Modeling Event Propagation via Graph Biased Temporal Point Process

Weichang Wu, Huanxi Liu, Xiaohu Zhang, Yu Liu, and Hongyuan Zha

Abstract—Temporal point process is widely used for sequential data modeling. In this article, we focus on the problem of modeling sequential event propagation in graph, such as retweeting by social network users and news transmitting between websites. Given a collection of event propagation sequences, the conventional point process model considers only the event history, i.e., embed event history into a vector, not the latent graph structure. We propose a graph biased temporal point process (GBTPP) leveraging the structural information from graph representation learning, where the direct influence between nodes and indirect influence from event history is modeled. Moreover, the learned node embedding vector is also integrated into the embedded event history as side information. Experiments on a synthetic data set and two real-world data sets show the efficacy of our model compared with conventional methods and state-of-the-art ones.

Index Terms—Graph representation, recurrent neural network (RNN), stochastic processes, temporal point processes (TPPs).

I. INTRODUCTION

E vent sequences modeling is widely used across different areas and applications. In e-commerce, the online purchase records over time can be modeled as event sequences. In health informatics, a series of treatments taken by patients can be tracked as event sequences. In seismology, a sequence of earthquakes recorded are modeled as event sequences. In social media such as Twitter, every time a user posts, transmits, or likes a tweet, it corresponds to a new event adding to the user behavior sequence. In all the abovementioned settings, event sequences modeling is of vital importance for predicting future events and recognizing hidden patterns given history sequences.

For modeling event sequences, temporal point process (TPP) [4] is a useful tool. For example, Zhou et al. [44] used the so-called multidimensional Hawkes processes (HPs) to model the sequential user actions in a social network, and the learned infectivity matrix is useful for uncovering the mutual influences between users. Mixtures of HPs [19] are modeled for inferring missing event attributes from the behavioral observation by considering the dependence among dyadic events. In [38], a water pipe failure prediction system is designed for effective replacement and rehabilitation. The water pipe failure sequence is formulated as a self-exciting stochastic process.

Marked TPP is (MTPP) an important domain in TPP for event sequences modeling. In MTPP, an event can carry extra information called marker. The marker typically refers to event type and lies in the discrete label space, i.e., a finite category set \( \{1, \ldots, m \} \). In e-commerce, the marker can refer to the users and items. In health informatics, markers can be the treatments and medications of a patient. In predictive maintenance, markers can carry important log data for when the failure occurs and what is the type. In all these examples, effectively modeling and predicting the dynamic behavior while leveraging the information contained in the markers is of vital importance for MTPP.

In this article, we focus on a special case of MTPP, where the event sequence is an event propagation process in a directed weighted graph and the marker denotes the node in the graph, for example, a retweeting sequence in the social network where the markers denote users in user network and a news transmission sequence between websites where the markers denote the website in the influence network.

Modeling and predicting event propagation is a challenging task. The difficulty lies in how to leverage the network structure and node proximity in the graph when modeling the event propagation sequence. Conventional TPP methods model the event propagation path as general event sequences, computing probabilities, and making predictions based on history events, such as in [6], [34], [36], and [38]. However, modeling event propagation without considering the connections of nodes in graph is inaccurate. As shown in Fig. 1, in a conventional TPP model, node \( V_4 \) is not connected to node \( V_5 \), while the indirect history influence is measured equally as the direct influence between the connected nodes \( V_4 \) and \( V_5 \).

Intuitively, only the connected nodes influence each other. However, when we predict the propagation of events in graph, merely considering the direct influence between connected nodes is also not appropriate without using the TPP method. As shown in Fig. 1, for the case that only events from \( V_2 \) propagate to \( V_5 \) through \( V_4 \), whereas events from \( V_2 \) cannot
The situation that only events from $V_2$ propagate to $V_5$, which is also inaccurate. Given observed two propagation sequences $\{V_1 \rightarrow V_2 \rightarrow V_4 \rightarrow V_5\}$ and $\{V_6 \rightarrow V_3 \rightarrow V_4\}$. Considering only the node connections cannot cope with the situation that only events from $V_2$ propagate to $V_5$ through $V_4$ but events from $V_1$ cannot propagate to $V_5$. While conventional TPP can deal with this case, it measures the indirect influence from node $V_1$ and $V_2$ to $V_5$ and $V_6$ to $V_4$ equally as indirect influence between connected nodes $V_2$, $V_3$, $V_5$, and $V_4$ which is also inaccurate.

Propagate to $V_5$, measuring that only the direct influence cannot handle this situation.

In this article, to model the event propagation process in graph considering both direct influence between connected nodes and indirect influence of propagation history, we propose a graph biased TPP (GBTPP) leveraging the structural information extracted from graph representation. Compared with a conventional TPP model, we make two major contributions as follows.

1) The direct influence between connected nodes is measured separately from the indirect history influence as a bias term, leveraging the first-order proximity between the nodes learned by node embedding. The intensity of the direct influence is controlled by a scale factor related to the event history.

2) The node embedding vector is added to the event propagation history embedding when modeling the indirect influence so that the structural information can be integrated into the model.

To verify the efficacy of our model, we experiment on a synthetic data set and two real-world data sets, including the Higgs Twitter data set [5] predicting tweets propagation in the social network Twitter, and a MemeTracker data set [17] predicting meme propagation between websites. Empirical results show that the proposed GBTPP model outperforms conventional methods and state-of-the-art ones.

II. RELATED WORK

Sequential event data are generated from lots of social activities, e.g., financial transactions, electronic health records, and e-commerce purchase records. In these scenarios, the sequential event data contain abundant information about which type of event happens at what time. For example, the daily routine of a person contains various places at different moments during one day. Stock managers buy or sell stocks at different instants of time. Patients with chronic diseases pay regular visits to the hospital to obtain their diagnoses each time.

Moreover, there are many underlying structures in sequential data besides its intrinsic temporal structure, especially for network and graph structure. Actually, when modeling sequential data, we also need to incorporate the accompanying graph structural information in many cases.

A. Sequential Data Modeling

As sequential event data are frequently produced from various domains and applications, modeling the event sequences, especially predicting future events, is of vital importance; based on the observed event sequence history, predicting which type of event will happen at what time in the future. This kind of prediction task is of great use in many applications, e.g., in the stock market, predicting when to buy or sell a particular stock has important business value. For mainstream assistant, making spatial and temporal predictions on when and where a person will visit a certain place will make personal service more suitable and relevant. For health-care services, predicting future clinical events and disease progression can help to provide personal medical services and reduce potential risks.

To model event sequences, existing literature works attempt to solve this problem in mainly two categories of methods.

First, the conventional varying-order Markov models [1] deal with this problem as a discrete-time sequence prediction task. Based on the observed history states sequence, prediction of the event type is given by the most likely state that the state transition process will evolve into on the next step. An obvious limit for the families of Markov models is that they assume that the state transition process proceeds with unit time step, and it cannot capture the temporal dependence of the continuous time and give predictions on the exact time of the next event. Moreover, Markov models cannot deal with long dependence of the history events when the event sequence is long, because the size of the state space grows exponentially with the number of the time steps considered in the Markov model. It is worth mentioning that semi-Markov models [14] can model continuous time intervals between two states to some extent, by assuming the intervals to follow some simple distributions, but it still has the state-space explosion problem when dealing with long time dependence.

