Self-Attentive Hawkes Processes

Qiang Zhang
University College London, United Kingdom
qiang.zhang.16@ucl.ac.uk

Aldo Lipani
University College London, United Kingdom
aldo.lipani@ucl.ac.uk

Omer Kirnap
University College London, United Kingdom
omer.kirnap.18@ucl.ac.uk

Emine Yilmaz
University College London, United Kingdom
emine.yilmaz@ucl.ac.uk

Abstract

Asynchronous events on the continuous time domain, e.g., social media actions and stock transactions, occur frequently in the world. The ability to recognize occurrence patterns of event sequences is crucial to predict *which type* of events will happen next and *when*. A *de facto* standard mathematical framework to do this is the Hawkes process. In order to enhance expressivity of multivariate Hawkes processes, conventional statistical methods and deep recurrent networks have been employed to modify its intensity function. The former is highly interpretable and requires small size of training data but relies on correct model design while the latter has less dependency on prior knowledge and is more powerful in capturing complicated patterns. We leverage pros and cons of these models and propose a *self-attentive Hawkes process* (SAHP). The proposed method adapts self-attention to fit the intensity function of Hawkes processes. This design has two benefits: (1) compared with conventional statistical methods, the SAHP is more powerful to identify complicated dependency relationships between temporal events; (2) compared with deep recurrent networks, the self-attention mechanism is able to capture longer historical information, and is more interpretable because the learnt attention weight tensor shows contributions of each historical event. Experiments on four real-world datasets demonstrate the effectiveness of the proposed method.

1 Introduction

Frequently, human need to tackle a large amount of irregular and asynchronous event sequences. These sequences can be, for example, user activities on social media platforms (Farajtabar et al., 2015), high-frequency financial transactions (Bacry and Muzy, 2014), healthcare records (Wang et al., 2016), gene positions in bioinformatics (Reynaud-Bouret et al., 2010), or earthquakes and aftershocks in geophysics (Ogata, 1998). These sequences are multi-dimensional and asynchronous. Different from discrete time series with equal sampling intervals, asynchronous event sequences have continuous timestamps. Events usually have correlation and can mutually influence each other: the occurrence of one type of event at a certain timestamp can cause or prevent the happening of future events of the same or another type. Fig. 1 shows different activities of three users on social media platforms and their mutual influence. Mining correlation among asynchronous event sequences paves the way to predict future and identify causality.

Temporal point processes are used to characterize asynchronous event sequences on the continuous time domain (Cox and Isham, 1980; Brillinger et al., 2002). They are stochastic processes with (marked) events on the continuous time domain. Point processes characterize the occurrence probability of an event with the so-called *intensity function*. One type of temporal point process is the Hawkes process, which assumes a history-dependent intensity function. The Hawkes process uses

Preprint. Under review.

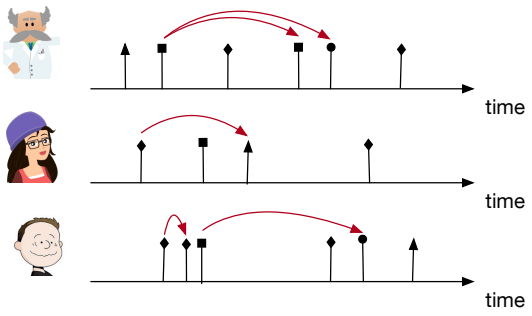

Figure 1: Three users on social media platforms exert different types of actions. The dark symbols mean different action types while the red arrows denote that one action is influencing another action.

an auto-regressive structure of the intensity to capture self-excitation and mutual-excitation among events, enabling to fully consider the influence of all historical events to predict future events. The Hawkes process is a *de facto* standard mathematical tool to model event streams, including for topic modeling and clustering of text document streams (He et al., 2015; Du et al., 2015a), constructing and inferring network structure (Yang and Zha, 2013; Choi et al., 2015; Etesami et al., 2016), personalized recommendations based on users' temporal behavior (Du et al., 2015b), discovering patterns in social interaction (Guo et al., 2015; Lukasik et al., 2016) and learning causality (Xu et al., 2016).

Given a sequence of asynchronous events, the Hawkes process is often used to predict subsequent events. Hawkes processes have been developed in statistics (Xu et al., 2016; Achab et al., 2017; Yang et al., 2017) and in deep learning (Du et al., 2016; Xiao et al., 2017b; Mei and Eisner, 2017). Statistical models require a pre-defined intensity function, of which parameters are usually interpretable. However, setting the form of the intensity function requires prior-knowledge on the domain. In deep learning, instead this prior-knowledge is not needed. By using more data, deep learning models mine complicated hidden patterns, which lead to higher prediction performance. However, existing deep learning approaches are based on recurrent architectures that are less interpretable than statistical models.

