Self-Organized Hawkes Processes

Shen Yuan   Hongteng Xu

May 24, 2021

- Introduction
- Proposed Model
- Learning Algorithm
- Experiments

Problems

The real-world applications of Hawkes processes often suffer from the following limitations:

- Complicated mutual-triggering pattern of the different event types within each real-world sequence.
- Huge number of event types and extremely few historical observations.

Motivation

The complicated global relation tends to consist of simple local relation.

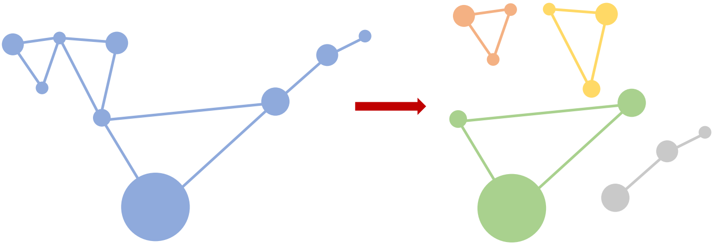

Figure 1: An illustration between the complex global interrelation and the simple local structures.

Contributions

This paper makes the following contributions:

- The self-organized Hawkes process(SOHP) model that learns heterogeneous local Hawkes processes based on subsets of observed event sequences.
- A method that apply SOHP model into sequential recommendation system, which achieves higher f1 scores than state-of-the-art methods.

- Introduction
- Proposed Model
- Learning Algorithm
- Experiments

Notation

- $t \in [0, T]$ the timestamp
- $\mathcal{C}$ the set of event types
- $\{(t_i, c_i)\}_{i=1}^{I}$ an event sequence
- $N_c(t)$ the number of type-c events occurring till time t
- $\mathcal{H}_t^{\mathcal{C}}$ the historical observations till time t.
- $\lambda(\cdot)$ the intensity function
- $\mathcal{L}(\cdot)$ the likelihood function
- $\mathcal{R}(\cdot)$ the regularization term

The Self-organized Hawkes Process Model

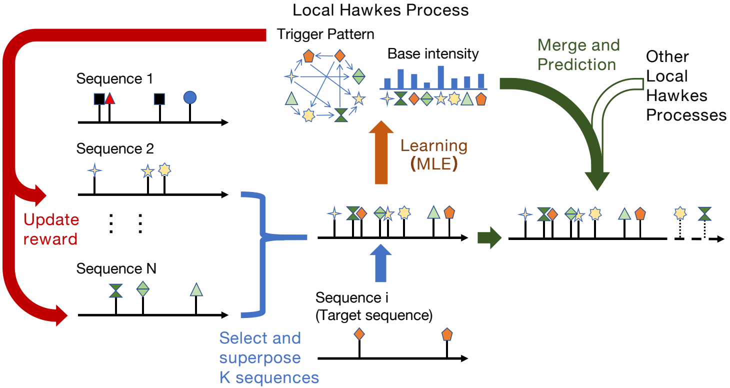

Figure 2: An illustration of the proposed self-organized Hawkes process model.

Intensity Function

$$\lambda_c(t) = \mu_c + \sum\nolimits_{t_i<t} \phi_{cc_i}(t, t_i) = \mu_c + \sum\nolimits_{t_i<t} a_{cc_i}\kappa(t - t_i),\ \forall\ c \in \mathcal{C}, \quad (1)$$

where,

- μ_c represents the basic happening rate of the type-c event;
- $\phi_{cc'}(t, t')$, $t' < t$ and $c, c' \in \mathcal{C}$, is called *impact function*, which represents the influence of the type-c' event at time t' on the type-c event at time t.

Objective Function

By the maximum likelihood estimation, we can learn the Hawkes process:

$$\min_{\boldsymbol{\theta}} -\log \mathcal{L}(\mathcal{N};\boldsymbol{\theta}) + \gamma\mathcal{R}(\boldsymbol{\theta}), \tag{2}$$

where,

- $\mathcal{N} = \{N^u\}_{u\in\mathcal{U}}$ is a set of event sequences;
- $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{A}\}$ represents the model parameter;
- $\mathcal{L}(\mathcal{N};\boldsymbol{\theta})$ is the likelihood function of the event sequences.

Prediction

Instead of simulation, we use (3) to predict the future event:

$$p(c|t+\Delta t, \mathcal{H}_t^{\mathcal{C}}) = \frac{\lambda_c(t+\Delta t)}{\sum_{c'\in\mathcal{C}} \lambda_{c'}(t+\Delta t)}. \tag{3}$$

Superposition Property

We are motivated by a fact that the complex global interrelation between events tend to consist of simple local structures. Meanwhile, there exists the superposition property of Hawkes process:

Theorem (Superposition Property)

For a set of independent Hawkes processes with a shared infectivity matrix, i.e., $\{N^u \sim HP(\boldsymbol{\mu}^u, \boldsymbol{A})\}_{u\in\mathcal{U}}$, *the superposition of their sequences satisfies* $\sum_{u\in\mathcal{U}} N^u \sim HP(\sum_{u\in\mathcal{U}} \boldsymbol{\mu}^u, \boldsymbol{A})$.