Second, TPP with conditional intensity functions is a more general framework for sequential event data modeling. TPP is powerful for modeling event sequence with timestamp in continuous time space. Early work dates back to the HPs [13], which shows appropriateness for self-exciting and mutual-exciting process such as earthquake and its aftershock [23], [24]. As an effective model for event sequence modeling, TPP has widely been used in various applications, including data mining tasks, e.g., social infectivity learning [18], conflict analysis [41], crime modeling [29], email network analytics [11], extremal behavior of stock price [7], and event prediction tasks, e.g., failure prediction [8], sales outcome forecasting [40], and literature citation prediction [31].
Traditional TPP models are modeled by parametric forms involving manual design of conditional intensity function \( \lambda(t) \) depicting event occurrence rate over time, which measures the instantaneous event occurrence rate at time \( t \). A few popular examples include the following.

1) **Poisson Process** [16]: The basic form is history independent \( \lambda(t) = \lambda_0 \) which can be dated back to the 1900s.

2) **Reinforced Poisson Processes** [25]: The model captures the ‘rich-get-richer’ mechanism by \( \lambda(t) = \lambda_0 f(i(t)) \) where \( f(i) \) mimics the aging effect while \( i(t) \) is the accumulation of history events;

3) **Self-Exciting Process (HP)** [12]: It provides an additive model to capture the self-exciting effect from history events \( \lambda(t) = \lambda_0 + \sum_{t < s} g_{exc}(t - t_i) \).

4) **Reactive Point Process** [8]: Generalization to the HP by adding a self-inhibiting term to account for the inhibiting effects from history \( \lambda(t) = \lambda_0 + \sum_{t < s} g_{exc}(t - t_i) - \sum_{t < s} g_{inh}(t - t_i) \).

One obvious limitation of the abovementioned TPP models is that they all assume that all the samples obey a single parametric form that is too idealistic for real-world data. By contrast, recurrent neural network (RNN)-based models [6, 22, 36] are devised for learning point process. In these parametric form that is too idealistic for real-world data. In these parametric form that is too idealistic for real-world data.

**B. Sequential Data Modeling With Graph Structure**

With the increase of data dimensionality and the data types become more diverse, sequential data often contain other structures, typically graphs for example. The most common type of graph data associated with sequential data is the spatial graph structure, which is in the form of spatiotemporal data. The spatiotemporal point process is often used to describe the spatial and temporal structure of spatiotemporal data. For example, in [9], the spatiotemporal point process is used to establish the spatiotemporal grain of continuous landscape tessellations and graphs, defining the grain of landscape tessellations and graphs with a variety of geometries. One of the cases is that the spatiotemporal data describing the spatial activity of brushtail possums are collected using very high-frequency (VHF) and global positioning system (GPS)-based radiotelemetry, given their associated temporal scale. Then, the intensity of spatiotemporal point process is computed based on the spatiotemporal data, and the grain of regular or irregular landscape tessellations or graphs in terms of point process intensity provides a general way of reporting grain.

Most recently, the spatiotemporal point process has also been used to capture crime linkage in [45]. Actually, a spatiotemporal–textual point process is proposed to utilize the text, time, and location information in the crime records. Specifically, in the conditional intensity function, the temporal effect is measured by the temporal kernel function, e.g., exponential function; the spatial correlation is measured by a coefficient matrix \( A = \{a_{ij}\} \), where each entry \( a_{ij} \) represents the strength of the influence from location indexed \( j \) by to the location indexed by \( i \); the text information is represented by a mapping function to project the bag-of-words representations into an \( m \)-bit binary Modus Operandi embedding space, where the embedding similarity is measured by inner product.

In addition to spatiotemporal point processes dealing with spatial graph information for sequential data, from the perspective of graph representation, exploring the influence of sequential data on graphs is another important issue, e.g., learning and representing dynamic graphs. In most recent work in [28], an end-to-end architecture leveraging RNNs is proposed to create a temporal node embedding over dynamic graphs. Specifically, by introducing the RNNs to reduce the sequential data into a temporal historical embedding, the model can calculate the final node embedding by combining the temporal dynamics with the graph structures. Compared with conventional works on dynamic graphs designed for specific prediction tasks, the node embedding method over dynamic graphs in [28] is a more generalized graph embedding technique that jointly optimizes the node representations and prediction task objectives, such as node classification and link prediction.

As discussed earlier, from the view of the application, we consider existing literature works [9], [28], [45] to be related to our work, as they all leverage information from graph structures, considering both temporal dynamic information and structural information, while from the view of methodology, there are differences and connections between our work and these existing ones. Especially, they encode the structural information into the TPPs in different ways.

Specifically, the spatiotemporal point process model implemented in [9] and [45] encode the spatial graph information by adding an extra marker dimension in conventional parametric MTPPs. Conventional MTPP uses conditional intensity function \( \lambda(t) = \mu + \sum_{i < s} g(t, t_i, x, x_i) \), \( \mu \) is the constant base intensity, \( \sum_{i < s} g(t, t_i, x, x_i) \) is the cumulative mutual intensity of event history \( H_i = \{s_i, t_i\} \), and \( g \) is the nonnegative triggering function [26]. In [45] where the spatiotemporal point process is used for crime linkage detection, an extra marker \( s \) is introduced representing crime location in the spatial graph so that the spatial graph information can be encoded into the intensity function by \( \lambda(x, t) = \mu(s) + \sum_{i < s} g(s, x, t, x_i) \). In our work, instead of building a conventional parametric model as in [45], we adopt RNN-based models and incorporate the structural information via graph embedding technique.

While recent work in [28] also employs the combination of RNN and graph embedding, it is in essence different from our work. In short, Singer et al. [28] focused on graph embedding learning over the temporal dynamic graph and modified the existing graph embedding technique by introducing RNN, whereas we focus on the information propagation process between nodes in a static graph and revise the existing RNN-based MTPP model by introducing graph embedding. In specific, there are two major differences methodologically compared with our work. First, for the use of RNN, Singer et al. [28] simply used the RNN to reduce each node’s historical embeddings \( X^{(0)} \) to a \( d \)-dimensional node
embedding vector, where \( X^{(e)} \in \mathbb{R}^{T \times d} \) is the historical \( T \) embeddings of node \( v \) over time and each of size \( d \). Instead of simply extracting temporal dynamic information from sequential data, we employ RNN as a component of the MTPP model, via encoding the information propagation history into a hidden vector \( h_v \), and further use \( h_v \) for learning the parameters of MTPP’s intensity function. Second, for the use of graph embedding, Singer et al. [28] used static graph embedding as initialization of the algorithm and the final node embedding vector is obtained by reducing the historical \( T \) embeddings of each node into a final one via RNN so that the evolution of a temporal graph over time can be captured, while in our work, the vanilla graph embedding is used to acquire each node’s embedding vector, as we focus on the propagation process in a static graph.

Though sequential data with spatial graph information is explored in spatiotemporal point processes [9], [45], when dealing with the more general graph structure, e.g., information propagation in social networks, conventional spatiotemporal point processes are not able to incorporate the graph structural information. While in [28], a graph embedding technique is developed over temporal dynamic graphs for graph representation rather than sequential data modeling, TPPs modeling on sequential data with graph structure is rarely studied.