In this paper, to enhance the interpretability of neural Hawkes processes and at the same time increase their prediction performance, we propose a Self-Attentive Hawkes Process (SAHP). Here, the self-attention mechanism is adapted to fit the intensity function. The proposed model is able to capture longer historical dependencies than recurrent neural counterparts. For model interpretability, the learnt attention weights indicate the contribution of one event type to predicting another. Through extensive experiments on four real-world datasets with different sequence lengths and different number of event types, we demonstrate the effectiveness in terms of prediction performance of the proposed model against the state-of-the-art.

2 Preliminary

2.1 Temporal Point Processes and Hawkes Processes

Temporal point processes. A temporal point process is a stochastic process whose realization is a list of discrete events at time $t_i \in \mathbb{R}^+$ with $i \in \mathbb{Z}^+$ (Cox and Isham, 1980; Daley and Vere-Jones, 2007). It can be equivalently represented as a counting process $N(t)$, which records the number of events that have happened until time t. A multivariate point process describes the temporal evolution of multiple event types $\mathcal{U} = \{1, \dots, U\}$. Let $\mathcal{S} = \{(v_i, t_i)\}_{i=1}^{L}$ be an event sequence where the tuple (v_i, t_i) is the i-th event of the sequence $\mathcal{S}$, $v_i \in \mathcal{U}$ is the event type, and t_i is the timestamp of the i-th event. One typical way to characterize point processes is via an intensity function $\lambda(t)$.

Hawkes processes. An Hawkes process (Hawkes, 1971) is a temporal point process with history-dependent intensity $\lambda(t) = \lambda(t|\mathcal{H}(t))$, where $\mathcal{H}(t) := \{(v_i, t_i) \,|t_i < t, v_i \in \mathcal{U}\}$ is the set of historical events before time t. The intensity function of a type-u event is defined as the conditional probability on the history $\mathcal{H}(t)$ that this event has not happened before t but will happen during $[t, t + dt)$. The definition is given as:

$$\lambda_u(t) = \mu_u + \sum_{(v_i,t_i)\in\mathcal{H}(t)} \phi(t - t_i), \tag{1}$$

where $\mu_u \geq 0$ (aka *base intensity*) is an exogenous component of the intensity function independent of the history, while $\phi(t) > 0$ is an endogenous component of the intensity function dependent on the history. Besides, $\phi(t)$ contains the peer influence of different event types. To highlight the peer influence represented by $\phi(t)$, we write $\phi_{u,u'}(t)$, which captures the impact of a historical type-u' event on a subsequent type-u event (Farajtabar et al., 2014). In this example, the occurrence of a past type-u' event increases the intensity function $\phi_{u,u'}(t-\tau)$ for $0 < \tau < t$.

Most commonly $\phi_{u,u'}(t)$ is parameterized as $\phi_{u,u'}(t) = \alpha_{u,u'}\kappa(t)\mathbb{1}(t > 0)$ (Zhou et al., 2013; Xu et al., 2016). The *excitation* parameter $\alpha_{u,u'}$ quantifies the initial influence of the type-u' event on the intensity of the type-u event. The *kick* function $\kappa(t)$ characterizes the time-decaying influence. Typically, $\kappa(t)$ is chosen to be exponential, i.e., $\kappa(t) = \exp(-\omega t)$, where ω is the *decaying* parameter controlling the intensity decaying speed.

To learn the intensity function, both statistical and neural methods have been developed. Parametric statistical methods pre-define the form of the intensity function, which makes it interpretable. However, these methods perform poorly when there is a mismatch between the pre-defined form and the true intensity pattern. Non-parametric statistical methods due to the need to store the full history of events are more complex and inconvenient for practical applications. Neural methods do not suffer from the pitfalls of the statistical methods. Existing neural Hawkes processes adapt a recurrent structure to learn the parameters of the intensity function. However, they need more training data than statistical methods.

2.2 Attention and Self-Attention

Attention. The attention mechanism enables machine learning models to focus on a subset of the input (Walther et al., 2004; Bahdanau et al., 2014). In Seq2Seq models with the attention mechanism the input sequence, in the encoder, is represented with a sequence of key vectors K and value vectors V, $(K, V) = [(\boldsymbol{k}_1, \boldsymbol{v}_1), (\boldsymbol{k}_2, \boldsymbol{v}_2), \ldots, (\boldsymbol{k}_N, \boldsymbol{v}_N)]$. While, the decoder side of the Seq2Seq model uses query vectors, $Q = [\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_M]$. These query vectors are used to find which part of the input sequence is more contributory (Vaswani et al., 2017). Given these two sequences of vectors (K, V) and Q, the attention mechanism computes a sequence of predictions $O = [\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_M]$ as follows:

$$\boldsymbol{o}_m = \left(\sum_n f(\boldsymbol{q}_m, \boldsymbol{k}_n) g(\boldsymbol{v}_n) \right) / \sum_n f(\boldsymbol{q}_m, \boldsymbol{k}_n), \tag{2}$$

where $m \in \{1, \ldots, M\}$, $n \in \{1, \ldots, N\}$, $\boldsymbol{q}_m \in \mathbb{R}^d$, $\boldsymbol{k}_n \in \mathbb{R}^d$, $\boldsymbol{v}_n \in \mathbb{R}^p$, $g(\boldsymbol{v}_n) \in \mathbb{R}^q$ and $\boldsymbol{o}_m \in \mathbb{R}^q$. The similarity function $f(\boldsymbol{q}_m, \boldsymbol{k}_n)$ characterizes the relation between $\boldsymbol{q}_m$ and $\boldsymbol{k}_n$, its common form is composed of: an embedded Gaussian, an inner-product, and a concatenation (Wang et al., 2018). The function $g(\boldsymbol{v}_n)$ is a linear transformation specified as $g(\boldsymbol{v}_n) := \boldsymbol{v}_n W_v$, where $W_v \in \mathbb{R}^{p \times q}$ is a weight matrix.

Self-attention. Self-attention is a special case of the attention mechanism (Vaswani et al., 2017), where the query vectors Q, like (K, V), are from the encoder side. Self-attention is a method of encoding sequences of input events by relating these events to each other based on a pairwise similarity function $f(\cdot, \cdot)$. It measures the dependency between each pair of events from the same input sequence.

Self-attention is very expressive and flexible for both long-term and local dependencies, which used to be modeled by recurrent neural networks (RNNs) and convolutional neural networks (CNNs) (Vaswani et al., 2017). Moreover, the self-attention mechanism has fewer parameters and faster convergence than RNNs. Recently, a variety of Natural Language Processing (NLP) tasks have experienced large improvements thanks to the self-attention mechanism (Vaswani et al., 2017; Devlin et al., 2018).

3 Related Work

Statistical Hawkes processes. Statistical point processes can be categorized as parametric and non-parametric. In a parametric way, Lee et al. (2016) generalize the constant excitation parameters to be stochastic, which increases the performance of the intensity function. Parametric models generally assume a specific form for the intensity function. This limits the model to the characterization

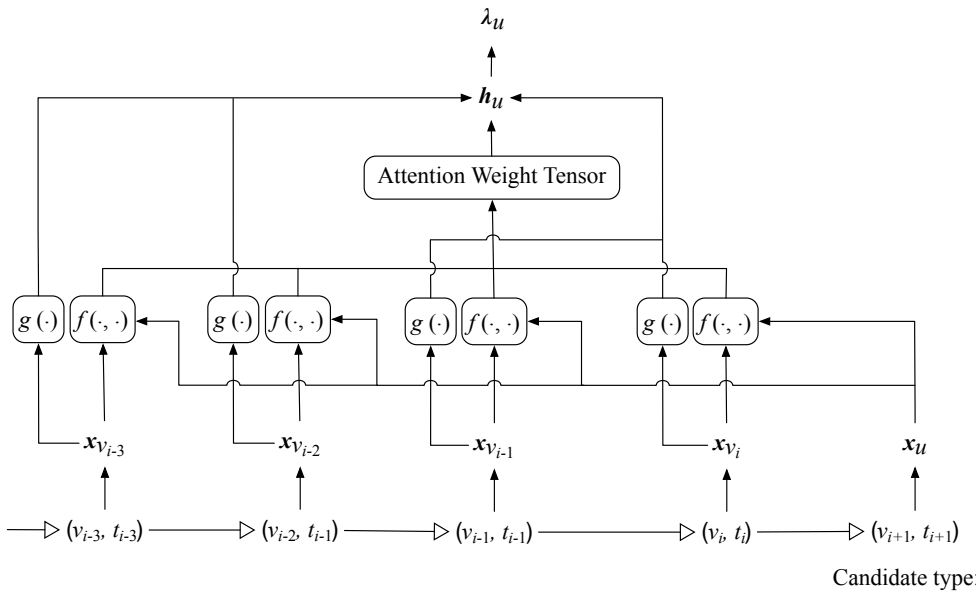

Figure 2: An event stream and the SAHP for one event type (u). The intensity function ($\lambda_u(t)$) is determined by a sequence of past events via the SAHP. The length of each temporal evolution arrow represents the time interval between subsequent events.

of simple scenarios (Achab et al., 2017). To improve flexibility, non-parametric models was first explored by Lewis and Mohler (2011). Another non-parametric strategy formulates non-parametric learning of Hawkes processes as set of Wiener-Hopf systems (Bacry and Muzy, 2016). However, this strategy is computation-costly when the number of event types is large. Alternatively, Hansen et al. (2015) and Xu et al. (2016) decompose kernels on a dictionary of functions $\kappa_1(t), \kappa_2(t), \ldots, \kappa_K(t)$, namely $\phi_{u,u'}(t) = \sum_{k=1}^{K} \alpha_{u,u',k} \kappa_k(t) \mathbb{1}(t > 0)$, where the coefficients $\alpha_{u,u',k}$ are estimated with group-lasso in order to induce a sparsity pattern on the coefficients $\alpha_{u,u',k}$. However, deciding the optimal number of K can be time consuming. Another work proposed by Yang et al. (2017) develops an online non-parametric learning method that limits the excitation to be positive. Our method captures an inhibition effect by applying $\tanh$ so as to expand α to negative. In general, non-parametric methods can be heavily data-dependent and complicated in practical applications.

Neural point processes. Neural networks, especially RNNs, have demonstrated their ability in dealing with sequential information (Graves et al., 2014; Zhang et al., 2018). Recurrent architectures have been proposed to expand expressivity of the intensity function Du et al. (2016) propose a discrete-time RNN to fit the time varying parameters of the intensity function. Mei and Eisner (2017) propose a continuous-time Long Short-Term Memory (LSTM) model that avoids the encoding time intervals between historical events as explicit inputs. However, RNN and its variants have been empirically proved to be less powerful than the self-attention mechanism in NLP (Vaswani et al., 2017; Devlin et al., 2018), which involves various sequential information modeling tasks. Moreover, RNN-modified Hawkes processes do not provide a simple way to interpret the peer influence among events (Karpathy et al., 2015; Krakovna and Doshi-Velez, 2016). On the contrary, our method learns an attention weight tensor, which can quantify the contribution of on event type to predicting another.

4 Self-Attentive Hawkes Process

We implement the self-attention mechanism with an embedded Gaussian to fit the parameters of the intensity function: the base intensity μ_u, the excitation parameter α_u, and the decaying parameter ω_u. To obtain a unique dense embedding for each event type, we use a linear embedding layer, namely $\boldsymbol{x}_u = \boldsymbol{e}_u W_E$, where $\boldsymbol{x}_u$ is the type-u embedding with dimension d_x, $\boldsymbol{e}_u$ is a one-hot vector representing the type-u and W_E is the embedding matrix.

Self-attention. The parameters of the intensity function of the current type-u event are fitted based on the historical events $\mathcal{H}(t)$ as follows:

$$\boldsymbol{h}_u = \left(\sum_{(v_i,t_i)\in\mathcal{H}(t)} f(\boldsymbol{x}_u, \boldsymbol{x}_{v_i}) g(\boldsymbol{x}_{v_i}) \right) / \sum_{(v_i,t_i)\in\mathcal{H}(t)} f(\boldsymbol{x}_u, \boldsymbol{x}_{v_i}), \tag{3}$$

where $\boldsymbol{x}_u$ is like query $\boldsymbol{q}$ in the attention terminology, $\boldsymbol{x}_{v_i}$ is the key $\boldsymbol{k}$ and $g(\boldsymbol{x}_{v_i})$ is the value $\boldsymbol{v}$. The similarity function $f(\cdot,\cdot)$ is specified as an embedded Gaussian:

$$f(\boldsymbol{x}_u, \boldsymbol{x}_{v_i}) = \exp\left(\boldsymbol{x}_u \boldsymbol{x}_{v_i}^T\right), \tag{4}$$

The temporal information is provided to the model, when training, by preventing the model to learn about future events via masking. We implement this inside of the attention mechanism by masking out all values in the input sequence which corresponds to future events. In addition, we add event embeddings with positional encodings that contain event timestamps. The positional encodings are implemented with global sine and cosine functions of different frequencies (Vaswani et al., 2017). Thus, embeddings of the same event-type are treated differently based on their timestamps.

Intensity function. The parameters base, excitation and decaying of the intensity function are computed via the following three non-linear transformation:

$$\mu_u = softplus\left(\boldsymbol{h}_u W_\mu\right), \quad \alpha_u = tanh\left(\boldsymbol{h}_u W_\alpha\right), \quad \omega_u = softplus\left(\boldsymbol{h}_u W_\omega\right). \tag{5}$$

The $softplus$ function is used for both base μ and decaying ω to constrain the value of these parameters to be positive. Moreover, the base intensity is modeled as a function of past events, which can capture the inherent criteria of some events. We use $tanh$ to compress excitation α in the range of $(-1, 1)$. This range of values allows us to capture both excitation and inhibition effects, where with inhibition we mean the effect that manifests when past events reduce the intensity function of future events (Mei and Eisner, 2017).

Finally, we express the intensity function as follows:

$$\lambda_u(t) = softplus\left(\mu_u + \alpha_u \exp(-\omega_u(t - t_i))\right) \text{ for } t \in (t_i, t_{i+1}], \tag{6}$$

where the $softplus$ is employed to constrain the intensity function to be positive.

