Self-Organized Hawkes Processes

Shen Yuan[1], **Hongteng Xu**[2]

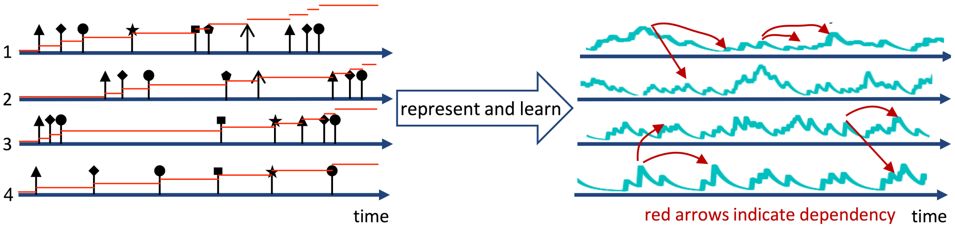
[1]University of Electronic Science and Technology of China,
[2]Gaoling School of Artificial Intelligence, Renmin University of China

中國人民大學
RENMIN UNIVERSITY OF CHINA

Background: Temporal Point Processes

- **Event sequence:** $S = \{(t_i, c_i)\}_{i=1}^{I}$, $c_i \in \mathcal{C}$. (Social behavior, shopping, etc.)
- **Counting processes:** $N(t) = \{N_c(t)\}_{k=1}^{K}$.
- **Intensity function:** The expected instantaneous happening rate of type-c events given the history.

$$\lambda_d(t) = \frac{\mathbb{E}[dN_c(t)|\mathcal{H}_t]}{dt}, \ \mathcal{H}_t = \{(t_i, c_i)|t_i < t, c_i \in \mathcal{C}\}.$$



Hawkes Processes

Hawkes Process $\mathrm{HP}_{\mathcal{C}}(\boldsymbol{\mu}, \boldsymbol{A})$ models the triggering pattern between different events:

$$\lambda_c(t) = \underbrace{\mu_c}_{\text{base intensity}} + \sum_{(t_i,c_i)\in\mathcal{H}_t} \underbrace{\phi_{cc_i}(t-t_i)}_{\text{impact function}} = \mu_c + \sum_{(t_i,c_i)\in\mathcal{H}_t} \underbrace{\alpha_{cc_i}}_{\text{infectivity}} \kappa(t-t_i). \tag{1}$$

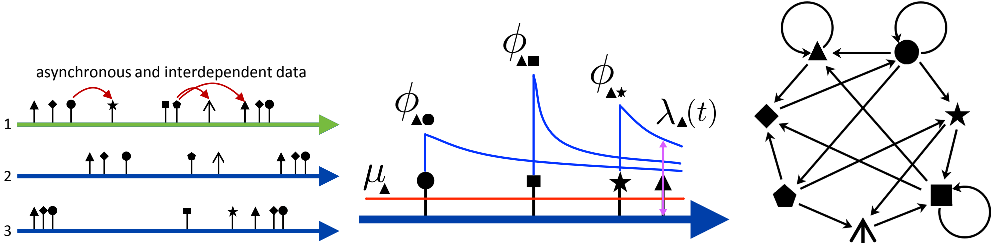
- $\boldsymbol{\mu} = [\mu_c]$: exogenous fluctuation of the system.
- $\boldsymbol{\Phi} = [\phi_{cc'}(t) = a_{cc'}\kappa(t)]$: endogenous triggering pattern of type-c' on type-c.

Hawkes Processes

Hawkes Process $\text{HP}_{\mathcal{C}}(\boldsymbol{\mu}, \boldsymbol{A})$ models the triggering pattern between different events:

$$\lambda_c(t) = \underbrace{\mu_c}_{\text{base intensity}} + \sum_{(t_i,c_i)\in\mathcal{H}_t} \underbrace{\phi_{cc_i}(t-t_i)}_{\text{impact function}} = \mu_c + \sum_{(t_i,c_i)\in\mathcal{H}_t} \underbrace{\alpha_{cc_i}}_{\text{infectivity}} \kappa(t-t_i). \quad (1)$$

- $\boldsymbol{\mu} = [\mu_c]$: exogenous fluctuation of the system.
- $\boldsymbol{\Phi} = [\phi_{cc'}(t) = a_{cc'}\kappa(t)]$: endogenous triggering pattern of type-c' on type-c.