Thus, we can superpose different Hawkes processes to learn the target Hawkes process.

Superposition Property

To satisfy the assumption imposed in the superposition property, we design a new self-organization mechanism, selecting event sequences for each local Hawkes process and adjusting the selection with the training progress. Thus, the objective function becomes

$$\min_{\{\boldsymbol{\theta}^u\}_{u\in\mathcal{U}}} \min_{\mathcal{N}^u\subset\mathcal{N}} -\sum_{u\in\mathcal{U}} \frac{1}{|N^u \cup \mathcal{N}^u|} \log \mathcal{L}(N^u \cup \mathcal{N}^u; \boldsymbol{\theta}^u), \qquad (4)$$

where $\mathcal{N}^u = \{N^{s_1}; \ldots; N^{s_K}\}$ represents the K neighbors of the u-th sequence.

A Reward-augmented Bandit Algorithm

For each target sequence, we treat the selection of its neighbors (*i.e.*, the training set of a Hawkes process) as a multi-armed bandit problem, selecting their neighbors according to the potential *rewards*.

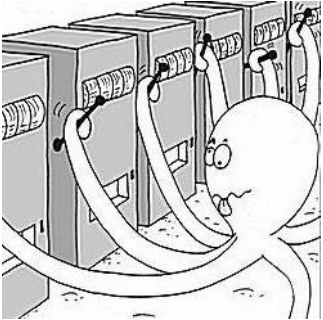

Figure 3: The multi-armed bandit problem.

A Reward-augmented Bandit Algorithm

We formulate a benefit matrix $\boldsymbol{B} = [b_{u,k}] \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{U}|}$, where $b_{u,k}$ represents the benefit from selecting the k-th sequence for the u-th Hawkes process.

Initial Benefit

We define the initial benefit based on the optimal transport distance between event sequences.

$$
\begin{aligned}
d(N^u, N^v) :=\ & \min_{\boldsymbol{T}\in\Pi(\frac{1}{|\mathcal{C}_u|}\mathbf{1}_{|\mathcal{C}_u|}, \frac{1}{|\mathcal{C}_v|}\mathbf{1}_{|\mathcal{C}_v|})} \sum\nolimits_{i\in\mathcal{C}_u, j\in\mathcal{C}_v} T_{ij} d(N_i^u, N_j^v) \\
=\ & \min_{\boldsymbol{T}\in\Pi(\frac{1}{|\mathcal{C}_u|}\mathbf{1}_{|\mathcal{C}_u|}, \frac{1}{|\mathcal{C}_v|}\mathbf{1}_{|\mathcal{C}_v|})} \langle \boldsymbol{D}_{uv}, \boldsymbol{T} \rangle
\end{aligned}
\tag{5}
$$

where,

- $\Pi(\frac{1}{|\mathcal{C}_u|}\mathbf{1}_{|\mathcal{C}_u|}, \frac{1}{|\mathcal{C}_v|}\mathbf{1}_{|\mathcal{C}_v|}) = \{\boldsymbol{T} \geq \mathbf{0} | \boldsymbol{T}\mathbf{1} = \frac{1}{|\mathcal{C}_u|}\mathbf{1}_{|\mathcal{C}_u|}, \boldsymbol{T}^\top\mathbf{1} = \frac{1}{|\mathcal{C}_v|}\mathbf{1}_{|\mathcal{C}_v|}\}$ represents the set of joint distributions with marginals $\frac{1}{|\mathcal{C}_u|}\mathbf{1}_{|\mathcal{C}_u|}$ and $\frac{1}{|\mathcal{C}_v|}\mathbf{1}_{|\mathcal{C}_v|}$;
- $\boldsymbol{D}_{uv} = [d(N_i^u, N_j^v)] \in \mathbb{R}^{|\mathcal{C}_u|\times|\mathcal{C}_v|}$ is a distance matrix, where $d(N_i^u, N_j^v) = \frac{1}{T}\int_0^T |N_i^u(t) - N_j^v(t)|\mathrm{d}t$ represents the discrepancy between the sequence of the type-i events and that of the type-j events.

Optimal Transport Distance

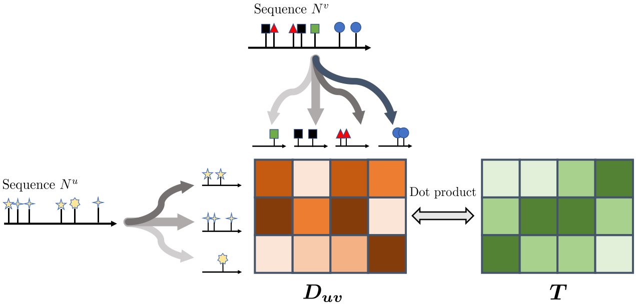

Figure 4: The optimal transport distance between heterogeneous event sequences.