In this article, we study this problem in the form of event propagation in social networks. Event or information propagation and diffusion in social networks has been studied under different methodologies, including TPP. For example, in [15], HP is used to model the evolutionary trend of the influence of events on search queries by proposing computational measures that quantify the influence of an event on a query to identify triggered queries. When dealing with event propagation sequences, a major limitation of existing studies is that the structural information of the latent graph \( G = (V, E) \) is not utilized. Conventional TPP models, including state-of-the-art method in [6], solve event propagation modeling as general event sequences modeling and take input \( \{v_i, t_i\} \), whereas our GBTPP model leverages the structural information and node proximity of graph \( G \) taking input \( \{v_i, t_i, y_i\} \), where \( y_i \) is the node embedding vector obtained by a graph representation learning method for \( G \).

III. PROPOSED MODEL

Given a collection of observed event propagation sequences \( \mathcal{C} = \{S^1, S^2, \ldots\} \) in a latent graph \( G = (V, E) \) with node set \( V \) and edges set \( E \), each \( S^k = ((t^k_1, v^k_1), (t^k_2, v^k_2), \ldots) \) is a sequence of pairs \( (t^k, v^k) \), where \( t^k \) is the time when event \( k \) propagates from node \( v_i \) to node \( v_{j+1} \).

To model the event propagation processes and make predictions on propagation node \( v_{t+1} \) and propagation time \( t \), we leverage the structural information in graph \( G \), our method contains two steps.

1) **Graph Representation:** Learn a representation for latent graph \( G = (V, E) \) from observed propagation sequences \( \mathcal{C} \). For each node \( v_i \in V \), we learn the node embedding vector \( y_i \) that preserves the first-order proximity.

2) **Graph Biased Temporal Point Process:** Train a GBTPP model based on the learned graph representation \( \{y^k_i\}_{k=1}^V \) and observed propagation sequences \( \mathcal{C} \). The GBTPP model integrates the node proximity as a bias term and uses a scale factor to control the intensity of this term.

We present the details of the graph representation and GBTPP as follows.

A. Graph Representation

Graph embedding and representation has been widely used in both academia and industry in recent years. Lots of literature works are proposed to convert a graph \( G = (V, E) \) into a \( d \)-dimensional space, in which the graph property is preserved [3]. In graph representation techniques, the graph is represented as either a \( d \)-dimensional vector (for a whole graph) or a set of \( d \)-dimensional vectors with each vector representing the embedding of part of the graph (e.g., node, edge, and substructure).

In general, the graph property is quantified by proximity measured by the first-order proximity and second-order proximity as follows.

1) **First-Order Proximity:** The local pairwise similarity between nodes connected by edges. It compares the direct connection strength between a node pair. The first-order proximity between nodes \( v_i \) and \( v_j \) is the weight of the edge \( e_{ij} \), i.e., \( A_{i,j} \). Two nodes are more similar if they are connected by an edge with larger weight.

2) **Second-Order Proximity:** The similarity of the neighbors’ neighborhood structures. The more similar two nodes’ neighborhoods are, the larger the second-order proximity value between them. Formally, the second-order proximity \( \text{sim}^{(2)} \) between node \( v_i \) and \( v_j \) is a similarity between \( v_i \)'s neighborhood \( \text{sim}^{(1)}(v_i) \) and \( v_j \)'s neighborhood \( \text{sim}^{(1)}(v_j) \).

In general, the learned graph representation preserves either first-order proximity such as [21] and [33] or second-order proximity such as [10], [20], and [42]. In some recent work [2], [30], [43], both the first- and second-order proximities are empirically calculated based on the joint probability and conditional probability of two nodes.

In this article, the first-order proximity is preserved. We learn the graph representation as a set of embedding vectors for the nodes in latent graph \( G = (V, E) \) keeping the first-order proximity, where each node \( v \) in graph \( G \) is represented by two \( d \)-dimensional vector \( y^v \) and \( y^v' \). For the directed weighted graph \( G \), the direct influence from node \( v_i \) to node \( v_j \), i.e., the weight of edge \( e_{ij} \), is computed as \( p(y^v', y^v_j) \), and the direct influence from \( v_j \) to \( v_i \) is computed as \( p(y^v_j, y^v_i) \).

Specifically, to obtain the node embedding for each node in \( G = (V, E) \), we have the following implementation.

1) **Edge Reconstruction Probability:** The learned node embedding should be able to reestablish edges in the original input graph. This can be realized by maximizing the probability of generating all observed edges using node embedding. The directed edge between a node pair \( v_i \) and \( v_j \) indicating their first-order proximity can be calculated as the joint
probability using the embedding $y^i_i$ of $v_i$ and $y^j_j$ of $v_j$

$$p^{(1)}(v_i, v_j) = \frac{1}{1 + \exp(-y^i_i y^j_j^T)}.$$  \hspace{1cm} (1)

2) Minimizing Distance-Based Loss: From event propagation sequences $\mathcal{C}$ in weighted directed graph $G$, we have the empirical estimation of the adjacent matrix $A$, in which $A_{i,j}$ is the empirical estimation for the weight of edge $e_{ij}$ computed by the normalized propagation number from $v_i$ to $v_j$ as $A_{i,j} = N_{ij}/N_{max}$, where $N_{ij}$ is the number of observed event propagation from $v_i$ to $v_j$ and $N_{max}$ is the global maximum number of event propagation between any given node.

To capture the structural information and connections between the nodes in graph $G$, the node proximity calculated based on node embedding in (1) should be as close to the node proximity calculated based on the observed edges as possible. Specifically, node proximity can be calculated based on node embedding or empirically calculated based on observed edges. Minimizing the differences between the two types of proximities can preserve the corresponding proximity.

For the first-order proximity, it can be computed as $p^{(1)}$ using node embedding defined in (1), while the empirical probability is $\hat{p}^{(1)}(v_i, v_j) = A_{i,j}/\sum_{e_{ij}\in E} A_{i,j}$, where $A_{i,j}$ is the empirical estimation for the weight of edge $e_{ij}$. The smaller the distance between $p^{(1)}$ and $\hat{p}^{(1)}$ is, the better first-order proximity is preserved.

By adopting KL-divergence as the distance function, we can minimize the difference between $p^{(1)}$ and $\hat{p}^{(1)}$, and the objective function preserving the first-order proximity in

$$C^{(1)}_{min} = \min - \sum_{e_{ij}\in E} A_{i,j} \log p^{(1)}(v_i, v_j).$$  \hspace{1cm} (2)

For each node $v_i$, we can learn the corresponding node embedding vector $\mathbf{y}_i = \{y^i_i, y^j_j\}$ by (2), indicating the first-order proximity by (1).

B. Graph Biased Temporal Point Process

Given history propagation sequence $\mathcal{H}_{t_{n-1}} = \{(t_1, v_1), (t_2, v_2), \ldots, (t_{n-1}, v_{n-1})\}$ and current node $v_n$, the GBTPP model aims to compute the probability $P(v_{n+1} | \mathcal{H}_{t_{n-1}}, v_n)$ of the event propagating to node $v_{n+1}$ given propagation history $\mathcal{H}_{t_{n-1}}$ and current node $v_n$, and the estimation of the propagation time $t_n$ by the likelihood $f(t_n | \mathcal{H}_{t_{n-1}}, v_n)$.