5 Optimization

Given the history $\mathcal{H}(t_{i+1}) = \{(v_1, t_1), \ldots, (v_i, t_i)\}$, the time density of the subsequent event is calculated as:

$$p_{i+1}(t) = P(t_{i+1} = t | \mathcal{H}(t_{i+1})) = \lambda(t) \exp\left(-\int_{t_i}^{t} \lambda(s) ds\right), \tag{7}$$

where $\lambda(t) = \sum_u \lambda_u(t)$. The prediction of the next event timestamp t_{i+1} is equal to the following expectation:

$$\hat{t}_{i+1} = \mathbb{E}[t_{i+1} | \mathcal{H}(t_{i+1})] = \int_{t_i}^{\infty} t p_{t_{i+1}}(t) dt. \tag{8}$$

While the prediction of the event type is equal to:

$$\hat{u}_{i+1} = \arg\max_{u\in\mathcal{U}} \int_{t_i}^{\infty} \frac{\lambda_u(t)}{\lambda(t)} p_{i+1}(t) dt. \tag{9}$$

Because this integral is not solvable analytically we approximate it via Monte Carlo sampling (Mei and Eisner, 2017).

To learn the parameters of the proposed method, we perform a Maximum Likelihood Estimation (MLE). Other advanced and more complex adversarial learning (Xiao et al., 2017a) and reinforcement learning (Li et al., 2018) methods have been proposed, however we use MLE for its simplicity. We use the same optimization method for our model and all baselines as done in their original papers.

Table 1: Statistics of the used datasets.

Dataset	Event Types U	Sequence Length			# of Event Tokens		
		Min	Mean	Max	Train	Validation	Test
Retweet	3	50	109	264	1,739,547	215,521	218,465
StackOverflow	22	41	72	736	343,998	39,247	97,168
MICMIC-II	75	2	4	33	∼1,946	∼228	∼245
MemeTrack	5,000	1	3	31	∼93,267	∼14,932	∼15,440

To apply MLE, we derive a loss function based on the negative log-likelihood. The likelihood of a multivariate Hawkes process over a time interval $[0, T]$ is given by:

$$\mathcal{L}(\lambda) = \sum_{i=1}^{L} \log \lambda_{v_i}(t_i) - \int_0^T \lambda(\tau) d\tau, \tag{10}$$

where the first term is the sum of the log-intensity functions of past events, and the second term corresponds to the log-likelihood of infinitely many non-events. Intuitively, the probability that there is no event of any type in the infinitesimally time interval $[t, t + dt)$ is equal to $1 - \lambda(t)dt$, the log of which is $-\lambda(t)dt$.

6 Experiments

To compare our method with the state-of-the-art, we conduct experiments on four real-world datasets. The datasets have been purposefully chosen in order to span over various properties, i.e., the number of event type ranges from 3 ot $5,000$; the average sequence length ranges from 3 to 109. Details about the datasets can be found in Table 1. Each dataset is split into a training set, a validation set and a testing set. The validation set is used to tune the hyper-parameters while the testing set is used to measure model performance. The evaluation measure used to compare the performance of the models is the log-likelihood (nats) as done in previous work (Mei and Eisner, 2017).

6.1 Real-World Datasets

Retweet Dataset. The Retweet dataset contains a total number of 166,076 retweet sequences. There are $U = 3$ types: "small", "medium" and "large" retweeters. The "small" retweeters are those who have fewer than 120 followers, "medium" retweeters have more than 120 but fewer than 1,363 followers, and the rest are "large" retweeters. As for retweet time, the first event in each sequence is labeled with 0, the next events are labeled with reference to their time interval with respect to the first event in this sequence.

StackOverflow. The StackOverflow dataset includes sequences of user awards in a two-year period. StackOverflow is a question-answering website where users are awarded based on their answers to questions proposed by others. There are in total $U = 22$ types of awards on StackOverflow. The award time records when a user receives the award.

MIMIC-II. The Multiparameter Intelligent Monitoring in Intensive Care (MIMIC-II) dataset is an electric medical record system containing 7 years of anonymous clinical visit records of patients in intensive care units. There are in total $U = 75$ types of events.

MemeTrack Dataset. The MemeTrack dataset has meme trajectory in articles from 1.5 million blogs and news sites from August 2008 to May 2009. As this dataset has trajectory of each meme across websites, the event type corresponds to the website that mentions it. The event time is when a website mentions it. The version of the dataset used in this paper is the one developed by Gomez Rodriguez et al. (2013), which selects 5,000 websites that most frequently mention memes ($U = 5,000$).

Table 2: Log-likelihood (nats) per event on the testing subset.

Dataset	HP	ONPHP	RMTPP	N-SM-MPP	SAHP
Retweet	-10.54	-7.65	-8.86	-7.94	**-7.22**
StackOverflow	-3.43	-2.54	-2.97	-2.47	**-2.22**
MIMIC-II	-3.84	-2.04	-2.59	-1.99	**-1.78**
MemeTrack	-13.81	-12.70	-12.77	-12.65	**-12.41**

6.2 Training Details

We implement the multi-head attention. This allows the model to jointly attend information from different representation subspaces (Vaswani et al., 2017). The number of heads is a hyper-parameter. We explore this hyper-parameter in the set $\{1, 2, 4, 8, 16\}$. Another hyper-parameter is the number of attention layers. We explore this hyper-parameter in the set $\{2, 3, 4, 5, 6\}$. To accelerate the self-attention convergence, we adapt the Adam as the basic optimizer and develop a warm-up stage for the learning rate during the training process. The initial learning rate is set to $1e{-}4$. To mitigate overfitting we apply dropout with rate set to 0.1. We train using mini-batches of size 32 over the training sets on an Nvidia GeForce GTX 1080 card. Early stopping is used when the validation loss does not decrease more than $1e{-}3$.

6.3 Baselines

We compare our method SAHP against the following state-of-the-art models, already mentioned in Section 3:

Hawkes Processes (HP). This is the most conventional Hawkes process statistical model which intensity is described in the Eq. 1. It uses an exponential kernel;

Online Non-parametric Hawkes Processes (ONPHP). This non-parametric method (Yang et al., 2017) approximates the intensity in a reproducing kernel Hilbert space in an online learning fashion;

Recurrent Marked Temporal Point Processes (RMTPP). This method (Du et al., 2016) uses RNN to learn a representation of influences from past events;

Neurally Self-Modulating Multivariate Point Processes (N-SM-MPP). This method (Mei and Eisner, 2017) uses a continuous-time RNN, which includes intensity decay and eliminate the need to encode event intervals as numerical inputs of the RNN.

7 Results and Discussion

The software used to run these experiments is attached as supplemental material and is publicly available at the following web-link: anonymouzed.

Performance on the testing dataset. To demonstrate effectiveness of our method, we compare our model against the baselines on the four testing sets. In Table 2 we show the performance per-event log-likelihood (nats) of these methods on each dataset. The lower the log-likelihood is, the more accurate a method predicts future events. Our method outperforms the baselines in all four datasets. As expected, the conventional HP method is the worst in predicting future events in all datasets. The N-SM-MPP method is better than its discrete counterpart RMTPP. The ONPHP method performs better than the rest of the baselines in small-scale event types, i.e., the Retweet dataset. However, as the number of event types increases, the N-SM-MPP method outperforms ONPHP.

Model hyper-parameters. The two hyper-parameters of our model, number of heads and number of attention layers, have complementary side-effects: increasing the number of heads increases the computational complexity of the model, while increasing the number of attention layers increases the memory needed to allocate the model. By examining the performance of the model on the testing set and varying these two hyper-parameters (as explained in Section 6.2), we find that the number of heads is more influential than the number of attention layers – increasing the number of heads makes the effect of the number of attention layers negligible.

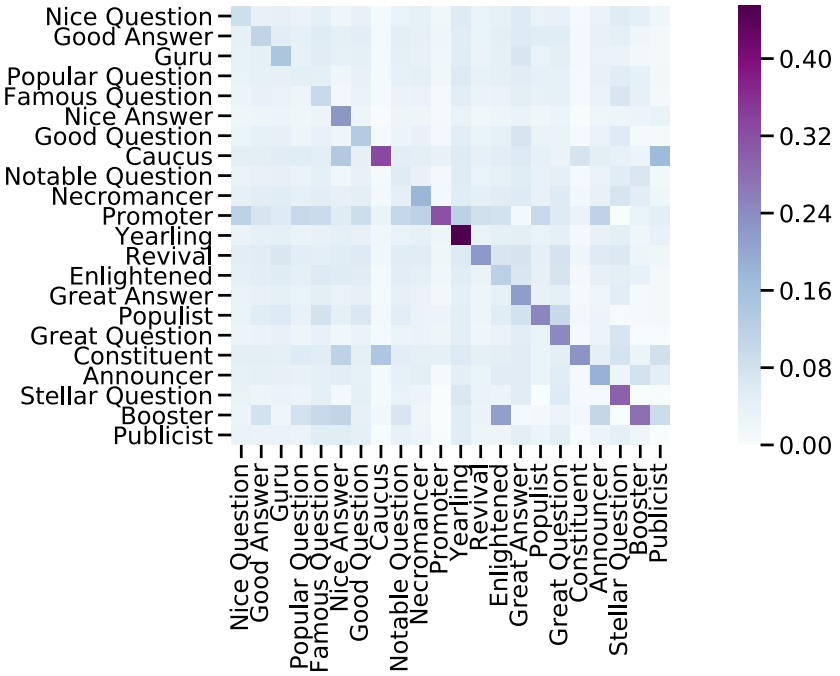

Figure 3: Visualization of statistical attention between event types on the StackOverflow dataset.