Algorithm Description

Algorithm 1 Learning SOHP via a reward-augmented bandit algorithm

Input: Event sequences $\{N^u\}_{u\in\mathcal{U}}$, distance matrix $\boldsymbol{D} = [d(N^u, N^v)] \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{U}|}$, maximum iterations L, the number of neighbors K, learning rate α.
Output: benefit matrix $\boldsymbol{B} = [b_{u,k}] \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{U}|}$ and model parameters $\{\boldsymbol{\theta}^u\}_{u\in\mathcal{U}}$.

```
for u = 1 : |U| do
    Initialize b_{u,k} = max_v d(N^u, N^v) − d(N^u, N^k) for k ∈ U.
    for l = 1 : L do
        if l < L then
            Set p = [b_{u,1} / Σ_i b_{u,i}, .., b_{u,|U|} / Σ_i b_{u,i}], sample {N^{s_1}, .., N^{s_K}} from N with p.
        else
            Select {N^{s_1}, .., N^{s_K}} with the K highest benefits.
        end if
        Learn the model parameter θ^u from {N^u} ∪ {N^{s_1}, .., N^{s_K}} by (2).
        for k = s_1 : s_K do
            b_{u,k} = b_{u,k} + α L(N^{s_k}; θ^u)
        end for
    end for
end for
```

Where: line 2 is $b_{u,k} = \max_v d(N^u, N^v) - d(N^u, N^k)$ for $k \in \mathcal{U}$; line 5 is $\boldsymbol{p} = [\frac{b_{u,1}}{\sum_i b_{u,i}}, .., \frac{b_{u,|\mathcal{U}|}}{\sum_i b_{u,i}}]$, sample $\{N^{s_1}, .., N^{s_K}\}$ from $\mathcal{N}$ with $\boldsymbol{p}$; line 11 is $b_{u,k} = b_{u,k} + \alpha\mathcal{L}(N^{s_k}; \boldsymbol{\theta}^u)$.

Merging learned Hawkes processes

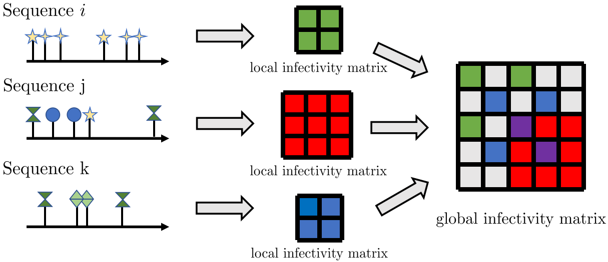

Figure 5: An illustration of merging local infectivity matrices. The purple ones correspond to the overlapped event types, whose values are accumulated together.

Datasets

We conduct our experiments on the Amazon dataset.

Figure 6: Amazon review data.

Datasets

Table 1: Statistics of our datasets

Categories	Musical Instruments	Baby	Video Games	Garden	Instant Video
#Users	471	1979	2142	1812	5948
#Items	678	2134	2104	2064	1344
#Ratings	1218	6070	6126	4976	15470

Experiments

Table 2: Summary of the performance for baselines and our model

Datasets	Musical Instruments			Baby			Video Games			Garden			Instant Video		
Measures@10(%)	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
SVD	0.106	1.061	0.193	0.121	0.735	0.207	0.065	0.498	0.113	0.055	0.395	0.094	0.126	1.052	0.221
kNN	0.382	2.671	0.649	0.389	2.513	0.638	0.661	4.915	1.163	0.237	1.751	0.405	0.817	6.901	1.435
BPR	0.467	3.750	0.811	0.389	2.469	0.635	0.658	4.864	1.112	0.110	0.762	0.185	0.859	7.049	1.503
SLIM	0.212	1.351	0.347	0.111	0.712	0.180	0.499	3.595	0.835	0.242	1.544	0.401	**1.333**	**11.428**	**2.351**
FPMC	0.594	4.193	1.006	0.283	1.912	0.470	0.556	3.799	0.927	0.171	1.117	0.285	0.931	7.413	1.622
SHP	0.361	2.406	0.604	0.258	1.734	0.432	0.317	2.037	0.525	0.199	1.350	0.331	0.933	7.406	1.623
Our methods	**0.658**	**5.149**	**1.138**	**0.389**	**2.640**	**0.651**	**0.700**	**5.108**	**1.180**	**0.248**	**1.801**	**0.436**	0.999	7.911	1.738

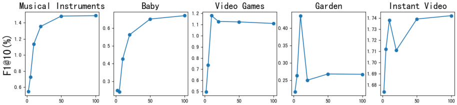

Figure 7: Performance of our model under different number of neighbors K.