As shown in Figs. 2 and 3, conventional TPP model embeds current node $v_n$ into history embedding vector $h_n$, whereas the GBTPP model measures the direct influence of current node by $y_n$ and indirect history influence by $h_{n-1}$. The architecture of GBTPP is presented in Fig. 4. We also illustrate the GBTPP model in Fig. 5. Specifically, we specify the proposed model as the following parts: input embedding, history embedding, graph bias term computation, and prediction.

1) Input Embedding: As shown in Fig. 4, the history input includes $\{v_{n-1}, t_{n-1}, y_{n-1}\}$ as a triple, including a sparse one-hot vector $v_{n-1}$ representing a node in graph $\mathcal{G} = (V, E)$, a continuous value $t_{n-1} \in (0, T)$ indicating the time of the event propagate from node $v_{n-1}$ to $v_n$, and the corresponding node embedding vector $y_{n-1} \in \mathbb{R}^d$.

The sparse one-hot vector representation of the node $v_i$ is projected into a latent space by an embedding layer with the weight matrix $W_{em}$ to achieve a more compact and efficient representation as $v_i = W_{em}^T v_i + b_{em}$, where $v_i$ is the embedding for $v_i$. Then, the representation vector $v_i$ is embedded into a common feature space $\mathbb{R}^d$ for both input embedding and next step history embedding, with a weight matrix $W^v$.

For the propagation time input $t_{n-1}$, we can extract the associated temporal features, e.g., the intervene duration $d_{n-1} = t_{n-1} - t_{n-2}$. Here, we slightly abuse the notation for
temporal feature still as $t_{n-1}$ for conciseness. The temporal feature $t_{n-1}$ is also embedded into common feature space $\mathbb{R}^H$ by weight matrix $W^t$.

Similarly, the node embedding $y_j$ is also projected from the node embedding space to the feature space by an embedding layer with weight matrix $W^v$. The history input triple $\{v_{n-1}, t_{n-1}, y_{n-1}\}$ is embedded into a common history feature space as $\{W^v v_{n-1}, W^v y_{n-1}, W^v t_{n-1}\}$.

**2) History Embedding:** In the history embedding part, the embedded input is added to the propagation history embedding $h_{n-1}$ with the last propagation trajectory embedding vector $h_{n-2}$ by an RNN so that we have an event propagation history embedding $h_{n-1}$ as

$$h_{n-1} = \max\{W^v v_{n-1} + W^v y_{n-1} + W^v t_{n-1}, W^h h_{n-2} + b_h, 0\}. \quad (3)$$

Compared with conventional TPPs, except for event marker, i.e., propagation node $v_{n-1}$ and time $t_{n-1}$, the node embedding $y_{n-1}$ indicating the structural information of $v_{n-1}$ in graph $G = (V, E)$ is used as side information input when computing history embedding.

**3) Graph Bias:** Given the embedded event propagation history, conventional TPPs compute the event propagation probability as $P(v_{n+1} | h_n)$ and the likelihood of time $t_n$ as $f(t_n | h_n)$. For example, in [6], the propagation probability is computed as

$$P(v_{n+1} = k | h_n) = \frac{\exp(V^h_k, h_n + b^h_k)}{\sum_{k=1}^V \exp(V^h_k, h_n + b^h_k)} \quad (4)$$

where $V$ is the number of nodes, $V^h_k$ is the $k$th row of parameter matrix $V$, and $b^h_k$ is the constant bias term. The conditional intensity function $\lambda(t)$ is also computed conditional on $h_n$ by

$$\lambda(t) = \exp(v^T \cdot h_n + w^i (t - t_{n-1}) + b^i) \quad (5)$$

where $v^i$ is a column vector and $w^i$, $b^i$ is scalar, and the likelihood of event propagation time $t_n$ is computed as $f(t_n) = \lambda(t) \exp(- \int_{t_{n-1}}^t \lambda(r) dr)$.

Compared with conventional methods that embed current node $v_n$ into propagation history $h_n$, we model the direct influence of current node and the indirect influence of the propagation history. Besides using a constant bias term as $b^h_k$ as in (4), a graph bias term $b(h_n, y_n, y_k)$ is introduced for event propagation probability as

$$b(h_{n-1}, y_n, y_k) = \text{ReLU}(U^h_h, h_{n-1}) p(y_n, y_k) \quad (6)$$

where $p(y_n, y_k)$ is the first-order proximity learned in the graph representation step measuring the direct influence of node $v_n$ to node $v_k$ as in (1), ReLU function $\text{ReLU}(U^h_h, h_{n-1})$ computes the scale factor that measures the intensity of this influence, and $U^h_h$ is the $n$th row of parameter matrix $U$.

Given the graph bias term $b(h_{n-1}, y_n, y_k)$, the node propagation probability of GBTPP model is given by

$$P(v_{n+1} = k | h_n, v_{n-1}) = \frac{\exp(V^h_k, h_{n-1} + b(h_{n-1}, y_n, y_k) + b^h_k)}{\sum_{k=1}^V \exp(V^h_k, h_{n-1} + b(h_{n-1}, y_n, y_k) + b^h_k)} \quad (7)$$

where the direct influence of current node $v_n$ is measured by the bias term $b(h_{n-1}, y_n, y_k)$ in (6), and the indirect influence of propagation history is computed using the history embedding vector $h_{n-1}$.

For conventional intensity function, the direct influence of current node $v_n$ is also measured by a separate bias term using node embedding $y_n$ as

$$\lambda^v(t) = \exp(v^v^T \cdot h_n) + v^v^T \cdot y_n + w^v (t - t_j) + b^v \quad (8)$$

where $v^v$ and $v^v$ are column vectors and $w^v$ and $b^v$ are scalars. We list the specific meaning of the terms computed in (8), and the same term is also used in conventional TPP in (5) except for the direct influence term. Specifically, the following holds.

1) The history influence term $v^v^T \cdot h_n$ represents the accumulative influence from the history nodes and the timing information of the past propagation.
2) The direct influence term $v^v^T \cdot y_n$ represents the influence current node $v_n$.
3) The exponential assumption term $w^v (t - t_j)$ assumes that the intensity is an exponential function of $t$, where the exponential function acts as a nonlinear transformation and guarantees that the intensity is positive.
4) The last base intensity term gives a base intensity level for the propagation process.

Based on the conditional intensity function $\lambda^v(t)$, we can derive the likelihood that the event propagates from $v_n$ to $v_{n+1}$
at the time \( t \) given the history \( h_{n-1} \) by the following equation:

\[
f^*(t) = \lambda^*(t) \exp \left( - \int_{t_{n-1}}^{t} \lambda^*(\tau) d\tau \right)
\]

\[
= \exp \left\{ v^T \cdot h_{n-1} + v^T \cdot y_n + w^T (t - t_{n-1}) + b' \right\} \\
- \frac{1}{t_0} \exp \left\{ v^T \cdot h_{n-1} + v^T \cdot y_n + w^T (t - t_{n-1}) + b' \right\} \\
\frac{1}{t_0} \exp \left\{ v^T \cdot h_{n-1} + b' \right\}.
\]

(9)

4) Prediction: For propagation node prediction, given the propagation probability in (7), the next propagation node \( b_{n+1} \) is given by

\[
b_{n+1} = \arg \max_{b_{n+1} \in V} P(u_{n+1}|h_{n-1}, y_n)
\]

where \( V \) is the node set for graph \( G = (V, E) \).