Model interpretability Apart from strong capacity in constructing the intensity function, another advantage of our method is the high interpretability. The proposed SAHP method is able to reveal peer influence among event types. To demonstrate that, we extract the attention weight that the type-u allocates to the type-v and accumulate such attention weight over all the sequences on the StackOverflow testing set. We remove the effect of the frequency of the (u, v) pairs in the dataset through dividing the accumulated attention weight via the (u, v) frequency. After normalization, we obtain the statistical attention distribution as shown in Fig. 7. The cell at the u-th row and v-th column means the statistical attention that the type-u allocates to the type-v. Two findings can be drawn form this figure: 1) for most cells in the diagonal line, when the model computes the intensity of one event, it attends to the history events of the same type; 2) dark cells in the non-diagonal line, such as the (*Constituent, Caucus*), the (*Boosters and Enlightened*) and the (*Caucus and Publicist*), the model attend to the latter when computing the likelihood of the former.

Model complexity analysis. Suppose N is the sequence length and d is the dimension of hidden units. We compare model complexity of the SAHP with recurrent neural Hawkes processes from two desiderata (Vaswani et al., 2017): maximum path length and sequential operations. As for the first desideratum, the SAHP has computational complexity of $O(1)$ while the recurrent counterparts have $O(n)$. This means that the SAHP is able to learn historical dependencies in event sequences regardless of the length of forward and backward signals. And this also explains why our model outperforms the recurrent counterparts in predicting the future. Also, the SAHP has $O(1)$ while recurrent structures require $O(n)$ sequential operations, thus the SAHP can be highly parallelized and is much faster.

8 Conclusion

The intensity function is the key of Hawkes processes in predicting asynchronous event sequences in the continuous time domain. In this paper, we propose a self-attentive Hawkes process where self-attention is adapted to enhance expressivity of the intensity function. Enhancement are in two aspects: model prediction and model interpretability. For the former, the proposed method outperforms both statistical Hawkes processes and recurrent neural Hawkes processes via better capturing event dependencies; while for the latter, the model is able to reveal peer influence via the learnt attention weights. Extensive experiments demonstrate the superiority of the proposed method.

References

Massil Achab, Emmanuel Bacry, Stéphane Gaïffas, Iacopo Mastromatteo, and Jean-François Muzy. Uncovering causality from multivariate hawkes integrated cumulants. *J. Mach. Learn. Res.*, 18(1), January 2017. ISSN 1532-4435.

Emmanuel Bacry and Jean-François Muzy. Hawkes model for price and trades high-frequency dynamics. *Quantitative Finance*, 14(7):1147–1166, 2014.

Emmanuel Bacry and Jean-François Muzy. First-and second-order statistics characterization of hawkes processes and non-parametric estimation. *IEEE Transactions on Information Theory*, 62(4):2184–2202, 2016.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014. URL `http://arxiv.org/abs/1409.0473`.

David R Brillinger, Peter M Guttorp, Frederic Paik Schoenberg, Abdel H El-Shaarawi, and Walter W Piegorsch. Point processes, temporal. *Encyclopedia of Environmetrics*, 3:1577–1581, 2002.

Edward Choi, Nan Du, Robert Chen, Le Song, and Jimeng Sun. Constructing disease network and temporal progression model via context-sensitive hawkes process. In *2015 IEEE International Conference on Data Mining*, pages 721–726. IEEE, 2015.

David Roxbee Cox and Valerie Isham. *Point processes*, volume 12. CRC Press, 1980.

Daryl J Daley and David Vere-Jones. *An introduction to the theory of point processes: volume II: general theory and structure*. Springer Science & Business Media, 2007.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Nan Du, Mehrdad Farajtabar, Amr Ahmed, Alexander J Smola, and Le Song. Dirichlet-hawkes processes with applications to clustering continuous-time document streams. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 219–228. ACM, 2015a.

Nan Du, Yichen Wang, Niao He, Jimeng Sun, and Le Song. Time-sensitive recommendation from recurrent user activities. In *Advances in Neural Information Processing Systems*, pages 3492–3500, 2015b.

Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. 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*, pages 1555–1564. ACM, 2016.

Jalal Etesami, Negar Kiyavash, Kun Zhang, and Kushagra Singhal. Learning network of multivariate hawkes processes: A time series approach. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, UAI'16. AUAI Press, 2016. ISBN 978-0-9966431-1-5.

Mehrdad Farajtabar, Nan Du, Manuel Gomez Rodriguez, Isabel Valera, Hongyuan Zha, and Le Song. Shaping social activity by incentivizing users. In *Advances in neural information processing systems*, pages 2474–2482, 2014.