(d) Observations (e) Intensity per entity (f) Granger causality

Learning and Challenges

$$\begin{aligned}
&\min_{\boldsymbol{\theta}} - \underbrace{\log \mathcal{L}(\mathcal{N}; \boldsymbol{\theta})}_{\text{log-likelihood}} + \underbrace{\gamma \mathcal{R}(\boldsymbol{\theta})}_{\text{regularizer}}, \text{ where } \boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{A}\}, \\
&\mathcal{L}(\mathcal{N}; \lambda) = \prod_{u \in \mathcal{U}} \Big(\prod_{(c_i^u, t_i^u) \in N^u} \lambda_{c_i^u}(t_i^u) \times \exp\big(-\sum_{c \in \mathcal{C}} \int_0^T \lambda_c^u(s) ds\big)\Big).
\end{aligned} \tag{2}$$

Learning and Challenges

$$\min_{\boldsymbol{\theta}} -\underbrace{\log \mathcal{L}(\mathcal{N};\boldsymbol{\theta})}_{\text{log-likelihood}} + \underbrace{\gamma\mathcal{R}(\boldsymbol{\theta})}_{\text{regularizer}}, \text{ where } \boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{A}\},$$
$$\mathcal{L}(\mathcal{N};\lambda) = \prod_{u\in\mathcal{U}}\Big(\prod_{(c_i^u, t_i^u)\in N^u} \lambda_{c_i^u}(t_i^u) \times \exp\big(-\sum_{c\in\mathcal{C}} \int_0^T \lambda_c^u(s)ds\big)\Big). \tag{2}$$

- The real-world event sequences $\mathcal{N} = \{N^u\}_{u\in\mathcal{U}}$ are often driven by multiple *local heterogeneous Hawkes processes* $\{\text{HP}_{\mathcal{C}_u}(\boldsymbol{\mu}_u, \boldsymbol{A}_u)\}_{u\in\mathcal{U}}$.
  - **Local:** $\mathcal{C}_u \subset \mathcal{C}$ and $|\mathcal{C}_u| \ll |\mathcal{C}|$.
  - **Heterogeneous:** $\mathcal{C}_u \neq \mathcal{C}_{u'}$ for $u \neq u'$.

Learning and Challenges

$$\min_{\boldsymbol{\theta}} - \underbrace{\log \mathcal{L}(\mathcal{N};\boldsymbol{\theta})}_{\text{log-likelihood}} + \underbrace{\gamma\mathcal{R}(\boldsymbol{\theta})}_{\text{regularizer}}, \text{ where } \boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{A}\},$$
$$\mathcal{L}(\mathcal{N};\lambda) = \prod_{u\in\mathcal{U}}\Big(\prod_{(c_i^u,t_i^u)\in N^u} \lambda_{c_i^u}(t_i^u) \times \exp\big(-\sum_{c\in\mathcal{C}} \int_0^T \lambda_c^u(s)ds\big)\Big). \tag{2}$$

- The real-world event sequences $\mathcal{N} = \{N^u\}_{u\in\mathcal{U}}$ are often driven by multiple *local heterogeneous Hawkes processes* $\{\mathrm{HP}_{\mathcal{C}_u}(\boldsymbol{\mu}_u, \boldsymbol{A}_u)\}_{u\in\mathcal{U}}$.
  - **Local:** $\mathcal{C}_u \subset \mathcal{C}$ and $|\mathcal{C}_u| \ll |\mathcal{C}|$.
  - **Heterogeneous:** $\mathcal{C}_u \neq \mathcal{C}_{u'}$ for $u \neq u'$.
- However, the huge number of event types and extremely few observations make the learning of the HPs over-fitting even intractable.