For propagation time prediction, given the time likelihood in (11), the predicted propagation time \( t_n \) from node \( v_n \) to the next node is given by

\[
t_n = \int_{t_{n-1}}^{\infty} t \cdot f^*(t) dt.
\]

(11)

Commonly, the integration in (11) does not have analytic solutions. A numerical integration technique [27] for 1-D function is used to compute (11).

\section{Learning Algorithm}

Given a collection of event propagation sequences \( C \equiv \{S^i\} \), where \( S^i = (t_j^i, v_j^i)_{j=1}^{n_{i}+1} \), the GBTPP model is learned by maximizing the joint log likelihood given as

\[
\sum_{i} \sum_{j} \left( \log P(v_{j+1}^i|h_{j-1}^i, y_j^i) + \log f(t_j^i|h_{j-1}^i, y_j^i) \right)
\]

(12)

where the node propagation probability \( P(v_{j+1}^i|h_{j-1}^i, y_j^i) \) is computed by (7) and the propagation time likelihood \( f(t_j^i|h_{j-1}^i, y_j^i) \) is computed by (9).

To optimize the log likelihood in (12), we implement backpropagation through time (BPTT) to train the GBTPP model. Specifically, supposing that the size of BPTT is \( b \) and the model in Fig. 4 is unrolled by \( b \) steps, then for each training iteration, \( b \) consecutive samples \( \{t_k^i, v_k^i\}_{k=i}^{i+b} \) are offered to apply the feedforward operation through the network. After we unroll the model for \( b \) steps through time, all the parameters are shared across these copies and will be updated sequentially in the backpropagation stage with respect to the loss function.

\section{Experiments}

We evaluate our GBTPP on a synthetic data set and two real-world data sets and compare it with both discrete- and continuous-time sequential models, including recurrent MTPP (RMTTP) [6]. Empirical results show that the GBTPP model achieves better performance on both propagation node prediction and time prediction.

\subsection{Baselines}

For evaluating the predictive performance of forecasting propagation node, we compare GBTPP with the following discrete-time models, including the following.

1) \textit{Majority Prediction}: For each time when making predictions, we always choose the most popular propagation node by frequency count based on all propagations through current node \( v_n \), regardless of propagation history. This is also known as the first-order Markov Chain (MC-1).

2) \textit{Markov Chain}: We also compare with Markov models with higher order, including second and three orders denoted as MC-2 and MC-3, respectively. Instead of considering only \( v_n \), previous propagation node \( v_{n-1} \) and \( v_{n-2} \) is also taken in.

For evaluating the performance of predicting propagation time, we compare with several conventional classical point process models, including the following.

1) \textit{Homogeneous Poisson Process (PP)} [16]: In homogeneous PP, the interevent times are independent and identically distributed random variables conforming to the exponential distribution. The conditional intensity function \( \lambda^*(t) = \lambda_0 \) is a constant over time and independent of the history \( \mathcal{H}_t \), producing an estimate of the average interevent gap.

2) \textit{Hawkes Process (HP)} [13]: As aforementioned in related work, HP is denoted as

\[
\lambda^*(t) = \gamma_0 + \alpha \sum_{t_j < t} \gamma (t, t_j)
\]

where \( \gamma (t, t_j) \geq 0 \) is the triggering kernel measuring temporal dependence and \( \gamma_0 \geq 0 \) is the base intensity independent of the history and the summation of kernel terms is history influence. The kernel function can be chosen in advance, e.g., \( \gamma (t, t_j) = \exp(-\beta(t - t_j)) \) as we used. The intensity function of HP depends on the history up to time \( t \). In general, HP is more expressive than PP as the events in past intervals can affect the occurrence of the events in later intervals.

3) \textit{Self-Correcting Process (SCP)}: It is denoted as

\[
\lambda^*(t) = \exp \left( \mu t - \sum_{t_j < t} \alpha \right)
\]

where \( \mu > 0, \alpha > 0 \). Compared with HP, SCP seeks to produce regular temporal patterns. Though the intensity increases steadily, each time a new event appears, the conditional intensity is decreased by multiplying a constant \( e^{-\alpha} < 1 \).

We also compare with a continuous-time Markov chain (CTMC) model that can jointly predict the node \( v_{n+1} \) and time \( t_n \) for the next propagation step. It learns continuous transition rates between two nodes and makes predictions on the next propagation node with the earliest transition time.

Finally, we compare with the state-of-the-art method RMTTP. Similar to the proposed GBTPP model, when dealing with history influence, the temporal dynamic propagation series are embedded into a history vector by RNN. The major
difference between GBTPP and RMTPP lies in that the structural information of the graph is leveraged in GBTPP through the node vector learned by graph representation, whereas in RMTPP, the event propagation sequence is viewed as generally marked event sequences.

Moreover, to further explore the utility of the graph bias term in the GBTPP model and investigate the usage of the node embedding vector by an ablation study, we compare with a node-specific recurrent point process (NRPP) in which the graph bias term is removed from GBTPP and the node embedding vector only serves as an event profile feature as in the history embedding part in GBTPP. A similar approach is first used in [39] where the event profile feature is incorporated as a regression prior to conventional multidimensional HPs. In NRPP and the history embedding part in GBTPP, the embedding vector of the specific node can be seen as a profile feature of the event. It is also a heuristic way to incorporate graph structural information when we first deal with this problem.

B. Data Set

To verify the potential of the proposed model, we evaluate its performance on two real-world data sets, including the Higgs Twitter data set [5] to predict retweeting between users and the MemeTracker data set [17] to predict meme propagation between websites.

Synthetic: To simulate event propagation processes in graph, e.g., user activities in social networks, we use a multidimensional HP to generate a synthetic data set. HP is widely used to model the generative process of user behavior in social networks, such as [18], [19], and [44]. To generate event propagation sequences in a graph with \( U \) nodes, we set \( U \) HPs that are coupled with each other; each of the HPs corresponds to an individual node and the influence between nodes are explicitly modeled. Specifically, the multidimensional HP is defined by a \( U \)-dimensional point process \( N_u, u = 1, \ldots, U \), with the conditional intensity for the \( u \)th dimension defined as

\[
\lambda_u(t) = \mu_u + \sum_{i < j < u} a_{uu'} g(t - t_i)
\]

where \( \mu_u \geq 0 \) is the base intensity for the \( u \)th HP and \( a_{uu'} \geq 0 \) captures the mutually exciting influence between the \( u \)th and \( u' \)th node. Larger value of \( a_{uu'} \) indicates that events are more likely to propagate from the \( u' \)th node to the \( u \)th node in the future. We collect the parameters into matrix-vector forms with \( \mu = (\mu_u) \) for the base intensity and \( A = (a_{uu'}) \) for the mutually exciting coefficients called infectivity matrix.

