Hierarchical Contrastive Learning for Temporal Point Processes

Qingmei Wang[1], **Minjie Cheng**[1], **Shen Yuan**[1], **Hongteng Xu**[1, 2]*

[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]Beijing Key Laboratory of Big Data Management and Analysis Methods
hongtengxu@ruc.edu.cn

Abstract

As an essential sequential model, the temporal point process (TPP) plays a central role in real-world sequence modelling and analysis, whose learning is often based on the maximum likelihood estimation (MLE). However, due to imperfect observations, such as incomplete and sparse sequences that are common in practice, the MLE of TPP models often suffers from overfitting and leads to unsatisfactory generalization power. In this work, we develop a novel hierarchical contrastive (HCL) learning method for temporal point processes, which provides a new regularizer of MLE. In principle, our HCL considers the noise contrastive estimation (NCE) problem at the event-level and that at the sequence-level jointly. Given a sequence, the event-level NCE maximizes the probability of each observed event given its history while penalizing the conditional probabilities of the unobserved events. At the same time, we generate positive and negative event sequences from the observed sequence and maximize the discrepancy between their likelihoods through the sequence-level NCE. Instead of using time-consuming simulation methods, we generate the positive and negative sequences via a simple but efficient model-guided thinning process. Experimental results show that the MLE method assisted by the HCL regularizer outperforms classic MLE and other contrastive learning methods in learning various TPP models consistently. The code is available at https://github.com/qingmeiwangdaily/HCL_TPP.

Introduction

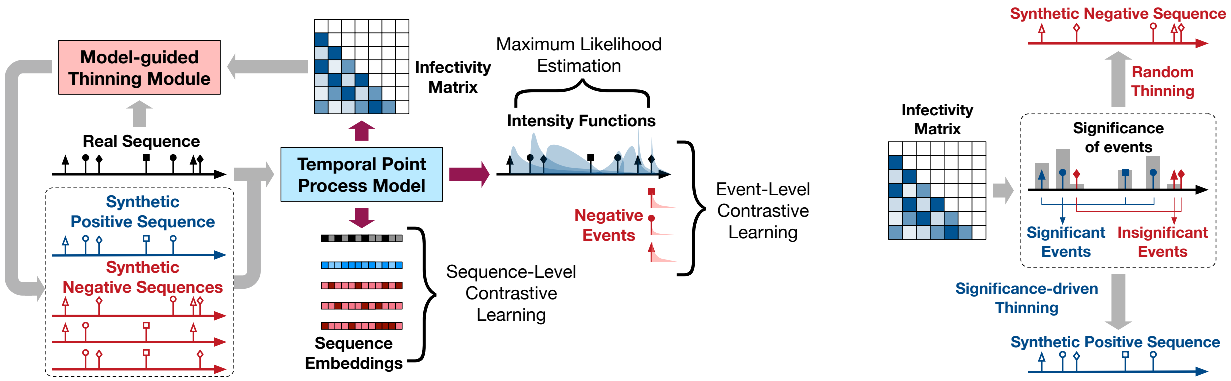
Continuous-time event sequences are ubiquitous in real-world scenarios, such as earthquakes (Lewis and Shedler 1979), financial transactions (Bacry, Mastromatteo, and Muzy 2015), social behaviors (Zhou, Zha, and Song 2013a), e-commercial behaviors (Xu, Carin, and Zha 2018), etc. Facing such event sequences, temporal point processes (TPPs) have been widely used as typical sequential models and achieved encouraging performance in various applications, including healthcare data modeling (Xu et al. 2016), social network analysis (Zhao et al. 2015), high-frequency trading prediction (Rambaldi, Bacry, and Lillo 2017), and so on. However, the learning of TPPs often suffers from the imperfectness issue of data: Real-world event sequences may be too short, censored, and/or with missing events and noisy timestamps, and learning TPPs from such data may lead to poor generalization power and unsatisfactory performance.

Many efforts have been made to learn TPPs robustly from imperfect observations. Typically, when learning TPPs via the maximum likelihood estimation (MLE) (Ozaki 1979), sparse and/or low-rank regularization is often applied to avoid the over-fitting of the models (Zhou, Zha, and Song 2013b; Xu, Farajtabar, and Zha 2016). Besides adding regularizers, various data augmentation methods, like random stitching (Xu, Luo, and Zha 2017), superposition (Xu et al. 2018), counterfactual inference (Noorbakhsh and Rodriguez 2021), and model-driven data imputation (Mei, Qin, and Eisner 2019), have been introduced to improve the quality of observed event sequences. Additionally, leveraging generative adversarial networks (GANs) (Goodfellow et al. 2014) and reinforcement learning (Sutton and Barto 2018), we can introduce discriminators (Xiao et al. 2017a, 2018; Yan et al. 2018) or reward functions (Li et al. 2018; Upadhyay, De, and Gomez-Rodrizuez 2018; Zhu et al. 2021) to guide the learning of TPP models.

More recently, to simplify the learning task itself, discriminative learning (Xu et al. 2016) and contrastive learning (Guo, Li, and Liu 2018; Mei, Wan, and Eisner 2020) are applied. In particular, instead of maximizing the likelihood of event sequences, these methods focus on a relatively easier task — distinguishing the real observations from potential negative samples. Although these methods have achieved some encouraging results, they often suffer from the following challenges. Firstly, the side information associated with the data, e.g., a pretrained TPP model (Mei, Wan, and Eisner 2020), is required to design the corresponding contrastive loss and simulate negative events, which is often unreliable and even unavailable. Secondly, the generation of negative events depends on time-consuming thinning algorithms (Lewis and Shedler 1979; Ogata 1981), which limits the scalability and efficiency of the methods. Thirdly, existing methods only consider negative events and the corresponding event-level contrastive learning while ignoring the necessity of "negative sequences", which may still lead to the over-fitting issue because of the natural uncertainty of individual events in TPPs.

To overcome the above challenges, we propose a novel learning method, called hierarchical contrastive learning

*Corresponding author.


Model-guided Thinning Module
Infectivity Matrix
Maximum Likelihood Estimation
Real Sequence
Temporal Point Process Model
Intensity Functions
Synthetic Positive Sequence
Synthetic Negative Sequences
Negative Events
Event-Level Contrastive Learning
Sequence-Level Contrastive Learning
Sequence Embeddings
Synthetic Negative Sequence
Random Thinning
Infectivity Matrix
Significance of events
Significant Events
Insignificant Events
Significance-driven Thinning
Synthetic Positive Sequence

(a) Hierarchical contrastive learning

(b) Model-guided thinning

Figure 1: The scheme of our learning method. (a) An illustration of hierarchical contrastive learning. (b) An illustration of model-guided thinning.

(HCL), for temporal point process models, which constructs event-level and sequence-level contrastive loss functions to regularize the MLE of TPPs in a hierarchical way. As shown in Figure 1(a), given a timestamp and the historical event sequence before the time, the target TPP model outputs i) the intensity vector indicating the instantaneous happening rates of the events with different types at time t; ii) the latent embedding of the historical event sequence; and iii) a lower-triangular infectivity matrix indicating the impacts of each event on its successors. Our HCL leverages the intensity vector to construct the event-level contrastive loss, maximizing the intensity of the real observed (i.e., positive) event type while suppressing those of the other unobserved (i.e., negative) event types. Furthermore, the infectivity matrix provides us with the evidence to recognize the events having significant impacts. Therefore, we develop a model-guided thinning module to generate positive sequences and negative sequences by randomly removing those insignificant and significant events from the original sequence, respectively, as shown in Figure 1(b). Given the embeddings of the positive and negative sequences, we construct the sequence-level contrastive loss to maximize the difference between them.

The sequence-level contrastive loss, which is seldom considered by existing work, helps to capture the uncertainty of historical impacts on the same event. Without using pre-trained model (Mei, Wan, and Eisner 2020) or sophisticated thinning processes (Lewis and Shedler 1979; Mei, Qin, and Eisner 2019), the proposed sequence generation method is purely dependent on the infectivity matrix derived by the intermediate TPP model during training, which is very efficient. Our HCL method is applicable to most existing TPP models, including Hawkes process, self-correcting process, and their neural network-based variants. Such high compatibility is beneficial for its application in practice. Experiments demonstrate the superiority of our HCL method.

Related Work

Temporal Point Processes Temporal point process is a classic statistical tool to model the event sequences in continuous-time domain. The simplest TPP model is Poisson process (Kingman 1992), which models event sequences in a memory-less way. For the event sequences whose events have a temporal dependency, the TPPs like Hawkes process (HP) (Hawkes 1971) and self-correcting process (SCP) (Isham and Westcott 1979) leverage additive or multiplicative mechanisms to build triggering patterns among different events quantitatively, whose triggering patterns can be extended to multivariate scenarios (Liniger 2009; Xu et al. 2016). Recently, modelling TPPs by deep neural networks helps to capture the event sequences with complicated dynamics, which has achieved encouraging performance in various applications. The recurrent marked temporal point process (RMTPP) in (Du et al. 2016) extends recurrent neural networks (RNNs) to continuous-time scenarios, which is one of the representative neural TPP models. Similarly, the neural Hawkes process in (Mei and Eisner 2017) works as a continuous-time LSTM (CT-LSTM) model. More recently, some attempts have been made to model the temporal dependency of events based on the well-known self-attention mechanism (Vaswani et al. 2017), e.g., the self-attentive Hawkes process (SAHP) (Zhang et al. 2020) and the transformer Hawkes process (THP) (Zuo et al. 2020).

Learning Methods of TPPs Typically, we learn TPP models based on maximum likelihood estimation (MLE), which aims at maximizing the likelihood of observed event sequences (Kingman 1992). As aforementioned, real-world data are often noisy and incomplete, and the MLE method may lead to misspecified TPPs. To improve the robustness of MLE, we leverage the prior knowledge and impose regularizers on model parameters, e.g., the sparse and low-rank regularizers of triggering patterns (Zhou, Zha, and Song 2013b) and the topological regularizer of Granger causality graphs (Zhang, Lipani, and Yilmaz 2021).

Besides imposing structural regularizers, another robust learning strategy is applying MLE to augmented data. Various data augmentation methods have been proposed, e.g., the random stitching (Xu, Luo, and Zha 2017) and superposition (Xu et al. 2018) for Hawkes processes and the counterfactual sequence generation for more general TPPs (Noorbakhsh and Rodriguez 2021). Additionally, the combination of the data augmentation methods and the MLE framework leads to the reinforcement learning (RL) methods of TPPs (Li et al. 2018; Upadhyay, De, and Gomez-Rodrizuez 2018). These RL methods often simulate event sequences based on the current model, calculate the rewards of the simulated event sequences, and update the model by the policy gradient algorithm. However, the simulation of event sequence requires the Ogata's thinning algorithm (Ogata 1981), which is time-consuming when the number of event types or the time length is large.

Instead of applying MLE, the work in (Xu et al. 2016) applies discriminative learning to multivariate TPPs, which maximizes the conditional probabilities of observed event types. This objective is also used as a regularizer in some recent work (Zuo et al. 2020; Zhang et al. 2020). The work in (Xiao et al. 2017a, 2018) develops generative adversarial networks (GANs) to learn TPPs, which leads to a min-max optimization problem. More recently, the idea of contrastive learning has been introduced into the learning task of TPPs. Essentially, the contrastive learning methods, e.g., the noise contrastive estimation (NCE) (Gutmann and Hyvärinen 2012), aim at distinguishing the negative samples and the positive ones conditioned on the observed data. Following the same idea, an NCE-based method called "Initiator" is developed in (Guo, Li, and Liu 2018). Similarly, the work in (Mei, Wan, and Eisner 2020) proposes to train TPPs by discriminating the observed events from the events sampled from a (pretrained/predefined) noisy point process, which is another version of NCE. This method is demonstrated to achieve consistent estimations of the TPPs with (nearly) continuous intensity functions. Still, it depends on a given noise point processes and needs to simulate negative events by the time-consuming Ogata's thinning method (Ogata 1981). Note that these NCE methods merely focus on the contrastive loss at the event-level. In the following content, we will show that it is possible to design a sequence-level contrastive loss with both effectiveness and efficiency and learn TPPs robustly.

Proposed Method

A Generalized Formulation of TPPs

Suppose that we observe an event sequence $\boldsymbol{s} = \{(t_n, c_n)\}_{i=1}^{N}$, where (t_n, c_n) is the n-th event, $t_n \in [0, T]$ is its timestamp, and $c_n \in \mathcal{C} = \{1, ..., C\}$ is its event type. We would like to develop a TPP model, which is characterized by a multivariate intensity function $\boldsymbol{\lambda}(t) = \{\lambda_c(t)\}_{c\in\mathcal{C},t\in[0,T]}$, where

$$\lambda_c(t) = \frac{\mathrm{d}\mathbb{E}[N_c(t)|\mathcal{H}_t^{\mathcal{C}}]}{\mathrm{d}t},\ \forall c \in \mathcal{C}, \tag{1}$$

which represents the instantaneous rate of the type-c event happening at time t given historical events (Liniger 2009). Here, $N_c(t)$ is the counting process of the event type and $\mathcal{H}_t^{\mathcal{C}} = \{(t_n, c_n) \in \boldsymbol{s} | t_n < t\}$ is the history till time t.

Typically, we implement the TPP as a parametric model, i.e., f_θ, where θ represents the model parameters. In this study, we consider a generalized formulation of commonly-used TPP models, which is illustrated in Figure 1(a) and is defined as follows:

$$\boldsymbol{\lambda}(t),\ \boldsymbol{G},\ \boldsymbol{e} = f_\theta(t, \boldsymbol{s}_t). \tag{2}$$

In this formulation, the TPP model takes a timestamp t and the historical events before t (i.e., $\boldsymbol{s}_t := \mathcal{H}_t^{\mathcal{C}}$ with length $L = |\boldsymbol{s}_t|$) as its input, and output three terms:

- The multivariate intensity vector at time t, i.e., $\boldsymbol{\lambda}(t) = [\lambda_c(t)] \in \mathbb{R}^C$.
- The lower-triangular infectivity matrix for events, denoted as $\boldsymbol{G} = [g_{ij}] \in \mathbb{R}^{L\times L}$, where g_{ij} with $i > j$ represents the influence of the j-th event on the i-th event, and $g_{ij} = 0$ if $i \leq j$.
- The embedding vector of the historical event sequence till time t, i.e., $\boldsymbol{e} \in \mathbb{R}^D$.

Note that the formulation in (2) is generalized, which covers many representative TPP models. Here, we give some examples below:

Hawkes Process (HP) (Zhou, Zha, and Song 2013b) Given t and $\boldsymbol{s}_t$, the HP derives the intensity of each event type at time t as

$$\lambda_c(t) = \mu_c + \sum\nolimits_{(t_n,c_n)\in\boldsymbol{s}_t} a_{cc_n}\kappa(t - t_n),\ \forall c \in \mathcal{C}. \tag{3}$$

Then, inspired by the EM algorithm of multivariate Hawkes process (Zhou, Zha, and Song 2013b), we can define the events' infectivity matrix $\boldsymbol{G} = [g_{ij}]$ as

$$g_{ij} = \frac{a_{c_ic_j}\kappa(t_i - t_j)}{\lambda_{c_i}(t_i)}. \tag{4}$$

The sequence embedding $\boldsymbol{e}$ can be set as the average of the intensity vector over time, i.e., $\boldsymbol{e} = \frac{1}{L}\sum_{i=1}^{L}\boldsymbol{\lambda}(t_i) \in \mathbb{R}^C$.

Self-Correcting Process (SCP) (Isham and Westcott 1979; Xu et al. 2016) Given t and $\boldsymbol{s}_t$, the self-correcting process derives the intensity of each event type at time t as

$$\lambda_c(t) = \exp\Big(\mu_c t - \sum\nolimits_{(t_n,c_n)\in\boldsymbol{s}_t} a_{cc_n}\Big),\ \forall c \in \mathcal{C}. \tag{5}$$

We can define the events' infectivity matrix $\boldsymbol{G} = [g_{ij}]$ as

$$g_{ij} = \frac{\exp(a_{c_ic_j})}{\lambda_{c_i}(t_i)}, \tag{6}$$

and the sequence embedding can be defined as the average of the intensity vector as well.

Transformer Hawkes Process (THP) (Zuo et al. 2020) Given t and $\boldsymbol{s}_t$, the THP leverages a transformer (Vaswani et al. 2017) to obtain the embedding of the event sequence $\boldsymbol{e}$. The events' infectivity matrix $\boldsymbol{G}$ is the intermediate result generated by the self-attention module in the transformer. Finally, the intensity vector at time t is obtained by

$$\boldsymbol{\lambda}(t) = \mathrm{Softplus}(\boldsymbol{\mu} + \boldsymbol{W}\boldsymbol{e}_{\boldsymbol{s}_t}) \tag{7}$$

where $\boldsymbol{\mu} = [\mu_c] \in \mathbb{R}^C$ is the base intensity vector, $\boldsymbol{W} = [\boldsymbol{w}_1^T; ...; \boldsymbol{w}_C^T] \in \mathbb{R}^{C\times D}$, and $\boldsymbol{w}_c \in \mathbb{R}^D$ for $c \in \mathcal{C}$.

Other models like the neural Hawkes process in (Mei and Eisner 2017) and the self-attentive Hawkes process in (Zhang et al. 2020) can also be captured in the same formulation. As aforementioned, given a sequence $\boldsymbol{s}$, we often learn the above TPP model via maximum likelihood estimation (MLE) (Ogata et al. 1978). The learning task is maximizing the log-likelihood of $\boldsymbol{s}$:

$$\max_{f_\theta}\mathcal{L}(\boldsymbol{s};\theta)=\max_{f_\theta}\sum_{n=1}^{N}\log\lambda_{c_n}(t_n)-\sum_{c\in\mathcal{C}}\int_0^T\lambda_c(s)\mathrm{d}s. \quad (8)$$

However, purely relying on MLE may lead to unsatisfactory learning results. To learn TPP models robustly, we further introduce two regularizers with the help of noise contrastive estimation (Gutmann and Hyvärinen 2012). In principle, the two regularizers aim at maximizing the discrepancy within an event sequence at the event level and the discrepancy across different sequences, respectively.

Event-Level Contrastive Loss

In the training phase, the event type associated with each timestamp is known. Therefore, for each event (t, c) and its history $\boldsymbol{s}_t$, the intensity vector $\boldsymbol{\lambda}(t)$ obtained has included both "positive" and "negative" samples — the c-th element $\lambda_c(t)$ is the intensity we should enlarge while the other elements are the values we should suppress. As a result, we can obtain an event-level contrastive loss:

$$\mathcal{L}_{\text{event}}(\boldsymbol{\lambda}(t))=\log\underbrace{\frac{\lambda_c(t)}{\lambda(t)}}_{p(c|t,\mathcal{H}_t^{\mathcal{C}})}+\sum_{c'\neq c}\log\Big(1-\underbrace{\frac{\lambda_{c'}(t)}{\lambda(t)}}_{p(c'|t,\mathcal{H}_t^{\mathcal{C}})}\Big), \quad (9)$$

where $\lambda(t)=\sum_{c=1}^{C}\lambda_c(t)$ is the overall intensity at time t. Each $\lambda_c(t)/\lambda(t)$ corresponds to the probability of the event type c conditioned on current time t and the history $\mathcal{H}_t^{\mathcal{C}}$.

Beyond the discriminative learning method in (Xu et al. 2016), which only maximizes the conditional probability of the "positive" event type, our event-level contrastive loss further penalizes the conditional probabilities of those "negative" event types. Furthermore, following the sampling strategies used in the original NCE (Gutmann and Hyvärinen 2012), we don't have to enumerate all negative event types when the number of event types is large. Instead, we can sample a subset of negative event types randomly and rewrite the event-level contrastive loss as follows:

$$\mathcal{L}_{\text{event}}(\boldsymbol{\lambda}(t))=\log\frac{\lambda_c(t)}{\lambda'(t)}+\sum\nolimits_{c'\sim\mathcal{C}'}\log\Big(1-\frac{\lambda_{c'}(t)}{\lambda'(t)}\Big), \quad (10)$$

where $\lambda'(t)=\sum\nolimits_{c'\sim\mathcal{C}'}\lambda_{c'}(t)$, and $\mathcal{C}'=\mathcal{C}\setminus\{c\}$.

Here, $c'\sim\mathcal{C}'$ means sampling a small subset randomly from the remaining negative event types. As a result, the event-level contrastive loss suppresses the uncertainty of the observed event sequence by enlarging the conditional probability of the positive event and reducing that of the negative event at the same time, which is more efficient than the discriminative learning method in (Xu et al. 2016).

Note that our event-level contrastive loss is different from the NCE loss used in (Guo, Li, and Liu 2018; Mei, Wan, and Eisner 2020), which neither depends on time-consuming simulation algorithms (Ogata 1981) to generate negative events nor requires a pretrained/predefined TPP model as the reference. Therefore, our learning method is easy to implement and has advantages in efficiency.

Sequence-Level Contrastive Loss

When learning a TPP model, the event sequences, rather than the individual events within each of them, work as the samples of the model. For the TPP model, whose event types have temporal dependencies, the likelihood of observing a sequence is different from that of observing individual events independently at the corresponding timestamps. Therefore, besides constructing contrastive loss for event types, we further propose the contrastive loss at the sequence level in this study and implement it efficiently.

The proposed sequence-level contrastive loss requires generating some "negative" event sequences based on the observed ones. Instead of applying Ogata's thinning method (Ogata 1981), we propose a simple but efficient simulation method called **model-guided thinning**. In principle, in the training phase, this method first estimates the significance of each event in an observed sequence based on the TPP model at the current stage. Then, it generates positive and negative event sequences by thinning the observed events based on their significance.

In this study, we obtain the significance of events based on the infectivity among them. The infectivity matrix $\boldsymbol{G}$ derived by the model f_θ indicates the events that have a huge influence on their followers. Taking $\boldsymbol{G}$ as an input, we can define the following function to obtain the significance of observed events quantitatively, i.e.,

$$\boldsymbol{g}=[g_l]_{l=1}^{L}=\text{softmax}(\boldsymbol{G}^T\mathbf{1}_L). \quad (11)$$

Here, $\boldsymbol{G}^T\mathbf{1}_L$ accumulates the influence of each event on the future events, which provides us with strong evidence for the significance of events. The softmax operation normalizes the significance accordingly.

According to the definition in (11), the small g_l means that the l-th event has ignorable impacts on future events. Removing it would have a limited influence on the happening of the future event. On the contrary, when g_l is large, the l-th event is significant to the future events, and accordingly, removing it would break the generative mechanism of the future events. As illustrated in Figure 1(b), we can sample a set of negative event sequences by removing the events randomly while sampling a positive event sequence by yielding the significance $\boldsymbol{g}$ and removing insignificant events with higher probabilities. Assuming that the embeddings of similar sequences will be close to each other as well, we construct the following sequence-level contrastive loss:

$$\begin{aligned}&\mathcal{L}_{\text{seq}}(\boldsymbol{s},\boldsymbol{s}_{(P)},\{\boldsymbol{s}_{k,(N)}\}_{k=1}^{K})\\&=\log\frac{e^{\mathbf{e}^T\boldsymbol{e}_{(P)}}}{e^{\mathbf{e}^T\boldsymbol{e}_{(P)}}+\sum_{k=1}^{K}e^{\mathbf{e}^T\boldsymbol{e}_{k,(N)}}}+\\&\sum_{k=1}^{K}\log\Big(1-\frac{e^{\mathbf{e}^T\boldsymbol{e}_{k,(N)}}}{e^{\mathbf{e}^T\boldsymbol{e}_{(P)}}+\sum_{k'=1}^{K}e^{\mathbf{e}^T\boldsymbol{e}_{k',(N)}}}\Big),\end{aligned} \quad (12)$$

Algorithm 1: The MLE+HCL method of TPPs

```
**Input:** A event sequence set $\mathcal{S}$. **Output:** A TPP $f_\theta$
Initialize the model parameter $\theta$ randomly.
for A batch of sequences $\mathcal{B} \subset \mathcal{S}$ do
    for Each $\boldsymbol{s} \in \mathcal{B}$ do
        Calculate $\mathcal{L}(\boldsymbol{s}; \theta)$ via (8).
        for Each event $(t_i, c_i) \in \boldsymbol{s}$ do
            Sample negative events $\mathcal{C}'$, get $\mathcal{L}_{\text{event}}(\boldsymbol{\lambda}(t_i))$.
        end for
        Sample sequences, get $\mathcal{L}_{\text{seq}}(\boldsymbol{e}, \boldsymbol{e}_{(P)}, \{\boldsymbol{e}_{k,(N)}\}_k)$.
        Construct the loss function via (13).
    end for
    Apply Adam (Kingma and Ba 2014) to update $f_\theta$.
end for
```

where we denote $\boldsymbol{s}$ as the original observed sequence, $\boldsymbol{s}_{(P)}$ as the positive sequence, and $\boldsymbol{s}_{k,(N)}$ as the k-th negative sequence. Accordingly, $\boldsymbol{e}$, $\boldsymbol{e}_{(P)}$, and $\boldsymbol{e}_{k,(N)}$ represent the embeddings of $\boldsymbol{s}$, $\boldsymbol{s}_{(P)}$, and $\boldsymbol{s}_{k,(N)}$.

Compared to Ogata's thinning algorithm (Ogata 1981), which is applied in other contrastive learning methods (Mei, Wan, and Eisner 2020) for generating negative samples, our model-guided thinning is much more efficient.. In theory, to generate a negative sequence with $\mathcal{O}(N)$ events, the complexity of Ogata's thinning is $\mathcal{O}(N^2)$ because each event is generated according to its history. On the contrary, the complexity of our thinning method is at most $\mathcal{O}(N)$ because its sampling process is memoryless — for each event, it just determines whether preserve it or not.

Hierarchical Contrastive Learning of TPPs

Taking the above two contrastive losses into account, we obtain a hierarchical contrastive learning (HCL) method to regularize the MLE of TPPs. In summary, given an event sequence before time t (i.e., $\boldsymbol{s}_t$ with length L) and the event at time t (i.e., (t, c)), our learning problem is

$$\begin{aligned}\max_{f_\theta}\ \underbrace{\mathcal{L}(\boldsymbol{s};\theta)}_{\text{Log-likelihood}} &+ \underbrace{\gamma_1 \sum\nolimits_{i=1}^{L} \mathcal{L}_{\text{event}}(\boldsymbol{\lambda}(t_i))}_{\text{Event-level contrastive loss}} + \\ &\underbrace{\gamma_2 \mathcal{L}_{\text{seq}}(\boldsymbol{s}, \boldsymbol{s}_{(P)}, \{\boldsymbol{s}_{k,(N)}\}_{k=1}^{K})}_{\text{Sequence-level contrastive loss}},\end{aligned} \tag{13}$$

where $\mathcal{L}(\boldsymbol{s};\theta)$ is the log-likelihood term in (8), $\mathcal{L}_{\text{event}}(\boldsymbol{\lambda}(t_i))$ is the event-level contrastive loss in (10), and $\mathcal{L}_{\text{seq}}(\boldsymbol{e}, \boldsymbol{e}_{(P)}, \{\boldsymbol{e}_{k,(N)}\}_{k=1}^{K})$ is the sequence-level contrastive loss in (12).

This learning problem can be solved efficiently by stochastic gradient descent (SGD). Algorithm 1 shows the scheme of our learning method in detail.

Experiments

To demonstrate the usefulness of our HCL method, we evaluate it on several synthetic and real-world datasets, with a comparison to representative learning methods of TPPs. The following experimental results show the superiority of our HCL method, and we further do ablation studies and analyze the robustness of the method to its hyperparameters. All the experiments are run on a server with two Nvidia 3090 GPUs.

Dataset	# Types	# Sequences	Max. length
Hawkes	5	10000	100
Missing	5	10000	100
Retweet	3	24000	264
StackOverflow	22	6633	736
Bookorder	2	200	3319

Table 1: The statistics of datasets

Implementation Details

Datasets We considered five datasets, whose statistics are given in Table 1 and details are summarized below:

Hawkes and **Missing** are synthetic data provided by (Mei, Qin, and Eisner 2019). Each of these two datasets contains ten thousand event sequences yielding 5-dimensional Hawkes processes. The event sequences in the Missing dataset are censored randomly, which imitate the real-world scenarios with missing events.

Retweet (Zhao et al. 2015): The Retweet dataset, including 24000 sequences, is collected from Twitter, a social website which is composed of sequences of tweets. Each tweet is treated as an event, which contains a timestamp and a user group tag based on the number of the user's followers.

Stack Overflow: The Stack Overflow's open source dataset contains two years of users' reward history on a question-answering website. Each sequence represents a user's reward history and each reward(i.e., event) contains a timestamp and a badge(i.e., event type).

BookOrder is collated from (Mei, Qin, and Eisner 2019). The maximum length of this sequence dataset is up to more than three thousand, implying difficulties in storage and training computational overhead.

Baselines and Hyperparameter Settings We test our method (**MLE+HCL**) on each of the above datasets and compare it with the following baselines. The baselines can be categorized into two categories: $i)$ the MLE-based methods, including the MLE with sparse regularization (**MLE+Reg**) (Zhou, Zha, and Song 2013b), the MLE with superposition-based data augmentation (**MLE+DA**) (Xu et al. 2018); $ii)$ the non-MLE methods, including the discriminative learning (**DIS**) method in (Xiao et al. 2017b)), and the contrastive learning methods **INITIATOR** (Guo, Li, and Liu 2018) and **NCE-TPP** (Mei, Wan, and Eisner 2020). For all the methods, we set the learning rate as 0.001, the batch size as 1 for the Bookorder dataset (because it owns extremely long event sequences) and 4 for the remaining datasets, and the dimension of sequence embedding as $D = 64$. For NCE-TPP, we follow its default setting, applying the neural Hawkes process as the noisy point process to generate negative samples. For MLE+HCL, we set the number of negative sequences as $K = 20$, whose rationality is given in the following analytic experiments.

Backbone Models To demonstrate the universality of our learning method, we apply two different TPPs to model each

Models	Data	Metrics	Methods					
			MLE+Reg	MLE+DA	DIS	INITIATOR	NCE-TPP	MLE+HCL
HP	Hawkes	Log-Like	-0.06 (0.05)	-0.52 (0.34)	-6.6 (1.11)	-0.22 (0.02)	-0.10 (0.08)	**-0.04 (0.05)**
		Type-Acc	0.38 (0.01)	0.38 (0.01)	0.32 (0.05)	0.35 (0.02)	0.33 (0.02)	**0.40 (0.00)**
	Missing	Log-Like	-0.06 (0.00)	-1.09 (0.00)	-3.38 (0.00)	-0.53 (0.04)	-0.04 (0.01)	**-0.02 (0.00)**
		Type-Acc	**0.42 (0.00)**	0.41 (0.01)	0.40 (0.01)	0.38 (0.02)	0.41 (0.01)	**0.42 (0.02)**
	Bookorder	Log-Like	-2.60 (0.40)	-3.28 (0.58)	-1.64 (0.36)	-1.71 (0.25)	-1.60 (0.38)	**-1.57 (0.30)**
		Type-Acc	0.57 (0.00)	0.57 (0.01)	**0.62 (0.01)**	0.60 (0.02)	**0.62 (0.00)**	**0.62 (0.00)**
	StackOverflow	Log-Like	-0.77 (0.01)	-2.31 (0.12)	-0.96 (0.07)	-0.79 (0.03)	-0.74 (0.04)	**-0.72 (0.01)**
		Type-Acc	0.45 (0.02)	0.43 (0.02)	0.40 (0.05)	0.49 (0.02)	**0.51 (0.06)**	0.50 (0.01)
	Retweet	Log-Like	-8.94 (0.20)	-10.73 (2.20)	—	-8.89 (0.11)	-8.92 (0.05)	**-8.84 (0.02)**
		Type-Acc	0.60 (0.01)	0.59 (0.00)	0.58 (0.02)	0.62 (0.05)	**0.66 (0.03)**	**0.66 (0.04)**
THP	Hawkes	Log-Like	0.11 (0.03)	-1.23 (0.43)	-0.68 (0.13)	0.03 (0.05)	0.12 (0.01)	**0.14 (0.02)**
		Type-Acc	0.38 (0.00)	0.26 (0.00)	0.34 (0.03)	0.24 (0.02)	**0.40 (0.01)**	0.38 (0.00)
	Missing	Log-Like	-0.47 (0.01)	-1.32 (0.21)	-0.75 (0.16)	-1.08 (0.10)	-0.50 (0.02)	**-0.34 (0.08)**
		Type-Acc	0.41 (0.01)	0.27 (0.00)	0.41 (0.00)	0.40 (0.00)	**0.42 (0.01)**	0.41 (0.00)
	Bookorder	Log-Like	-1.69 (0.31)	-4.50 (1.79)	-1.64 (0.36)	-1.70 (0.43)	-1.60 (0.30)	**-1.58 (0.21)**
		Type-Acc	0.62 (0.00)	0.62 (0.01)	0.62 (0.00)	0.53 (0.01)	**0.64 (0.00)**	**0.64 (0.00)**
	StackOverflow	Log-Like	**-0.77 (0.00)**	-2.31 (0.12)	-0.96 (0.07)	-0.89 (0.10)	**-0.77 (0.02)**	-0.79 (0.01)
		Type-Acc	0.43 (0.00)	0.42 (0.02)	0.49 (0.03)	0.40 (0.05)	0.39 (0.02)	**0.44 (0.01)**
	Retweet	Log-Like	-7.35 (0.35)	-9.14 (0.46)	—	-10.20 (0.53)	-7.33 (0.29)	**-7.27 (0.18)**
		Type-Acc	0.53 (0.00)	0.50 (0.01)	0.54 (0.00)	0.53 (0.01)	0.54 (0.00)	**0.56 (0.00)**

* “—” means the learning method fails to converge. The best results are bolden. In each cell, the averaged performance is shown, and the parentheses contains the standard deviation.

Table 2: Comparisons for various methods on learning TPPs from different datasets

of the above datasets and learn the two models via various methods. The first TPP is the Hawkes process (HP) with exponential impact functions, which has been commonly used in many works (Zhou, Zha, and Song 2013b; Bacry, Mastromatteo, and Muzy 2015). The second TPP is the transformer Hawkes process (THP) proposed in (Zuo et al. 2020), which is one of the state-of-the-art TPP models and achieves encouraging performance in various applications. Essentially, the THP can be viewed as an extension of the classic HP — the triggering patterns captured by the classic HP are parametrized by the transformer architecture of the THP. Accordingly, the infectivity between different events is described by the self-attention mechanism.

Evaluation Measurements Given a dataset and a backbone model, we train the model via a learning method with 5-fold cross validation. In each trial, given the testing event sequences, we consider two evaluation measurements: i) the log-likelihood per event (**Log-Like**); ii) the prediction accuracy of event types (**Type-Acc**). These two measurements evaluate the performance of the learning method in two aspects: the testing log-likelihood indicates the data fidelity achieved by the learning method, while the prediction accuracy shows the prediction power of the learned model. For each measurement, its averaged value and standard deviation are recorded.

Comparison Experiments

Applying various learning methods to train backbone models, we obtain the learning results shown in Table 2. We can find that the superiority of our MLE+HCL method is consistent on all the datasets and robust to the selection of backbone models — for each of the backbone models, our MLE+HCL method outperforms the baselines in most situations. In particular, for the MLE-based methods, our HCL regularizer works better than the sparse regularizer and the data augmentation method, which achieves higher testing log-likelihood and prediction accuracy. Additionally, compared to the data augmentation method, in which the random superposition increases the uncertainty of data, our MLE+HCL method achieves more minor standard deviation.

Compared to other non-MLE methods, our MLE+HCL approach owns better performance and efficiency. As shown in Table 2, our MLE+HCL is at least comparable to the state-of-the-art contrastive learning methods (i.e., INITIATOR and NCE-TPP) on testing log-likelihood and prediction accuracy. Additionally, the discriminative learning method in (Xu et al. 2016) fails to converge when dealing with the Retweet dataset because of the lack of the log-likelihood term. Leveraging the advantages of both MLE and contrastive learning, our MLE+HCL works better on both performance and stability.

Another advantage of our HCL method is its efficiency. As aforementioned, the most time-consuming step of previous contrastive learning methods (Mei, Wan, and Eisner 2020; Guo, Li, and Liu 2018) is generating samples because they rely on Ogata’s thinning algorithm (Ogata 1981). On the contrary, our HCL applies a model-guided thinning method. The event sequences are generated by a simple categorical sampling process, which is much faster than Ogata’s thinning algorithm. Figure 2 compares the two thinning methods on the runtime of generating one event sequence for a Hawkes process. We can see that with the increase of the number of events, the advantage of our thinning method

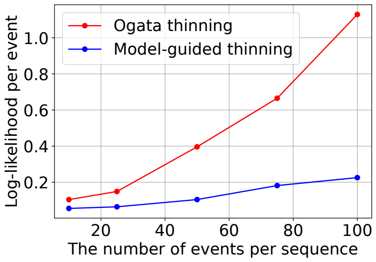
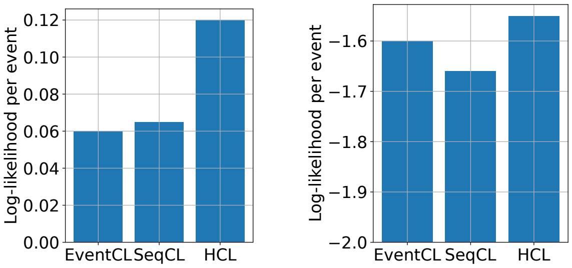


Figure 2: The runtime comparison on the two thinning methods. For our model-guided thinning method, the runtime of constructing the infectivity matrix is taken into account.



(a) Synthetic Hawkes data (b) Real-world BookOrder data

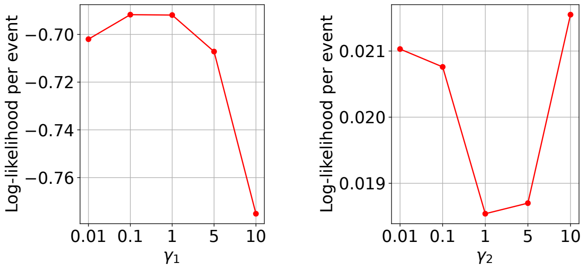
Figure 3: The ablation study of our HCL method.

becomes more and more obvious.

Analytic Experiments

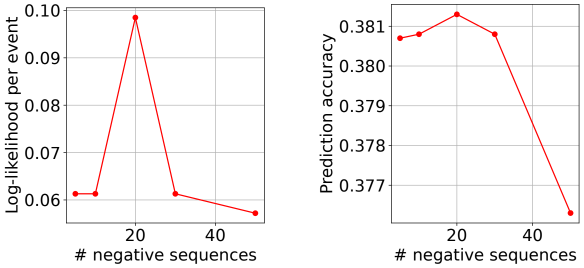
Ablation Study Our HCL method takes both the event-level contrastive loss and the sequence-level one into account. To demonstrate the necessity of these two losses, we perform an ablation study on both synthetic and real-world datasets. Taking the testing log-likelihood as the measure, we check the influence of the event-level loss and that of the sequence-level loss and compare them with the proposed HCL regularizer. The results are shown in Figure 3. We can find that considering only one contrastive loss leads to the degradation of performance. The HCL considering the two losses jointly, achieves the highest testing log-likelihood.

Robustness Analysis Our HCL method owns three key hyperparameters: the weights of the two contrastive losses, i.e., γ_1 and γ_2, and the number of negative sequences K. We test the robustness of our HCL method to the two weights and show the analytic results in Figure 4. We can find that results of our HCL method are relatively stable when γ_1 and γ_2 change in a wide range. For the number of negative sequences per sample, we show its influence on the learning results in Figure 5. When K is small, the negative sequences may have poor diversity and can be close to the original sequence because of the randomness of sampling. When K is large, the negative sequences will be dominant compared to the positive sequence. As shown in Figure 5, for both testing log-likelihood and prediction accuracy, the best performance is achieved when applying the default setting $K = 20$, which achieves a trade-off between the diversity and the redundancy of negative sequences.



(a) StackOverflow ($\gamma_2 = 1$) (b) Missing data ($\gamma_1 = 1$)

Figure 4: The robustness of our HCL method to the weights of the contrastive losses.



(a) Log-likelihood per event (b) Prediction accuracy

Figure 5: The influence of the number of negative event sequences. The dataset is the synthetic Hawkes dataset.

Conclusion

In this paper, we present a hierarchical contrastive learning method for temporal point processes, which provides a new regularizer for the scheme of maximum likelihood estimation. The proposed method not only considers the event-level contrastive learning like existing work (Guo, Li, and Liu 2018; Xu et al. 2016) did, but also designs a simple but effective sequence-level contrastive loss guided by the triggering pattern hidden behind the target model. The contrastive learning mechanism, especially the sequence-level part, is more efficient than the Ogata's thinning-based method (Mei, Wan, and Eisner 2020). The complexity of our thinning method is $\mathcal{O}(N)$ and can be $\mathcal{O}(1)$ in parallel while the complexity of the Ogata's thinning is $\mathcal{O}(N^2)$. Beyond the scheme of MLE, we would like to combine the proposed HCL method with other learning frameworks in the future, e.g., the Wasserstein GAN strategy (Xiao et al. 2017a) and the reinforcement learning strategy (Li et al. 2018; Zhu et al. 2021). Additionally, we will try to find theoretical support for our HCL method.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No. 62106271), CAAI-Huawei MindSpore Open Fund, the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China (No. 22XNKJ06, 22XNH033). Dr. Hongteng Xu also would like to thank the support from the Beijing Key Laboratory of Big Data Management and Analysis Methods.

References

Bacry, E.; Mastromatteo, I.; and Muzy, J.-F. 2015. Hawkes processes in finance. *Market Microstructure and Liquidity*, 1(01): 1550005.

Du, N.; Dai, H.; Trivedi, R.; Upadhyay, U.; Gomez-Rodriguez, M.; and Song, L. 2016. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1555–1564.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2*, 2672–2680.

Guo, R.; Li, J.; and Liu, H. 2018. INITIATOR: noise-contrastive estimation for marked temporal point process. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2191–2197.

Gutmann, M. U.; and Hyvärinen, A. 2012. Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics. *Journal of machine learning research*, 13(2).

Hawkes, A. G. 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1): 83–90.

Isham, V.; and Westcott, M. 1979. A self-correcting point process. *Stochastic processes and their applications*, 8(3): 335–347.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingman, J. F. C. 1992. *Poisson processes*, volume 3. Clarendon Press.

Lewis, P. W.; and Shedler, G. S. 1979. Simulation of nonhomogeneous Poisson processes by thinning. *Naval research logistics quarterly*, 26(3): 403–413.

Li, S.; Xiao, S.; Zhu, S.; Du, N.; Xie, Y.; and Song, L. 2018. Learning temporal point processes via reinforcement learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 10804–10814.

Liniger, T. J. 2009. *Multivariate hawkes processes*. Ph.D. thesis, ETH Zurich.

Mei, H.; and Eisner, J. 2017. The neural hawkes process: a neurally self-modulating multivariate point process. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6757–6767.

Mei, H.; Qin, G.; and Eisner, J. 2019. Imputing missing events in continuous-time event streams. In *International Conference on Machine Learning*, 4475–4485. PMLR.

Mei, H.; Wan, T.; and Eisner, J. 2020. Noise-contrastive estimation for multivariate point processes. *Advances in neural information processing systems*, 33: 5204–5214.

Noorbakhsh, K.; and Rodriguez, M. G. 2021. Counterfactual Temporal Point Processes. *arXiv preprint arXiv:2111.07603*.

Ogata, Y. 1981. On Lewis' simulation method for point processes. *IEEE transactions on information theory*, 27(1): 23–31.

Ogata, Y.; et al. 1978. The asymptotic behaviour of maximum likelihood estimators for stationary point processes. *Annals of the Institute of Statistical Mathematics*, 30(1): 243–261.

Ozaki, T. 1979. Maximum likelihood estimation of Hawkes' self-exciting point processes. *Annals of the Institute of Statistical Mathematics*, 31(1): 145–155.

Rambaldi, M.; Bacry, E.; and Lillo, F. 2017. The role of volume in order book dynamics: a multivariate Hawkes process analysis. *Quantitative Finance*, 17(7): 999–1020.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Upadhyay, U.; De, A.; and Gomez-Rodrizuez, M. 2018. Deep reinforcement learning of marked temporal point processes. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 3172–3182.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Xiao, S.; Farajtabar, M.; Ye, X.; Yan, J.; Song, L.; and Zha, H. 2017a. Wasserstein learning of deep generative point process models. *Advances in neural information processing systems*, 30.

Xiao, S.; Xu, H.; Yan, J.; Farajtabar, M.; Yang, X.; Song, L.; and Zha, H. 2018. Learning conditional generative models for temporal point processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Xiao, S.; Yan, J.; Yang, X.; Zha, H.; and Chu, S. 2017b. Modeling the intensity function of point process via recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Xu, H.; Carin, L.; and Zha, H. 2018. Learning registered point processes from idiosyncratic observations. In *International Conference on Machine Learning*, 5443–5452. PMLR.

Xu, H.; Farajtabar, M.; and Zha, H. 2016. Learning granger causality for hawkes processes. In *International Conference on Machine Learning*, 1717–1726. PMLR.

Xu, H.; Luo, D.; Chen, X.; and Carin, L. 2018. Benefits from superposed hawkes processes. In *International Conference on Artificial Intelligence and Statistics*, 623–631. PMLR.

Xu, H.; Luo, D.; and Zha, H. 2017. Learning hawkes processes from short doubly-censored event sequences. In *International Conference on Machine Learning*, 3831–3840. PMLR.

Xu, H.; Wu, W.; Nemati, S.; and Zha, H. 2016. Patient flow prediction via discriminative learning of mutually-correcting processes. *IEEE transactions on Knowledge and Data Engineering*, 29(1): 157–171.

Yan, J.; Liu, X.; Shi, L.; Li, C.; and Zha, H. 2018. Improving maximum likelihood estimation of temporal point process via discriminative and adversarial learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2948–2954.

Zhang, Q.; Lipani, A.; Kirnap, O.; and Yilmaz, E. 2020. Self-attentive Hawkes process. In *International conference on machine learning*, 11183–11193. PMLR.

Zhang, Q.; Lipani, A.; and Yilmaz, E. 2021. Learning Neural Point Processes with Latent Graphs. In *Proceedings of the Web Conference 2021*, 1495–1505.

Zhao, Q.; Erdogdu, M. A.; He, H. Y.; Rajaraman, A.; and Leskovec, J. 2015. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 1513–1522.

Zhou, K.; Zha, H.; and Song, L. 2013a. Learning triggering kernels for multi-dimensional hawkes processes. In *International conference on machine learning*, 1301–1309. PMLR.

Zhou, K. e.; Zha, H.; and Song, L. 2013b. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *Artificial Intelligence and Statistics*, 641–649. PMLR.

Zhu, S.; Li, S.; Peng, Z.; and Xie, Y. 2021. Imitation learning of neural spatio-temporal point processes. *IEEE Transactions on Knowledge and Data Engineering*.

Zuo, S.; Jiang, H.; Li, Z.; Zhao, T.; and Zha, H. 2020. Transformer hawkes process. In *International conference on machine learning*, 11692–11702. PMLR.