Mehrdad Farajtabar, Yichen Wang, Manuel Gomez Rodriguez, Shuang Li, Hongyuan Zha, and Le Song. Coevolve: A joint point process model for information diffusion and network co-evolution. In *Advances in Neural Information Processing Systems*, pages 1954–1962, 2015.

Manuel Gomez Rodriguez, Jure Leskovec, and Bernhard Schölkopf. Structure and dynamics of information pathways in online media. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*. ACM, 2013.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

Fangjian Guo, Charles Blundell, Hanna Wallach, and Katherine Heller. The bayesian echo chamber: Modeling social influence via linguistic accommodation. In *Artificial Intelligence and Statistics*, pages 315–323, 2015.

Niels Richard Hansen, Patricia Reynaud-Bouret, Vincent Rivoirard, et al. Lasso and probabilistic inequalities for multivariate point processes. *Bernoulli*, 21(1):83–143, 2015.

John Hawkes. On the hausdorff dimension of the intersection of the range of a stable process with a borel set. *Probability Theory and Related Fields*, 19(2):90–102, 1971.

Xinran He, Theodoros Rekatsinas, James Foulds, Lise Getoor, and Yan Liu. Hawkestopic: A joint model for network inference and topic modeling from text-based cascades. In *International conference on machine learning*, pages 871–880, 2015.

Andrej Karpathy, Justin Johnson, and Li Fei-Fei. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*, 2015.

Viktoriya Krakovna and Finale Doshi-Velez. Increasing the interpretability of recurrent neural networks using hidden markov models. *arXiv preprint arXiv:1611.05934*, 2016.

Young Lee, Kar Wai Lim, and Cheng Soon Ong. Hawkes processes with stochastic excitations. In *International Conference on Machine Learning*, pages 79–88, 2016.

Erik Lewis and George Mohler. A nonparametric em algorithm for multiscale hawkes processes. *Journal of Nonparametric Statistics*, 1(1):1–20, 2011.

Shuang Li, Shuai Xiao, Shixiang Zhu, Nan Du, Yao Xie, and Le Song. Learning temporal point processes via reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 10781–10791, 2018.

Michal Lukasik, PK Srijith, Duy Vu, Kalina Bontcheva, Arkaitz Zubiaga, and Trevor Cohn. Hawkes processes for continuous time sequence classification: an application to rumour stance classification in twitter. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 393–398, 2016.

Hongyuan Mei and Jason M Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems*, pages 6754–6764, 2017.

Yosihiko Ogata. Space-time point-process models for earthquake occurrences. *Annals of the Institute of Statistical Mathematics*, 50(2):379–402, 1998.

Patricia Reynaud-Bouret, Sophie Schbath, et al. Adaptive estimation for hawkes processes; application to genome analysis. *The Annals of Statistics*, 38(5):2781–2822, 2010.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6000–6010, 2017. URL `http://papers.nips.cc/paper/7181-attention-is-all-you-need`.

Dirk Walther, Ueli Rutishauser, Christof Koch, and Pietro Perona. On the usefulness of attention for object recognition. In *Workshop on Attention and Performance in Computational Vision at ECCV*, pages 96–103, 2004.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

Yichen Wang, Bo Xie, Nan Du, and Le Song. Isotonic hawkes processes. In *International conference on machine learning*, pages 2226–2234, 2016.

Shuai Xiao, Mehrdad Farajtabar, Xiaojing Ye, Junchi Yan, Le Song, and Hongyuan Zha. Wasserstein learning of deep generative point process models. In *Advances in Neural Information Processing Systems*, pages 3247–3257, 2017a.

Shuai Xiao, Junchi Yan, Xiaokang Yang, Hongyuan Zha, and Stephen M Chu. Modeling the intensity function of point process via recurrent neural networks. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017b.

Hongteng Xu, Mehrdad Farajtabar, and Hongyuan Zha. Learning granger causality for hawkes processes. In *International Conference on Machine Learning*, pages 1717–1726, 2016.

Shuang-Hong Yang and Hongyuan Zha. Mixture of mutually exciting processes for viral diffusion. In *International Conference on Machine Learning*, pages 1–9, 2013.

Yingxiang Yang, Jalal Etesami, Niao He, and Negar Kiyavash. Online learning for multivariate hawkes processes. In *Advances in Neural Information Processing Systems*, pages 4937–4946, 2017.

Qiang Zhang, Rui Luo, Yaodong Yang, and Yuanyuan Liu. Benchmarking deep sequential models on volatility predictions for financial time series. *arXiv preprint arXiv:1811.03711*, 2018.

Ke Zhou, Hongyuan Zha, and Le Song. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *Artificial Intelligence and Statistics*, pages 641–649, 2013.