In this experiment, we set \( U = 100 \) and generate propagation sequences with randomly initialized parameter \( A \) and \( \mu \). Similar to [44], the base intensity parameters \( \mu \) are sampled from uniform distribution on \([0, 0.001]\), and the infectivity matrix \( A \) is generated by \( A = UV \), where \( U \) and \( V \) are both \( 100 \times 9 \) matrices with entries \([10(i - 1) + 1 : 10(i + 1), i], i = 1, \ldots, 9\) sampled randomly from \([0, 0.1]\) and all other entries are set zero. Then, we scale \( A \) so that the spectral radius of \( A \) is 0.8 to ensure that the point process is well-defined with finite intensity. In the end, we sample 50,000 sequences from the multidimensional HP specified by \( A \) and \( \mu \) for the training and testing of baselines and proposed GBTPP model by tenfold cross validation.

Higgs: The Higgs data set is a public data set built by monitoring the spreading processes on Twitter before, during, and after the announcement of the discovery of a new particle with the features of the elusive Higgs boson on July 4, 2012. Messages between first and seventh July 2012 about this discovery posted on Twitter are considered. There are four directional networks available in the data set based on user activities, including a retweet network (retweeting between users), a reply network (replying to existing tweets), a mention network (mentioning other users), and a social network (friends/followers social relationships among user involved in the abovementioned activities). In the experiment, we study the tweet propagation process using the largest strongly connected component in the directed and weighted retweet network with 984 nodes (users) and 3850 edges. First, a graph embedding \( \{y_k\}_{k=1} \) is learned by graph representation, where \( y_k \) is the embedded node vector for node \( v_k \), and then, the GBTPP model is trained on the retweet activities.

Meme: The MemeTracker data set is also a public data set, which is widely in TPP works [22], [35], [44]. The data set contains the information flows captured by hyperlinks between different sites with timestamps. It tracks meme diffusion over public media, containing more than 172 million news articles or blog posts. The memes are sentences, such as ideas and proverbs, and the time is recorded when it spreads to certain websites. In the experiment, we extract the top 500 popular sites and 62,593 meme propagation cascades between them. First, the adjacent matrix is estimated by \( A_{ij} = N_{ij} / N_{\text{max}} \), where \( N_{ij} \) is the number of observed meme propagations from website \( v_i \) to \( v_j \) in the propagation cascades and \( N_{\text{max}} \) is the global maximum number of meme propagations between websites. Given adjacent matrix \( A_{ij} \), the graph embedding \( \{y_k\}_{k=1} \) is learned where \( y_k \) is the embedded node vector for website \( v_k \), and then, the GBTPP model is trained on meme propagation cascades.

Our experiments are conducted under Ubuntu 64 bit 16.04LTS, with i5-8600K 3.60 GHz × 6 CPU, 16-GB RAM, and NVIDIA GeForce GTX 1070Ti GPU. All the experimental results are given by tenfold cross validation. It is worth mentioning that it is required to maintain the integrity of the propagation process data sample when we train and test our model. Therefore, instead of deleting a chunk of the propagation sequence and train on the rest, we consider a sequence as one complete and independent sample when dividing the whole data set into ten subsets to perform tenfold cross validation.

To elaborate the tenfold cross validation implementation, given a collection \( C \) of observed event propagation sequences \( C = \{S^i, S^{i+1}, \ldots\} \), where \( S^i \) is the \( i \)-th propagation sequence \( S^i = \{(t^{i}, v^{i}_{1}, (t^{i}, v^{i}_{2}), \ldots\} \) recording the event propagation process in graph \( G = (V, E) \), the collection \( C \) is randomly
MemeTracker data sets in Fig. 6. The top-5 node prediction on the Higgs Twitter and MemeTracker data sets. A propagator node ˆ\(v\) can act as a recommender system recommending next propagation.

In fact, our model is widely used for recommender systems. In fact, our model is widely used for recommender systems.

Fold cross validation can be implemented: Of the ten subsets = \(S\) \(C\), one single subset is retained as the validation data for testing the model, and the remaining nine subsets are used as training data.

Specifically, we then prepare training or testing samples from the training or validation data set for our model, including input data and output label. For a sequence, e.g., of length \(N\), \(S = ((t_1, v_1), (t_2, v_2), \ldots, (t_N, v_N))\) in the training or validation data, we acquire \(N-1\) training or testing samples as we observe \(N-1\) propagations between nodes, e.g., the propagation from form node \(v_n\) to \(v_{n+1}\), the model input includes propagation history \(H_{n-1} = ((t_1, v_1), (t_2, v_2), \ldots, (t_{n-1}, v_{n-1}))\) and current node \(v_n\), and the output label is node \(v_{n+1}\) and propagation time \(t_n\).

C. Experimental Results

We use prediction accuracy (# correct predictions divide total predictions) to evaluate propagation node prediction and root mean square error (RMSE) to evaluate propagation time prediction. The empirical results with standard deviation are presented in Table I.

Moreover, we further compute the top-\(K\) precision curve for propagation node prediction on the Higgs Twitter and MemeTracker data sets in Fig. 6. The top-\(K\) precision curve is widely used for recommender systems. In fact, our model can act as a recommender system recommending next propagation node \(v_{n+1}\) in the period of propagation time \(t_n\), e.g., recommending interested tweets for user \(v_{n+1}\) at the time around \(t_n\) or recommending popular news and memes to the editors of website \(v_{n+1}\) around time \(t_n\).

Specifically, we have the following findings and discussions.

1) Recurrent Model Versus Parametric Model: As shown in Table I, GBTPP and RMTPP outperform conventional parametric methods such as Markov chain and point process models such as PP, HP, and SCP, as well as joint prediction model CTMC on both propagation node prediction and time prediction. The main advantage lies in that conventional models make strong assumptions on the distribution form and generative process of the data, whereas GBTPP and RMTPP use RNNs to automatically learn the influences from propagation history.

2) RMTPP Versus GBTPP: Compared with the state-of-the-art RMTPP model, the GBTPP model achieves better performance than RMTPP, especially on real-world data sets. Though we use a multidimensional HP to simulate event propagation sequences, the actual graph structure cannot be simulated such as real-world data set. Correspondingly, it explains to the results that the GBTPP model achieves comparably better performance than RMTPP on real-world data sets than the synthetic one.

Two major innovations contribute to the promotion of GBTPP model compared with RMTPP model: 1) the structural information of the graph is used in the form of node embedding as side information and 2) the direct influence between connected node is separately measured as an extra bias term from the indirect influence of propagation history. It verifies our hypothesis that as a special case of event sequence modeling, event propagation modeling in the graph requires a more suitable model to deal with the structural information and reflect the fact that event propagation is more likely to happen between the connected nodes in graph.