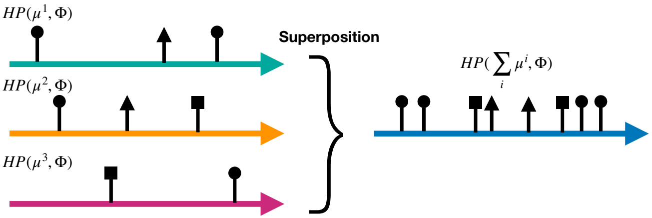
Proposed Learning Task

Learning multiple local heterogeneous Hawkes processes from few short sequences.

Proposed Learning Task

Learning multiple local heterogeneous Hawkes processes from few short sequences.

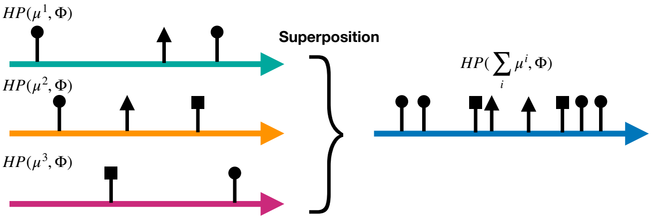
- **Key 1:** Leverage the **Superposition property** to increase the event density of each individual sequence. [Xu et al. AISTATS'18]



Proposed Learning Task

Learning multiple local heterogeneous Hawkes processes from few short sequences.

- **Key 1:** Leverage the **Superposition property** to increase the event density of each individual sequence. [Xu et al. AISTATS'18]



The problem becomes **learning each individual HP from selective subset of sequences.**

$$\min_{\{\boldsymbol{\theta}^u\}_{u\in\mathcal{U}}} \min_{\mathcal{N}^u \subset \mathcal{N}} -\sum_{u\in\mathcal{U}} \frac{1}{|N^u \cup \mathcal{N}^u|} \log \mathcal{L}(N^u \cup \mathcal{N}^u; \boldsymbol{\theta}^u), \tag{3}$$

Select Neighbors via A Bandit Algorithm

- **Key 2:** Design a Reward-augmented Bandit Algorithm to select neighbors for each sequence in the training phase.
- Formulate a benefit matrix $\boldsymbol{B} = [b_{u,v}] \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{U}|}$, where $b_{u,v}$ represents the benefit from selecting the v-th sequence for the u-th Hawkes process.

Select Neighbors via A Bandit Algorithm

- **Key 2:** Design a Reward-augmented Bandit Algorithm to select neighbors for each sequence in the training phase.
- Formulate a benefit matrix $\boldsymbol{B} = [b_{u,v}] \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{U}|}$, where $b_{u,v}$ represents the benefit from selecting the v-th sequence for the u-th Hawkes process.

1. Initialize $b_{u,v} = \max_k d(N^u, N^k) - d(N^u, N^v)$ for $v \in \mathcal{U}$, where $d(N^u, N^k)$ is the optimal transport distance between different event sequences.

Select Neighbors via A Bandit Algorithm

- **Key 2:** Design a Reward-augmented Bandit Algorithm to select neighbors for each sequence in the training phase.
- Formulate a benefit matrix $\boldsymbol{B} = [b_{u,v}] \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{U}|}$, where $b_{u,v}$ represents the benefit from selecting the v-th sequence for the u-th Hawkes process.

1. Initialize $b_{u,v} = \max_k d(N^u, N^k) - d(N^u, N^v)$ for $v \in \mathcal{U}$, where $d(N^u, N^k)$ is the optimal transport distance between different event sequences.

2. Select neighbor sequences $\{s_v\}$ for each sequence u by a bandit algorithm (*e.g.*, the Upper Confidence Bound method).

Select Neighbors via A Bandit Algorithm

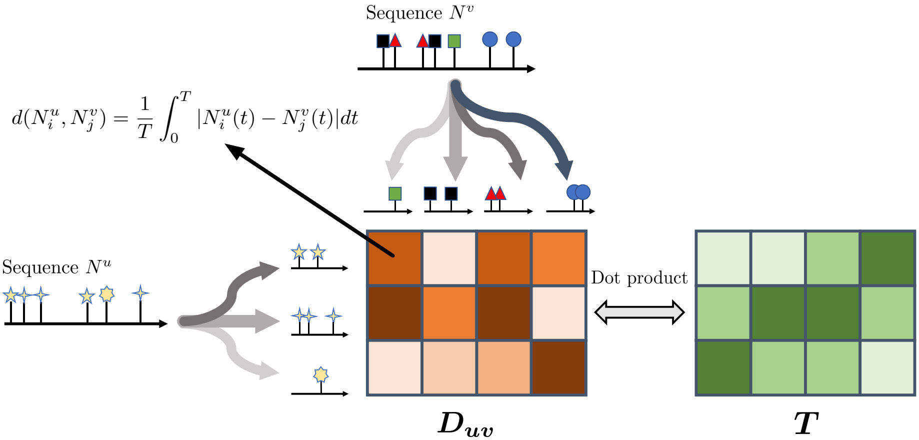
- **Key 2:** Design a Reward-augmented Bandit Algorithm to select neighbors for each sequence in the training phase.
- Formulate a benefit matrix $\boldsymbol{B} = [b_{u,v}] \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{U}|}$, where $b_{u,v}$ represents the benefit from selecting the v-th sequence for the u-th Hawkes process.

1. Initialize $b_{u,v} = \max_k d(N^u, N^k) - d(N^u, N^v)$ for $v \in \mathcal{U}$, where $d(N^u, N^k)$ is the optimal transport distance between different event sequences.
2. Select neighbor sequences $\{s_v\}$ for each sequence u by a bandit algorithm (*e.g.*, the Upper Confidence Bound method).
3. Update $b_{u,v} = b_{u,v} + \alpha\mathcal{L}(N^{s_v}; \boldsymbol{\theta}^u)$.

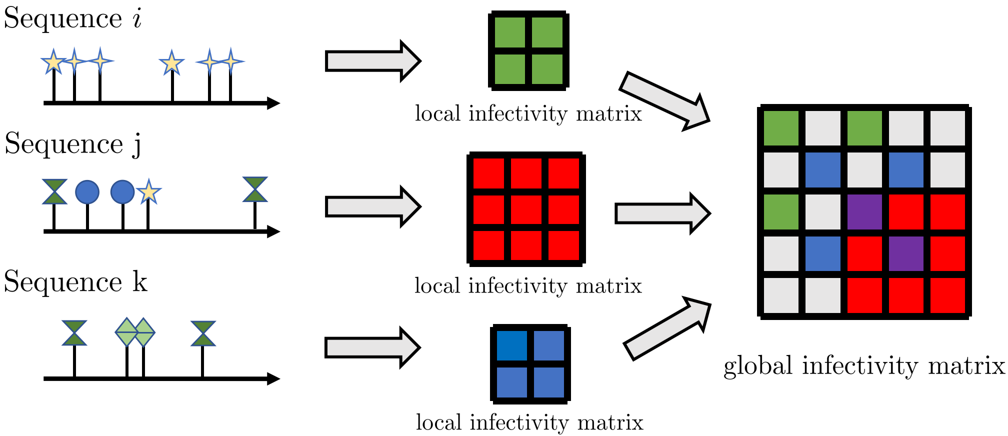
Optimal Transport Distance between Sequences

$$b_{u,v} = \max_k d(N^u, N^k) - d(N^u, N^v) \text{ for } v \in \mathcal{U}\text{, where}$$

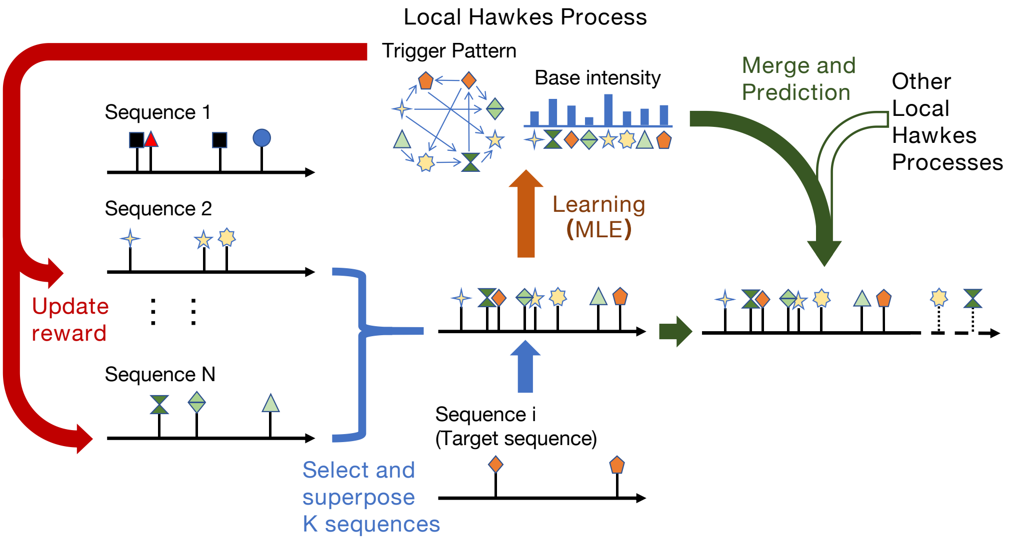
$$d(N^u, N^v) := \min_{\boldsymbol{T} \in \Pi(\frac{1}{|\mathcal{C}_u|}\mathbf{1}_{|\mathcal{C}_u|}, \frac{1}{|\mathcal{C}_v|}\mathbf{1}_{|\mathcal{C}_v|})} \sum_{i \in \mathcal{C}_u, j \in \mathcal{C}_v} T_{ij} d(N_i^u, N_j^v) = \min_{\boldsymbol{T} \in \Pi(\frac{1}{|\mathcal{C}_u|}\mathbf{1}_{|\mathcal{C}_u|},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



Merging learned Hawkes processes for exploration



Self-organized Hawkes Processes



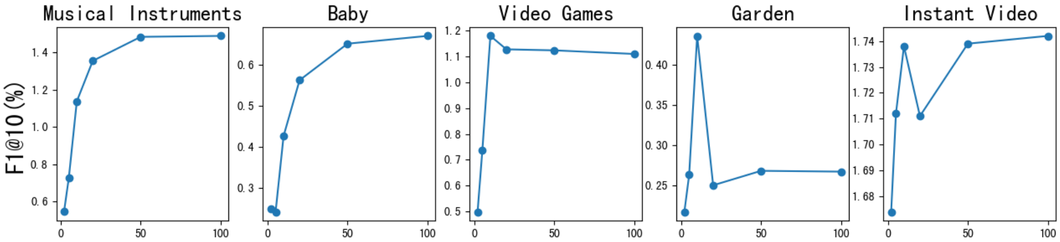
Experiments: Continuous-Time Sequential Recommendation

For the items with ≥ 40 purchasing behaviors, learning user behaviors in 3 months and predict their behaviors in the next 3 months.

$$p(c|t+\Delta t, \mathcal{H}_t^{\mathcal{C}}) = \frac{\lambda_c(t+\Delta t)}{\sum_{c'\in\mathcal{C}} \lambda_{c'}(t+\Delta t)}. \tag{4}$$

Data	Musical Instruments			Baby			Video Games			Garden			Instant Video		
@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
Ours	**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

Experiments: Influence of the number of neighbors K



Summary

- The self-organized Hawkes process(SOHP) model that learns heterogeneous local Hawkes processes based on selective subsets of event sequences.
- A learning algorithm combining bandit algorithm and optimal transport is proposed.
- A method that apply SOHP model into sequential recommendation system, which achieves higher f1 scores than state-of-the-art methods.
- The code is available at `https://github.com/UESTC-DaShenZi/MHP`.