Continuous-Time Recommendation via Self-Organized Hawkes Processes*

Anonymous YOFO submission

Paper ID 100

Abstract. xu: Use 6-8 sentences to introduce the aim of this work, the principle of your method and highlight its novelties and superiority.
The superposition of Hawkes processes has been widely used in many application scenarios because of its effectiveness. The existing models, however, almost adopt the random algorithm to select sequences. We propose a novel framework based on Bandit algorithm for learning the superposition of Hawkes processes. Instead of randomly selecting the sequence, the matching degree between sequences is calculated iteratively which the superposed event sequences are selected according to. We design the reward of the Bandit algorithm, consider the optimal transport distance as the initial reward and leverage the greedy strategy to update the reward. Our model achieves better results than the state-of-the-art method in the recommendation experiment.

Keywords: Hawkes process · Bandit algorithm · Superposition.

1 Introduction

xu: In the first paragraph, use 2-4 sentences to highlight the significance of Hawkes processes.

xu: The drawbacks of Hawkes processes: fail to sparse data; scalability issue. What is worse, numerous event types + extremely few events (take RecSys as an example). How to solve this challenging case is the key contribution of this work.

xu: Then, introduce the proposed self-organized Hawkes process briefly WITH a figure illustrating your scheme. Bandit algorithm. Superpose.

xu: Summary your novelties and contributions.

In the real world, there exists many event sequences occurring in continuous time, such as high frequency finance [1], social network[7], and earthquake[13], etc. It is essential to model the interaction of these event and predict future happening rates of events. Hawkes process is a variant of point process able to model these event sequences, which includes exogenous fluctuation and endogenous triggering term.

However, Hawkes process exists many deficiencies. Firstly, Hawkes process performs poorly in sparse event sequences. On the other hand, it has a scalability issue which means massive event types. Therefore, recommendation system

* Supported by organization x.

hardly applies Hawkes process because it usually learns from a large number of item types and sparse purchase records. To solve this, we propose the self-organized Hawkes processes.

As shown in Figure 1, our model select and superpose K sequences based on expected gain or matching degree, and update it by calculating maximum likelihood estimation iteratively. We design the reward in the bandit algorithm and apply the greedy strategy to update the expected gain. Besides, we analyze some attributes of the model, and demonstrate the usefulness in the recommendation systems.

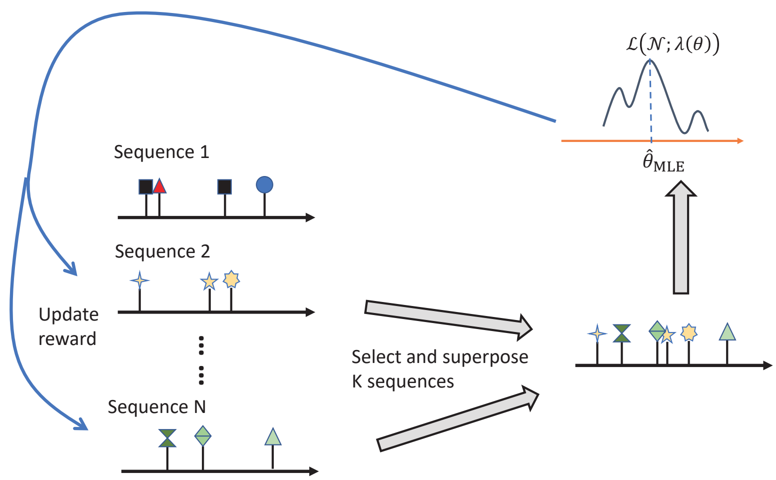

Fig. 1. Self-organized Hawkes processes

2 Proposed Model

2.1 Continuous-time recommendation based on Hawkes processes

xu: Introduce Hawkes process and the pipeline of the recommendation system based on it.

Multi-dimensional Hawkes Process For a user u, we denote $N_t^u = \{(c_i^u, t_i^u) | t_i^u < t\}$ as her purchasing behaviors till time t, where $c_i^u \in \mathcal{C}$ is the item she bought at time t_i^u. Assume that her behaviors yield a multivariate Hawkes process of the items, her expected instantaneous purchasing rate at time t (*i.e.*, her conditional intensity function) is

$$\lambda_c^u(t) = \mu_c + \sum_{t_i^u < t} a_{cc_i^u} \kappa(t - t_i^u),\ \forall\ c \in \mathcal{C}, \tag{1}$$

where $\boldsymbol{A} = [a_{cj}] \in \mathbb{R}^{|\mathcal{C}|\times|\mathcal{C}|}$ is the infectivity matrix capturing the triggering patterns among the items, $\boldsymbol{\mu} = [\mu_c] \in \mathbb{R}^{|\mathcal{C}|}$ represents the basic happening rates of the items, $\kappa(\cdot)$ is the kernel function.

Learning Given a purchasing sequence $\mathcal{H}$, we can learn the Hawkes process by Maximum Likelihood Estimation (MLE)

$$\min_{\lambda} \ -\log \mathcal{L}(N;\lambda) + \alpha\mathcal{R}(\lambda) \tag{2}$$

where $\mathcal{L}(\cdot)$ is the maximum likelihood function. In temporal point process, it can be computed as

$$\mathcal{L}(N;\lambda) = \prod_{(c_i,t_i)\in N} \lambda_{c_i}(t_i) \times \exp\left(-\int \lambda(s)\mathrm{d}s\right) \tag{3}$$

And $\mathcal{R}(\cdot)$ represents the regularization term. This method applies the idea that the observed events are most probable and updates parameters to maximize the likelihood of the observed events.

In addition, we could fit the observed counting processes via the integral of intensity functions to learn model *i.e.* Least-Square Estimation(LS).

$$\min_{\{\lambda_c\}_{c=1}^{\mathcal{C}}} \sum_{i=1}^{\mathcal{I}} \sum_{c=1}^{\mathcal{C}} \frac{1}{t_i^2} \left[|\hat{N}_c(t_i)| - \int_0^{t_i} \lambda_c(s)\mathrm{d}s\right]^2 \tag{4}$$

where $|N_c(t_i)|$ deontes the number of item c bought till time t_i.

Prediction After the conditional intensity function is obtained, we could predict the user's next behavior in the future. Given a time interval Δt, the probability that each item will be purchase could be computed by:

$$p(c|t+\Delta t, \mathcal{H}_t) = \frac{\lambda_c(t+\Delta t)}{\lambda(t+\Delta t)} \tag{5}$$

There is, however, an obvious problem. For most users, they tend to have an extremely short purchasing history, which makes it difficult to learn parameters of point process from a single user's history. In order to figure out this problem, we proposed the self-organized Hawkes processes, which will be detailedly described in next section.

2.2 Self-organized Hawkes processes

xu: Introduce SOHP

Suppose that event sequences are defined as $\mathcal{N} = \{N^u\}_{u\in|\mathcal{U}|}$. For each sequence N^u, we select K sequences $\{N^{s_1};\ldots;N^{s_K}\}$ based on a bandit algorithm, and learn a Hawkes process accordingly from the superposition of them.

[18] verified the improvement by the random selection on selecting sequences. However, the distinction between sequences varies with the length of sequences. When the sequence length is short and the difference between the sequences is small, the random algorithm can make up for the information missing. When the sequence length is long, the difference between the sequences is large, and random superposition will make the information confusion caused by the distinction cover up the information compensation, where the random algorithm is not suitable to use. Thus, we adopted the bandit algorithm to select the sequences, which not only improved the interpretability of the model, but also improved the results in the subsequent experiments.

We apply the offline bandit strategy in this model, which means neighbors of each user decided before learning Hawkes process will be unchanged in the training phase. Thus, it's necessary to design a measurement scheme of similarity of purchasing sequences.

3 Learning Algorithm

xu: Give the details of your algorithms and analyze its complexity and other properties.

3.1 A reward-augmented bandit algorithm

We propose a novel method to apply bandit algorithm to learning Hawkes process. Suppose that event sequences are defined as $\mathcal{N} = \{N^u\}_{u \in |\mathcal{U}|}$. The aim is to find the set of the most matching sequences for each one. To achieve this target, we design the objective function as:

$$\max \sum_{u \in |\mathcal{U}|} \max_{\mathcal{N}' \subset \mathcal{N}} \mathcal{L}(N^u \cup \mathcal{N}'; \lambda) \tag{6}$$

where $\mathcal{L}(\cdot)$ is the maximum likelihood function, $\mathcal{N}'$ denotes the set of the most similar sequences for sequence u, and λ is the intensity function learning from $N^u \cup \mathcal{N}'$. We regard $\mathcal{L}(\cdot)$ as the reward of choices $\mathcal{N}'$ in bandit algorithm, which measures matching degree of the sequences.

The steps of bandit algorithm are shown in Algorithm 1. The computational complexity of the algorithm is $\mathcal{O}(|\mathcal{U}|^2 L)$ due to $|\mathcal{U}| \gg K$. This algorithm reduces the computational complexity of learning Hawkes process. In each iteration, we only learn the superposition of K sequences, and the number of event types is much less than the total number of that, which greatly reduced the dimension of Hawkes process.

In Algorithm 2, with the benefit matrix B obtained in Algorithm 1, we select K sequences $\{N^{s_1}; \dots; N^{s_K}\}$ of the most similar ones and superpose them for each sequence N, respectively. After learning that, instead of leveraging the intensity function with parameter $\boldsymbol{\theta}_{\mathcal{N}'}$ to predict, we update the $\boldsymbol{\theta}$ with $\boldsymbol{\theta}_{\mathcal{N}'}$. Until all the sequences have been traversed, we use $\boldsymbol{\theta}$ to predict. The deficiency of the former is that only items appearing in the K sequences will be recommend, where the algorithm degenerates to the k-nearest neighbors algorithm [10].

Algorithm 1 Similarity based on Bandit algorithm

Input: Event sequences $\mathcal{N} = \{N^u\}_{u \in \mathcal{U}}$, distance matrix of sequences $V = [d(N^u, N^v)] \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{U}|}$, iterations L, number of neighbors K, learning rate α
Output: benefit matrix $B \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{U}|}$

```
for u = 1 : |U| do
    B_{u,:} = max(V_{u,:}) − V_{u,:}
    for l = 1 : L do
        Set p = [B_{u,1} / Σ_i B_{u,i}; ... ; B_{u,|U|} / Σ_i B_{u,i}]
        Sample K sequences {N^{s_1}; ... ; N^{s_K}} from N with p
        Initialize N′ = {N^u} ∪ {N^{s_1}; ... ; N^{s_K}}
        Learn MHP(θ) from N′
        for k = s_1 : s_K do
            B_{u,k} = B_{u,k} + αL(N′; θ)
        end for
    end for
end for
```

3.2 Distance between event sequences

The bandit algorithm faces the challenge of the combinatorial explosion of neighbors when calculating the initial benefit. Its computational complexity is $\mathcal{O}(|\mathcal{U}|^{K+1}/K!)$ due to that each user needs to be selected at least once, and each iteration needs to select K neighbors. To solve this, we calculated the optimal transport distance between the purchase sequences for every two users as the initial benefit for each user in the bandit algorithm.

For $N^u = \{N_i^u\}_{i \in \mathcal{V}_u}$ and $N^v = \{N_j^v\}_{j \in \mathcal{V}_v}$, where $\mathcal{V}_u$ and $\mathcal{V}_v$ are respectively the sets of items purchased by user u and v. Here, we leverage optimal transport distance to calculate :

$$\begin{aligned} d(N^u, N^v) &:= \min_{\boldsymbol{T} \in \prod(\frac{1}{|\mathcal{V}_u|}\mathbf{1}_{|\mathcal{V}_u|}, \frac{1}{|\mathcal{V}_v|}\mathbf{1}_{|\mathcal{V}_v|})} \sum_{i \in \mathcal{V}_u, j \in \mathcal{V}_v} T_{ij} d(N_i^u, N_j^v) \\ &= \min_{\boldsymbol{T} \in \prod(\frac{1}{|\mathcal{V}_u|}\mathbf{1}_{|\mathcal{V}_u|}, \frac{1}{|\mathcal{V}_v|}\mathbf{1}_{|\mathcal{V}_v|})} \langle \boldsymbol{D}_{uv}, \boldsymbol{T} \rangle \end{aligned} \tag{7}$$

where $\boldsymbol{T}$ is the optimal transport matix which measures the correlation between different event types, $\boldsymbol{D}_{uv} = [d(N_i^u, N_j^v)] \in \mathbb{R}^{|\mathcal{V}_u| \times |\mathcal{V}_v|}$ calculates the distance for sequence N^u and N^v, and $d(N_i^u, N_j^v) = \frac{1}{T}\int_0^T |N_i^u(t) - N_j^v(t)| \mathrm{d}t$ represents the discrepancy between the sequence of the item i and that of the item j.