3) NRPP Versus GBTPP: As shown in Table I and Fig. 6, while the NRPP model achieves comparable results against RMTPP as it incorporates node embedding vectors as node-specific profile feature, the GBTPP model still has distinct superiority over NRPP on both numerical results and top-\(K\) precision curve. These empirical results suggest that though the better performance of

| Model       | MC-1 | MC-2 | MC-3 | CTMC | RMTPP | NRPP | GBTPP | PP    | HP    | SCP   | CTMC  | RMTPP | NRPP | GBTPP |
|-------------|------|------|------|------|-------|------|------|-------|-------|-------|-------|-------|------|-------|
| Metrics     | Accuracy (%) | RMSE |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Synthetic   | 17.46 | 25.27 | 33.74 | 32.08 | 46.82 | 46.63 | 47.26 | 3.457 | 2.164 | 2.845 | 3.420 | 1.852 | 1.905 | 1.728 |
|             | (2.24) | (2.53) | (1.87) | (2.74) | (1.38) | (1.56) | (1.55) | (0.374) | (0.283) | (0.317) | (0.265) | (0.241) | (0.256) | (0.228) |
| Higgs       | 10.92 | 14.60 | 16.35 | 17.41 | 22.26 | 22.53 | 24.59 | 3.267 | 2.518 | 2.343 | 2.355 | 1.741 | 1.622 | 1.396 |
|             | (2.06) | (1.44) | (1.73) | (2.58) | (1.90) | (1.64) | (1.29) | (0.381) | (0.346) | (0.369) | (0.335) | (0.272) | (0.275) | (0.264) |
| Meme        | 15.72 | 20.05 | 22.93 | 25.56 | 32.14 | 32.75 | 35.82 | 2.361 | 1.958 | 1.484 | 1.762 | 1.059 | 0.979 | 0.825 |
|             | (2.21) | (2.14) | (1.59) | (2.17) | (1.52) | (1.97) | (1.73) | (0.412) | (0.368) | (0.276) | (0.347) | (0.254) | (0.242) | (0.227) |

Fig. 6. Top-5 node prediction accuracy on the Higgs and MemeTracker data sets.
GBTTP model benefits from introducing graph structural information, the utility method, e.g., modeling direct and indirect influence using graph bias term in GBTTP, is more important. The improvement is quite limited only by introducing graph embedding vector and higher network complexity.

V. Conclusion

TPPs are widely used for modeling event sequences, whereas event propagation sequence modeling is rarely considered as a special case, where the structural information and direct connections between nodes are not utilized. In this article, we study the problem of event propagation modeling by the GBTTP. Compared with the state-of-the-art method, we have two innovations: 1) the direct influence between the connected nodes is separately measured as an extra bias term from the indirect influence of the propagation history, through prelearned graph representation and 2) when modeling the indirect influence of the propagation history, the structural information of the graph is used in the form of node embedding as side information. We evaluate the GBTTP model on the Higgs Twitter data set predicting retweeting in social network and the MemeTracker data set predicting meme propagation between the websites. Experimental results collaborate on the effectiveness of our approach compared with conventional methods and state-of-the-art method.

As a future work, the GBTTP can be extended to modeling event propagation in dynamic networks by introducing graph embedding method, e.g., [28] over dynamic temporal graphs. In addition, the semantic information of the event, e.g., text information of news and tweets if it is available, can be incorporated into the model to improve the performance.

REFERENCES

[1] R. Begleiter, R. El-Yaniv, and G. Yona, “On prediction using variable order Markov models,” J. Artif. Intell. Res., vol. 22, pp. 385–421, Dec. 2004.
[2] A. Bordes, S. Chopra, and J. Weston, “Question answering with subgraph embeddings,” in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2014, pp. 615–620.
[3] H. Cai, V. W. Zheng, and K. C.-C. Chang, “A comprehensive survey of graph embedding: Problems, techniques, and applications,” IEEE Trans. Knowl. Data Eng., vol. 30, no. 9, pp. 1616–1637, Sep. 2018.
[4] D. Daley and D. Vere-Jones, An Introduction to the Theory of Point Processes: General Theory and Structure, vol. 2. Springer, 2007.
[5] M. De Domenico, A. Lima, P. Mougel, and M. Musolesi, “The anatomy of a scientific rumor,” Sci. Rep., vol. 3, no. 1, p. 2980, Dec. 2013.
[6] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song, “Recurrent marked temporal point processes: Embedding event history to vector,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining. New York, NY, USA: ACM, 2016, pp. 1555–1564.
[7] P. Embrechts, T. Liniger, and L. Liu, “Multivariate Hawkes processes: An application to financial data,” J. Appl. Probab., vol. 48, no. A, pp. 367–378, Aug. 2011.
[8] S. Ertikin, C. Radin, and T. H. McCormick, “Reactive point processes: A new approach to predicting power failures in underground electrical systems,” Ann. Appl. Statist., vol. 9, no. 1, pp. 122–144, Mar. 2015.
[9] T. R. Etherington and G. L. W. Perry, “Using point process intensity to establish the spatio-temporal grain of continuous landscape tessellations and graphs,” Landscape Ecol., vol. 27, no. 8, pp. 1085–1090, Oct. 2012.
[10] J. Feng et al., “Gake: Graph aware knowledge embedding,” in Proc. 26th Int. Conf. Comput. Linguistics, Tech. Papers (COLING), 2016, pp. 641–651.
[11] E. W. Fox, M. B. Short, F. P. Schoenberg, K. D. Coronges, and A. L. Bertozzi, “Modeling e-mail networks and inferring leadership using self-exciting point processes,” J. Amer. Stat. Assoc., vol. 111, no. 514, pp. 564–584, Apr. 2016.
[12] A. G. Hawkes, “Point spectra of some mutually exciting point processes,” J. Roy. Stat. Soc., B (Methodol.), vol. 33, no. 3, pp. 438–443, Oct. 1971.
[13] A. G. Hawkes, “Spectra of some self-exciting and mutually exciting point processes,” Biometrika, vol. 58, no. 1, pp. 83–90, 1971.
[14] J. Janssen and N. Limnios, Semi-Markov Models and Applications. Springer, 2013.
[15] S. K. Karmaker Santu, L. Li, D. H. Park, Y. Chang, and C. Zhai, “Modeling the influence of popular trending events on user search behavior,” in Proc. 26th Int. Conf. World Wide Web Companion (WWW Companion), Geneva, Switzerland: International World Wide Web Conference Committee, 2017, pp. 535–544.
[16] J. F. C. Kingman, Poisson Processes. Hoboken, NJ, USA: Wiley, 1993.
[17] J. Leskovec, L. Backstrom, and J. Kleinberg, “Meme-tracking and the dynamics of the news cycle,” in Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), New York, NY, USA: ACM, 2009, pp. 497–506.
[18] L. Li and H. Zha, “Dryadic event attribution in social networks with mixtures of Hawkes processes,” in Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage. (CIKM), New York, NY, USA: ACM, 2013, pp. 1667–1672.
[19] L. Li and H. Zha, “Learning parametric models for social infectivity in multi-dimensional Hawkes processes,” in Proc. 28th AAAI Conf. Artif. Intell., 2014, pp. 101–107.
[20] I. Liu, W. K. Cheung, S. Li, and L. Liao, “Aligning users across social networks using network embedding,” in Proc. 25th IJCAI, New York, NY, USA, Jul. 2016, pp. 1774–1780.
[21] T. Man, H. Shen, S. Liu, X. Jin, and X. Cheng, “Predict anchor links across social networks via an embedding approach,” in Proc. 25th Int. Joint Conf. Artif. Intell. (IJCAI), New York, NY, USA, Jul. 2016, pp. 1823–1829.
[22] H. Mei and J. M. Eisner, “The neural Hawkes process: A neurally self-modulating multivariate point process,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 6754–6764.
[23] Y. Ogata, “Statistical models for earthquake occurrences and residual analysis for point processes,” J. Amer. Stat. Assoc., vol. 83, no. 401, pp. 9–27, Mar. 1988.
[24] Y. Ogata, “Space-time point-process models for earthquake occurrences,” Ann. Inst. Stat. Math., vol. 50, no. 2, pp. 379–402, Jun. 1998.
[25] R. Pent Outline, “A survey of random processes with reinforcement,” Probab. Surv., vol. 4, pp. 1–79, 2007.
[26] A. Reinhart, “A review of self-exciting spatio-temporal point processes and their applications,” Stat. Sci., vol. 33, no. 3, pp. 299–318, Aug. 2018.
[27] M. C. Seiler and F. A. Seiler, “Numerical recipes in C: The art of scientific computing,” Risk Anal., vol. 9, no. 3, pp. 415–416, 1989.
[28] U. Singer, I. Guy, and K. Radinsky, “Node embedding over temporal graphs,” 2019, arXiv:1903.08889. [Online]. Available: http://arxiv.org/abs/1903.08889.
[29] A. Stomakhin, M. B. Short, and A. L. Bertozzi, “Reconstruction of missing data in social networks based on temporal patterns of interactions,” Inverse Problems, vol. 27, no. 11, Nov. 2011, Art. no. 115013.
[30] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, “LINE: Large-scale information network embedding,” in Proc. 24th Int. Conf. World Wide Web (WWW), Florence, Italy, May 2015, pp. 1067–1077.
[31] D. Wang, C. Song, and A.-L. Barabási, “Quantifying long-term scientific impact,” Science, vol. 342, no. 6154, pp. 137–142, Oct. 2013.
[32] Y. Wang, H. Shen, S. Liu, J. Gao, and X. Cheng, “Cascade dynamics modeling with attention-based recurrent neural network,” in Proc. 26th Int. Joint Conf. Artif. Intell., Melbourne, VIC, Australia, Aug. 2017, pp. 2985–2991.
[33] A. Wei, L. Xu, B. Cao, and P. S. Yu, “Cross view link prediction by learning noise-resilient representation consensus,” in Proc. 26th Int. Conf. World Wide Web. Geneva, Switzerland: International World Wide Web Conference Committee, Apr. 2017, pp. 1611–1619.
[34] W. Wu, J. Yan, X. Yang, and H. Zha, “Decoupled learning for factorial marked temporal point processes,” in Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining. New York, NY, USA: ACM, Jul. 2018, pp. 2516–2525.
[35] S. Xiao, M. Farajtabar, X. Ye, J. Yan, L. Song, and H. Zha, “Wasserstein learning of deep generative point process models,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 3247–3257.
[36] S. Xiao, J. Yan, X. Yang, H. Zha, and S. M. Chu, “Modeling the intensity function of point process via recurrent neural networks,” in Proc. 31st AAAI Conf. Artif. Intell., 2017, pp. 1597–1603.

