 \\ 1 - \lambda^*(t)h + o(h), & \text{if } m = 0 \end{cases}$$

Suppose the process' conditional intensity function is of the form

$$\lambda^*(t) = \lambda + \int_0^t \mu(t-u)dN(u)$$

for some $\lambda > 0$ and $\mu : (0, \infty) \rightarrow [0, \infty)$ which are called the background intensity and excitation function respectively. Suppose that $\mu(.) \neq 0$, then a process $N(.)$ is a Hawkes process.

Simulation Algorithms - Inhomogeneous Poisson

Algorithm 1 Generate an inhomogeneous Poisson process by thinning.

INPUT: T is the time to simulate;
$\lambda(.)$ is the intensity function;
M is bounded value;

OUTPUT: The vector P containing the times of occurrences $\{t_1, t_2, ..., t_n\}$;

Require: $\lambda(.) \leq M$ on $[0, T]$.

Step 1: Set $P \leftarrow []$, $t \leftarrow 0$

Step 2: while $t < T$ do

a. Generate next candidate point $E \leftarrow \text{Exp}\ (M)$, $t \leftarrow t + E$
b. Keep it with some probability $U \leftarrow \text{Unif}(0, M)$
c. if $t < T$ and $U \leq \lambda(t)$ then $P \leftarrow [P, t]$

Simulation Algorithms - Inhomogeneous Poisson

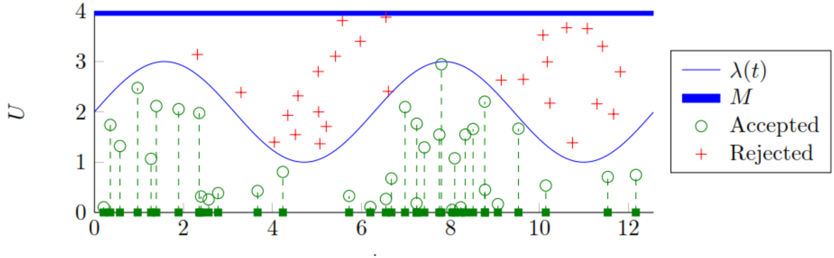


Figure 2: Inhomogeneous Poisson.

Simulation Algorithms - Intensity-based Hawkes Process

Algorithm 2 Generate a Hawkes process by thinning.

INPUT: T is the time to simulate;

$\lambda^*(.)$ is the conditional intensity function;

OUTPUT: The vector P containing the times of occurrences $\{t_1, t_2, ..., t_n\}$;

Require: $\lambda^*(.)$ non-increasing in periods without any arrivals.

Step 1: Set $\varepsilon \leftarrow 10^{-10}$, $P \leftarrow []$, $t \leftarrow 0$

Step 2: while $t < T$ do

a. Find new upper bound: $M \leftarrow \lambda^*(t + \varepsilon)$

b. Generate next candidate point $E \leftarrow \text{Exp}(M)$, $t \leftarrow t + E$

c. Keep it with some probability $U \leftarrow \text{Unif}(0, M)$

d. if $t < T$ and $U \leq \lambda^*(t)$ then $P \leftarrow [P, t]$

Simulation Algorithms - Intensity-based Hawkes Process

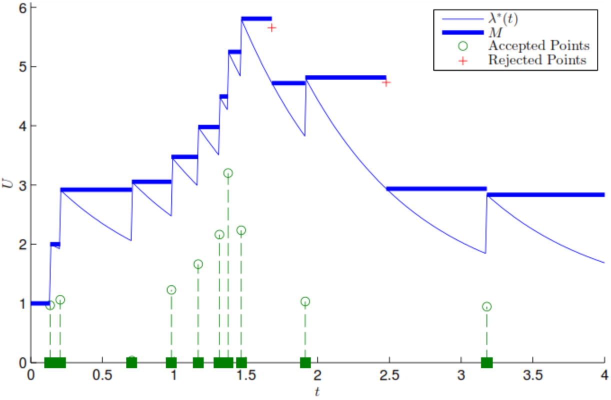


Figure 3: Intensity-based Hawkes Process.

Simulation Algorithms - Cluster-based Hawkes Process

Algorithm 3 Generate a Hawkes process by clusters.

INPUT: T is the time to simulate;
(λ, α, β) are parameters of the conditional intensity function;
OUTPUT: P is the union of all the clusters $\{C_1, C_2, ..., C_k\}$;
Step 1: Set $P \leftarrow \{\}$
Step 2: Generate the immigrants:
a. $k \leftarrow \text{Poi}(\lambda T)$
b. $C_1, C_2, ..., C_k \xleftarrow{\text{independent and identically distributed}} \text{Unif}(0, T)$
Step 3: Generate the descendants:
a. $D_1, D_2, ..., D_k \xleftarrow{\text{independent and identically distributed}} \text{Poi}(\alpha/\beta)$
Step 4: for $i \leftarrow 1$ to k do
a. if $D_i > 0$ then
a.1 $E_1, E_2, ..., E_{D_i} \xleftarrow{\text{independent and identically distributed}} \text{Exp}(\beta)$
a.2 $P \leftarrow P \cup \{C_i + E_1, C_i + E_2, ..., C_i + E_{D_i}\}$

Simulation Algorithms - Cluster-based Hawkes Process

Step 5: Remove descendant outside $[0, T]$

$$P \leftarrow \{P_i : P_i \in P, P_i \leq T\}$$

Step 6: Add in immigrants and sort: $P \leftarrow \text{Sort}\ (P \cup \{C_1, C_2, ..., C_k\})$

Simulation Algorithms - Cluster-based Hawkes Process

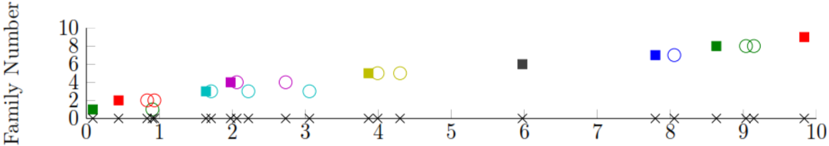


Figure 4: Cluster-based Hawkes Process.

THANK YOU!