This reduces the computational complexity to $\mathcal{O}(M^2 |\mathcal{U}|^2)$, where M denotes the average length of sequences.

4 Related Work

xu: Enumerate related works and analyze their pros and cons.

Algorithm 2 Learning self-organized Hawkes processes

Input: Event sequences $\mathcal{N} = \{N^u\}_{u\in\mathcal{U}}$, benefit matrix $B \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{U}|}$, number of neighbors K

Output: Parametes $\boldsymbol{\theta}$

1: Initialize $\boldsymbol{\theta}$ with zero
2: **for** $u = 1 : |\mathcal{U}|$ **do**
3: Set $s_1, \dots, s_K$ as the indices of the top K maximums of $B_{u,:}$
4: Set $\mathcal{N}' = \{N^u\} \cup \{N^{s_1}; \dots; N^{s_K}\}$
5: Learn MHP($\boldsymbol{\theta}_{\mathcal{N}'}$) from $\mathcal{N}'$
6: Add $\boldsymbol{\theta}_{\mathcal{N}'}$ to $\boldsymbol{\theta}$
7: **end for**

4.1 Sequential recommendation

By modeling the sequential behaviors from user historical records, sequential recommendation predicts future interests and recommend items. Besides the shopping recommendation, sequential recommendation has also been widely used in various application scenarios, such as web recommendation [21], music recommendation [4], and Point-of-Interest recommendation [5][9], etc.

These years, many cutting-edge techniques have been applied into the sequential recommend, *e.g.*, [14](FPMC) integrated matrix factorization and Markov chains, [16](HRM) regarded representation learning as latent factors, they modeled the sequential behavior patterns via every two contiguous historical records. [19](DREAM) based on recurrent neural network (RNN) learnt the global sequential behavior patterns.

These models all captured the contiguous behavior information without taking time interval between them into considering. Instead, we leverage the time stamps of every behavior to calculate efficiently the mutual effect of them.

4.2 Hawkes process

Because of its effective modeling of the interaction in real-world event sequences, Hawkes process has been widely used in many scenarios, such as high frequency finance [1], and fake news mitigation [8]. Meanwhile, many variants based on Hawkes process have been researched and developed, such as Hawkes process with self-attention [20] [22], Hawkes process with Granger causality graph [17]. Recently, the superposition of Hawkes process has been verified to be effective in both theory and experiments [18]. The model, however, applied a random algorithm to select sequences, where we demonstrate that a bandit algorithm is more valid.

4.3 Bandit problem

Bandit problem, *i.e.*, the multi-armed bandit problem denotes a problem where we need to make choices to maximize expected gain under some constraints[15].

These years, more bandit algorithms have been proposed, such as Upper Confidence Bounds algorithm [12] [6] [3], adaptive epsilon-greedy strategy based on Bayesian ensembles [11], and behavior constrained Thompson Sampling [2], etc. In this paper, we make an attempt to apply the bandit algorithm into superposition of Hawkes process.

5 Experiments

5.1 Implementation details

xu: Introduce the data, the baselines, the evaluation criteria, and the configurations of hyperparameters in your work.

5.2 Comparisons and analysis

xu: Show your experimental results here. Besides direct comparisons, try to add more analytic experiments to demonstrate the rationality of the method — show the necessity of using self-organized Hawkes process model and its learning algorithm.

xu: Table 1. Comparison with existing methods on recommendation accuracy.

xu: Runtime comparison between your method and other Hawkes-based variants.

xu: The influence of the number of neighbors for each dataset.

xu: The influence of different reward designs.

6 Conclusion

xu: Summarize your work and say something about your future work.

In this paper, we proposed a framework leveraging the superposition of Hawkes processes integrated with bandit algorithm for sequential recommendation. To do so, we design the reward of the Bandit algorithm, and exert the greedy strategy to update the reward iteratively. What's more, we analyse the computation complexity and properties of this model. In the future, we plan to design new formulation of rewards and update rewards with other strategies. Meanwhile, it's exciting to apply our framework to more application scenarios.

References

1. Bacry, E., Mastromatteo, I., Muzy, J.F.: Hawkes processes in finance. Market Microstructure and Liquidity **1**(01), 1550005 (2015)
2. Balakrishnan, A., Bouneffouf, D., Mattei, N., Rossi, F.: Incorporating behavioral constraints in online ai systems. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 3–11 (2019)

3. Bouneffouf, D., Parthasarathy, S., Samulowitz, H., Wistub, M.: Optimal exploitation of clustering and history information in multi-armed bandit. arXiv preprint arXiv:1906.03979 (2019)
4. Chen, S., Moore, J.L., Turnbull, D., Joachims, T.: Playlist prediction via metric embedding. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 714–722 (2012)
5. Cheng, C., Yang, H., Lyu, M.R., King, I.: Where you like to go next: Successive point-of-interest recommendation. In: Twenty-Third international joint conference on Artificial Intelligence (2013)
6. Chu, W., Li, L., Reyzin, L., Schapire, R.: Contextual bandits with linear payoff functions. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. pp. 208–214. JMLR Workshop and Conference Proceedings (2011)
7. Farajtabar, M., Rodriguez, M.G., Zamani, M., Du, N., Zha, H., Song, L.: Back to the past: Source identification in diffusion networks from partially observed cascades. In: Artificial Intelligence and Statistics. pp. 232–240. PMLR (2015)
8. Farajtabar, M., Yang, J., Ye, X., Xu, H., Trivedi, R., Khalil, E., Li, S., Song, L., Zha, H.: Fake news mitigation via point process based intervention. In: International Conference on Machine Learning. pp. 1097–1106. PMLR (2017)
9. Feng, S., Li, X., Zeng, Y., Cong, G., Chee, Y.M.: Personalized ranking metric embedding for next new poi recommendation. In: IJCAI'15 Proceedings of the 24th International Conference on Artificial Intelligence. pp. 2069–2075. ACM (2015)
10. Fix, E.: Discriminatory analysis: nonparametric discrimination, consistency properties, vol. 1. USAF school of Aviation Medicine (1985)
11. Gimelfarb, M., Sanner, S., Lee, C.G.: {\epsilon}-bmc: A bayesian ensemble approach to epsilon-greedy exploration in model-free reinforcement learning. arXiv preprint arXiv:2007.00869 (2020)
12. Li, L., Chu, W., Langford, J., Schapire, R.E.: A contextual-bandit approach to personalized news article recommendation. In: Proceedings of the 19th international conference on World wide web. pp. 661–670 (2010)
13. Ogata, Y.: Statistical models for earthquake occurrences and residual analysis for point processes. Journal of the American Statistical association **83**(401), 9–27 (1988)
14. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized markov chains for next-basket recommendation. In: Proceedings of the 19th international conference on World wide web. pp. 811–820 (2010)
15. Robbins, H.: Some aspects of the sequential design of experiments. Bulletin of the American Mathematical Society **58**(5), 527–535 (1952)
16. Wang, P., Guo, J., Lan, Y., Xu, J., Wan, S., Cheng, X.: Learning hierarchical representation model for nextbasket recommendation. In: Proceedings of the 38th International ACM SIGIR conference on Research and Development in Information Retrieval. pp. 403–412 (2015)
17. Xu, H., Farajtabar, M., Zha, H.: Learning granger causality for hawkes processes. In: International Conference on Machine Learning. pp. 1717–1726. PMLR (2016)
18. Xu, H., Luo, D., Chen, X., Carin, L.: Benefits from superposed hawkes processes. In: International Conference on Artificial Intelligence and Statistics. pp. 623–631. PMLR (2018)
19. Yu, F., Liu, Q., Wu, S., Wang, L., Tan, T.: A dynamic recurrent model for next basket recommendation. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. pp. 729–732 (2016)

20. Zhang, Q., Lipani, A., Kirnap, O., Yilmaz, E.: Self-attentive hawkes processes. arXiv preprint arXiv:1907.07561 (2019)
21. Zhang, Y., Zhang, M., Liu, Y., Tat-Seng, C., Zhang, Y., Ma, S.: Task-based recommendation on a web-scale. In: 2015 IEEE International Conference on Big Data (Big Data). pp. 827–836. IEEE (2015)
22. Zuo, S., Jiang, H., Li, Z., Zhao, T., Zha, H.: Transformer hawkes process. In: International Conference on Machine Learning. pp. 11692–11702. PMLR (2020)