[37] H. Xu, W. Wu, S. Nemati, and H. Zha, “Patient flow prediction via discriminative learning of mutually-correcting processes,” IEEE Trans. Knowl. Data Eng., vol. 29, no. 1, pp. 157–171, Jan. 2017.

[38] J. Yan et al., “Towards effective prioritizing water pipe replacement and rehabilitation,” in Proc. 33rd Int. Joint Conf. Artif. Intell., 2013, pp. 2931–2937.

[39] J. Yan et al., “Modeling contagious merger and acquisition via point processes with a profile regression prior,” in Proc. IJCAI, 2016, pp. 2690–2696.

[40] J. Yan et al., “On machine learning towards predictive sales pipeline analytics,” in Proc. 29th AAAI Conf. Artif. Intell., 2015, pp. 1945–1951.

[41] A. Zammit-Mangion, M. Dewar, V. Kadirkamanathan, and J. Yan et al., “Point process modelling of the afghan war diary,” Proc. Nat. Acad. Sci. USA, vol. 109, no. 31, pp. 12414–12419, Jul. 2012.

[42] C. Zhang et al., “Regions, periods, activities: Uncovering urban dynamics via cross-modal representation learning,” in Proc. 26th Int. Conf. World Wide Web. Geneva, Switzerland: International World Wide Web Conferences Steering Committee, Apr. 2017, pp. 361–370.

[43] Q. Zhang and H. Wang, “Not all links are created equal: An adaptive embedding approach for social personalized ranking,” in Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR), Pisa, Italy, Jul. 2016, pp. 917–920.

[44] K. Zhou, H. Zha, and L. Song, “Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes,” in Proc. AISTATS, 2013, pp. 641–649.

[45] S. Zhu and Y. Xie, “Spatial-temporal-textual point processes with applications in crime linkage detection,” 2019, arXiv:1902.00440. [Online]. Available: http://arxiv.org/abs/1902.00440

Weichang Wu received the B.S. degree in electronic engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2013. He is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China.

His current research interests include machine learning and data mining, especially sequential data analysis and temporal point processes.

Hua Xu was born in Hunan, China, in 1982. He received the master’s and Ph.D. degrees in pattern recognition and intelligent system from Shanghai Jiao Tong University, Shanghai, China, in 2007 and 2010, respectively.

He is currently a Senior Engineer with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University. His research interests include pattern recognition, computer vision, and machine learning.

Xiaohui Zhang received the master’s degree in software engineering from Northwest University, Xi’an, China, in 2012.

In 2016, he was responsible for leading the implementation of Orange Cloud Cutting, China Telecom BestPay Company Ltd., Shanghai, China. He is currently the Senior Director and the Head of the Innovation Center, China Telecom BestPay Company Ltd.

Mr. Zhang received the Special Contribution Award that is used to commend him for his special contributions to the company in intelligent operation and maintenance, blockchains, system architecture, and so on in 2018.

Yu Liu received the master’s degree in communication and information system from Donghua University, Shanghai, China, in 2009.

He is currently a Senior Engineer with China Telecom BestPay Company Ltd., Shanghai. He has published in DBAplus “Building an automated database monitoring system with Zabbix and Ansible,” “Redis retrieval performance is insufficient, and reshaping beat to solve historical slow log tracking.”

Hongyuan Zha received the Ph.D. degree in scientific computing from Stanford University, Stanford, CA, USA, in 1993.

He is currently a Professor with the School of Computational Science and Engineering, College of Computing, Georgia Institute of Technology, Atlanta, GA, USA, and an Adjunct Professor with Shanghai Jiao Tong University, Shanghai, China, and East China Normal University, Shanghai. Since 1993, he has been working on information retrieval, machine learning applications, and numerical methods.

Prof. Zha was a recipient of the Leslie Fox Prize (second prize) of the Institute of Mathematics and its Applications in 1991, the Outstanding Paper Awards of the 26th International Conference on Advances in Neural Information Processing Systems (NIPS 2013), and the Best Student Paper Award (advisor) of the 34th ACM SIGIR International Conference on Information Retrieval (SIGIR 2011). He was an Associate Editor of the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING.