Marked Temporal Dynamics Modeling Based on Recurrent Neural Network

Yongqing Wang(✉), Shenghua Liu, Huawei Shen, Jinhua Gao, and Xueqi Cheng

CAS Key Laboratory of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{wangyongqing,gaojinhua}@software.ict.ac.cn, {liushenghua,shenhuawei,cxq}@ict.ac.cn

Abstract. We are now witnessing the increasing availability of event stream data, i.e., a sequence of events with each event typically being denoted by the time it occurs and its mark information (e.g., event type). A fundamental problem is to model and predict such kind of marked temporal dynamics, i.e., when the next event will take place and what its mark will be. Existing methods either predict only the mark or the time of the next event, or predict both of them, yet separately. Indeed, in marked temporal dynamics, the time and the mark of the next event are highly dependent on each other, requiring a method that could simultaneously predict both of them. To tackle this problem, in this paper, we propose to model marked temporal dynamics by using a *mark-specific* intensity function to explicitly capture the dependency between the mark and the time of the next event. Experiments on two datasets demonstrate that the proposed method outperforms the state-of-the-art methods at predicting marked temporal dynamics.

Keywords: Marked temporal dynamics · Recurrent neural network · Event stream data

1 Introduction

There is an increasing amount of event stream data, i.e. a sequence of events with each event being denoted by the time it occurs and its mark information (e.g. event type). Marked temporal dynamics offers us a way to describe this data and potentially predict events. For example, in microblogging platforms, marked temporal dynamics could be used to characterize a user's sequence of tweets containing the posting time and the topic as mark [9]; in location based social networks, the trajectory of a user gives rise to a marked temporal dynamics, reflecting the time and the location of each check-in [15]; in stock market, marked temporal dynamics corresponds to a sequence of investors' trading behaviors, i.e., bidding or asking orders, with the type of trading as mark [4]; An ability to predict marked temporal dynamics, i.e., predicting when the next event will take place and what its mark will be, is not only fundamental to understanding

J. Kim et al. (Eds.): PAKDD 2017, Part I, LNAI 10234, pp. 786–798, 2017.
DOI: 10.1007/978-3-319-57454-7_61

the regularity or patterns of these underlying complex systems, but also has important implications in a wide range of applications, from viral marketing and traffic control to risk management and policy making.

Existing methods for this problem fall into three main paradigms, each with different assumptions and limitations. The first category of methods focuses on predicting the mark of the next event, formulating the problem as a discrete-time or continuous-time sequence prediction task [12,25]. These methods gained success at modeling the transition probability across marks of events. However, they lack the power at predicting when the next event will occur.

The second category of methods, on contrary, aims to predict when the next event will occur [10]. These methods either exploit temporal correlations for prediction [20,22] or conduct prediction by modeling the temporal dynamics using certain temporal process, such as self-exciting Hawkes process [2,6], various Poisson process [9,21], and other auto-regressive processes [8,16]. These methods have been successfully used in modeling and predicting temporal dynamics. However, these models are unable to predict the mark.

In recent years, researchers attempt to directly model the marked temporal dynamics [11]. A recent work [7] used recurrent neural network to automatically learn history embedding, and then predict both, yet separately, the time and the mark of the next event. This work assumes that time and mark are independent on each other given the historical information. Yet, such assumption fails to capture the dependency between the time and the mark of the next event. For example, when you have lunch is affected by your choice on restaurants, since different restaurants imply difference in geographic distance and quality of service. The separated prediction by maximizing the probability on mark and time does not imply the most likely event. In sum, we still lack a model that could capture the interdependency of mark and time when predicting the next event.

In this paper, we propose a novel model based on recurrent neural network (RNN), named RNN-TD, to capture the dependence between the mark of an event and its occurring time. The key idea is to use a mark-specific intensity function to model the occurring time for events with different marks. The benefits of our proposed model are three-fold: (1) It models the mark and the time of the next event simultaneously; (2) The mark-specific intensity function explicitly captures the dependency between the occurring time and the mark of an event; (3) The involvement of RNN simplifies the modeling of dependency on historical events.

We evaluate the proposed model by extensive experiments on large-scale real world datasets from Memetracker[1] and Dianping[2]. Compared with several state-of-the-art methods, RNN-TD outperforms them at prediction of marks and times. We also conduct case study to explore the capability of event prediction in RNN-TD. The experimental results indicate that it can better model marked temporal dynamics.

[1] http://www.memetracker.org.
[2] http://www.dianping.com.

2 Model

In this paper, we focus on the problem of modeling marked temporal dynamics. Before diving into the details of the proposed model, we first clarify two main motivations underlying our model.

2.1 Motivation

In real scenarios, mark and time of next event are highly dependent on each other. We use a case from Dianing to illustrate this phenomenon. We extract the trajectories starting from the same location (mark #6) and examine if the time interval between two consecutive events are discriminative to each other with respect to different marks. The distribution of time interval with different target marks are represented in Fig. 1(a). We can observe that large variance exists in the distributions when consumers make different choices. This motivates us to model mark-specific temporal dynamics.

Second, existing works [12] attempted to formulate marked temporal dynamics by Markov random processes with varying orders. However, the generation of next event requires strong prior knowledge on dependency of history. Besides, long dependency on history causes state-space explosion problem in practice. Therefore, we propose a RNN-based model which learns the dependency by deep structure. It embeds history information into vectorized representation when modeling sequences. The generation of next event is only dependent on history embedding.

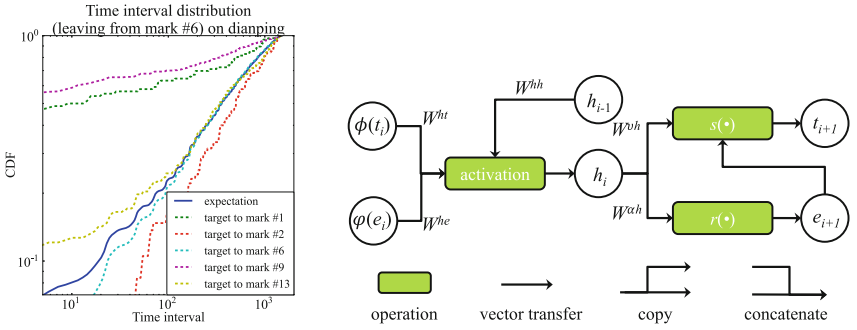


Fig. 1. (a) High variance existed in time interval distribution when targeting to different marks. (b) The architecture of RNN-TD. Given the event sequence $S = \{(t_i, e_i)\}_{i=1}$, the i-th event (t_i, e_i) is mapped through function $\phi(t)$ and $\varphi(e)$ into vector spaces as inputs in RNN. Then the inputs $\phi(t_i)$ and $\varphi(e_i)$ associated with the last embedding h_{i-1} are fed into hidden units in order to update h_i. Dependent on embedding h_i, RNN-TD outputs the next event type e_{i+1} and correspondent time t_{i+1}.

2.2 Problem Formulation

An event sequence $S = \{(t_i, e_i)\}$ is a set of events in ascending order of time. The tuple (t_i, e_i) records the i-th event in the sequence S, and the variables $t_i \in \mathcal{T}$ and $e_i \in \mathcal{E}$ denote the time and the mark respectively, where $\mathcal{E}$ is a countable state space including all possible marks and $\mathcal{T} \in \mathbb{R}^+$ is the time space in which observed marks take place. We could have various instantiation in different applications.

The likelihood of an observed sequence S can be written as

$$P(S) = \prod_{i=1}^{|S|} p(t_i, e_i | H_{t_i}),$$

where $H_{t_i} = \{(t_l, e_l) | t_l < t_i, e_l \in \mathcal{E}\}$ refers to all the historical events occurring before t_i. In practice, the joint probability of a pair of mark and time can be written by Bayesian rule as follows

$$p(t_i, e_i | H_{t_i}) = r(e_i | H_{t_i}) s(t_i | e_i, H_{t_i}), \tag{1}$$

where $r(e_i | H_{t_i})$ refers to the probability that the mark of next event is e_i and $s(t_i | e_i, H_{t_i})$ is the probability distribution function of time given a specific mark.

Next we propose a general model to parameterize $r(e_i | H_{t_i})$ and $s(t_i | e_i, H_{t_i})$ in marked temporal dynamics modeling, named RNN-TD. Recurrent neural network (RNN) is a feed-forward neural network for modeling sequential data. In RNN, the current inputs are fed into hidden units by nonlinear transformation, jointly with the outputs from the previous hidden units. The feed-forward architecture is replicative in both inputs and outputs so that the representation of hidden units is dependent on not only current inputs but also encoded historicial information. The adaptive size of hidden units and nonlinear activation function (e.g., sigmoid, tangent hyperbolic or rectifier function) make neural network capable of approximating arbitrary complex function [3].

The architecture of RNN-TD is depicted in Fig. 1(b). The inputs of an event (t_i, e_i) is vectorized by mapping function $\phi(\cdot)$ and $\varphi(\cdot)$. Then the i-th inputs associated with the last embedding h_{i-1} are fed into hidden units in order to update h_i. Given the i-th event (t_i, e_i), the embedding h_{i-1} and mapping function ϕ and φ, the representation of hidden units in RNN-TD can be calculated as

$$h_i = \sigma\left(W^{ht}\phi(t_i) + W^{he}\varphi(e_i) + W^{hh}h_{i-1}\right), \tag{2}$$

where σ is the activation function, and W^{ht}, W^{he} and W^{hh} are weight matrices in neural network. The procedure is iteratively executed until the end of sequence. Thus, the embedding h_i encodes the i-th inputs and the historical context h_{i-1}.

Based on the history embedding h_i, we can derive the probability of the $(i+1)$-th event in an approximative way,

$$p(t_{i+1}, e_{i+1} | H_{t_{i+1}}) \approx p(t_{i+1}, e_{i+1} | h_i) = r(e_{i+1} | h_i) s(t_{i+1} | e_{i+1}, h_i). \tag{3}$$

Firstly we formalize the conditional transition probability $r(e_{i+1}|h_i)$. The conditional transition probability can be derived by a softmax function which is commonly used in neural network for parameterizing categorical distribution, that is,

$$r(e_{i+1}|h_i) = \frac{\exp\left(W_k^{\alpha h} h_i\right)}{\sum_{j=1}^{K} \exp\left(W_j^{\alpha h} h_i\right)}, \tag{4}$$

where row vector $W_k^{\alpha h}$ is k-th row of weight matrix indexed by the mark e_{i+1}.

Then we consider the probability distribution function $s(t_{i+1}|e_{i+1}, h_i)$. The probability distribution function describes *the observation that nothing but mark e_{i+1} occurred until time t_{i+1} since the last event*. We define a random variable T_e as the occuring time of next event with mark e, and the probability distribution function $s(t_{i+1}|e_{i+1}, h_i)$ can be formalized as

$$s(t_{i+1}|e_{i+1}, h_i) = P(T_{e_{i+1}} = t_{i+1}|e_{i+1}, h_i) \prod_{e \in \mathcal{E} \setminus e_{i+1}} P(T_e > t_{i+1}|e_{i+1}, h_i), \tag{5}$$

where the probability $P(T_e > t_{i+1}|e_{i+1}, h_i)$ depicts that the occuring time of event with mark e is out of the range $[0, t_{i+1}]$, and $P(T_{e_{i+1}} = t_{i+1}|e_{i+1}, h_i)$ is the conditional probability density function representing the fact that mark e_{i+1} occurs at t_{i+1}.

To formalize the Eq. (5), we define *mark-specific conditional intensity function* [1]

$$\lambda_e(t_{i+1}) = \frac{f_e(t_{i+1}|e_{i+1}, h_i)}{1 - F_e(t_{i+1}|e_{i+1}, h_i)}, \tag{6}$$

where $F_e(t_{i+1}|e_{i+1}, h_i)$ is the cumulative distribution function of $f_e(t_{i+1}|e_{i+1}, h_i)$, referring to the probability that mark e_{i+1} will happen in $[0, t_{i+1}]$. According to Eq. (6), we can derive the cumulative distribution function

$$F_e(t_{i+1}|e_{i+1}, h_i) = 1 - \exp(-\int_{t_i}^{t_{i+1}} \lambda_e(\tau) d\tau). \tag{7}$$

Thus, we have $P(T_e > t_{i+1}|e_{i+1}, h_i) = 1 - F_e(t_{i+1}|e_{i+1}, h_i)$. Then we can derive the mark-specific conditional probability density function by Eq. (7) as

$$P(T_e = t_{i+1}|e_{i+1}, h_i) = f_e(t_{i+1}|e_{i+1}, h_i) = \lambda_e(t_{i+1}) \exp(-\int_{t_i}^{t_{i+1}} \lambda_e(t) dt). \tag{8}$$

Substituting Eqs. (7) and (8) into the likelihood of Eq. (5), we can get

$$s(t_{i+1}|e_{i+1}, h_i) = \lambda_{e_{i+1}}(t_{i+1}) \exp(-\int_{t_i}^{t_{i+1}} \lambda(t) dt), \tag{9}$$

where $\lambda(\tau) = \sum_{e \in \mathcal{E}} \lambda_e(\tau)$ is the summation of all conditional intensity function.

The key to specify probability distribution function $s(t_{i+1}|e_{i+1}, h_i)$ is parameterization of mark-specific conditional intensity function λ_e. We parameterize λ_e conditioned on h_i as follows,

$$\lambda_e(t) = \nu_e \cdot \tau(t; t_i) = \exp\left(W_k^{\nu h} h_i\right) \tau(t; t_i), \tag{10}$$

where row vector $W_k^{\nu h}$ denotes to the k-th row of weight matrix corresponding to mark e. In Eq. (10), the mark-specific conditional intensity function is splited into two parts: $\nu_e = \exp(W_{j'}^{\nu h} h_i)$ is a nonnegative scalar as the constant part with respect to time t, and $\tau(t; t_i) \geq 0$ refers to an arbitrary time shaping function [10]. For simplicity, we consider two well-known parametric models for time shaping function: exponential and constant, i.e., $\exp(wt)$ and c.

Given a collection of event sequences $\mathcal{C} = \{S_m\}_{m=1}^N$, we suppose that each event sequence S_m is independent on each other. As a result, the logarithmic likelihood of a set of event sequences is the sum of the logarithmic likelihood of the individual sequence. Given the source of event sequence, the negative logarithmic likelihood of the set of event sequences $\mathcal{C}$ can be estimated as,

$$\mathcal{L}(\mathcal{C}) = -\sum_{m=1}^{N} \sum_{i=1}^{|S_m|-1} \left[W_k^{\alpha h} h_i - \log \sum_{j=1}^{K} \exp\left(W_j^{\alpha h} h_i\right) + W_k^{\nu h} h_i + \log \tau(t; t_i) - \sum_{e \in \mathcal{E}} \exp\left(W_{j'}^{\nu h} h_i\right) \int_{t_i}^{t_{i+1}} \tau(t; t_i) dt \right].$$

In addition, we want to induce sparse structure in vector ν in order that not all event types are available to be activated based on h_i. For this purpose, we introduce lasso regularization on ν, i.e., $\|\nu\|_1$ [23]. Overall, we can learn parameters of RNN-TD by minimizing the negative logarithmic likelihood

$$\arg\min_{W} \mathcal{L}(\mathcal{C}) + \gamma \|\nu\|_1, \tag{11}$$

where γ is the trade-off parameter.

Finally, we estimate the next most likely events in two steps by RNN-TD: (1) estimate the time of each mark by expectation $t_{i+1} = \int_{t_i}^{\infty} t \cdot s(t|e_{i+1}, h_i) dt$; (2) calculate the likelihood of events according to the mark-specific expectation time, and then rank events in descending order of likelihood.

3 Optimization

In this section, we introduce the learning process of RNN-TD. We apply back-propagation through time (BPTT) [5] for parameter estimation. With BPTT method, we need to unfold the neural network in consideration of sequence size $|S_m|$ and update the parameters once after the completed forward process in sequence. We employ Adam [13], an efficient stochastic optimization algorithm, with mini-batch techniques to iteratively update all parameters. We also apply early stopping method to prevent overfitting in RNN-TD. The stopping criterion is achieved when the performance has no more improvement in validation

set. The mapping function of $\phi(t)$ is defined by temporal features associated with t, e.g., logarithm time interval $\log(t_i - t_{i-1})$ and discretization of numerical attributes on year, month, day, week, hour, mininute, and second. Besides, we employ orthogonal initialization method for RNN-TD in order to speed up convergence in training process. The embedding learned by word2vec [18,19] is used to initialize the parameter of mapping function $\varphi(e)$. The good initialization provided by the embedding can speed up convergence for RNN [17].

4 Experiments

Firstly, we introduce baselines, evaluation metrics and datasets of our experiments. Then we conduct experiments on real data to validate the performance of RNN-TD in comparison with baselines.

4.1 Baselines

Both mark prediction and time prediction are evaluated, and the following models are chosen for comparisons in the two prediction tasks.

(1) Mark sequence modeling.
- **MC:** The markov chain model is a classic sequence modeling method. We compare with markov chain with order varying from one to three, denoted as MC1, MC2 and MC3.
- **RNN:** RNN is a state-of-the-art model for discrete time sequence, successfully applied in language model. To fairly justify the performance between RNN and our proposed method, we use the same inputs in both RNN and RNN-TD.

(2) Temporal dynamics modeling. We choose point processes and mark-specific point processes with different characterizations as baselines.
- **PP-poisson:** The intensity function related to mark is parameterized by a constant, depicting the leaving rate from last event.
- **PP-hawkes:** The intensity function related to mark e is parameterzied by

$$\lambda(t; e) = \lambda(0; e) + \alpha \sum_{t_i < t} \exp\left(-\frac{t - t_i}{\sigma}\right), \tag{12}$$

  where $\sigma = 1$ and $\lambda(0; e)$ is a intrinsic rate defined on mark e when $t = 0$.
- **MSPP-poisson:** We define the mark-specific intensity function by a parametric matrix, depicting the rate from one mark to another.
- **MSPP-hawkes:** The mark-specific intensity function is parameterized by Eq. (12) where the constant rate is specialized according to mark pairs in parametric matrix.

  We also compare with the model that has the ability to generate both mark and temporal sequences.
- **RMTPP:** Recurrent marked temporal point process (RMTPP) [7] is a method which independently models both mark and time information based on RNN.

4.2 Evaluation Metrics

Serveral evaluation metrics are used when measuring the performance in mark prediction and time prediction tasks. We regard the mark prediction task as a ranking problem with respect to transition probability. The prediction performance is evaluated by *Accuracy* on top k (Acc@k) and *Mean Reciprocal Rank* (MRR) [24]. On time prediction task, we define tolerance θ over the prediction error between estimated time and practical occuring time. The prediction accuracy on time prediction with respect to tolerance θ is formulated as,

$$\text{Acc@}\theta = \frac{\sum_{m=1}^{N}\sum_{i=1}^{|S_m|-1}\delta\left(|E(t; e_{i+1}, h_i) - t_{i+1}| < \theta\right)}{\sum_{m=1}^{N}(|S_m| - 1)},$$

where δ is an indicator function. Larger scores in Acc@k, MRR and Acc@θ indicate better predictions.

4.3 Datasets

We conduct experiments on two real datasets from two different scenarios to evaluate the performance of different methods:

Table 1. Performance of mark prediction on two datasets

		MRR	Acc@1	Acc@3	Acc@5	Acc@10	Acc@20
Memetracker	MC1	0.4634	0.2948	0.4595	0.6659	0.8253	0.9209
	MC2	0.4788	0.3155	0.4706	0.6773	0.8301	0.9186
	MC3	0.4670	0.3149	0.4583	0.6550	0.7891	0.8619
	RNN	0.4780	0.3202	0.4746	0.6825	0.8315	0.9201
	RMTPP	0.4833	0.3241	0.4834	0.6926	0.8386	0.9267
	RNN-TD(c)	0.4820	0.3220	0.4790	0.6895	0.8393	0.9270
	RNN-TD(exp)	0.4849	0.3266	0.4835	0.6929	0.8400	0.9273
	RNN-TD*(c)	0.4820	0.3220	0.4790	0.6895	0.8393	0.9270
	RNN-TD*(exp)	**0.4851**	**0.3266**	**0.4844**	**0.6937**	**0.8407**	**0.9274**
Dianping	MC1	0.6174	0.5231	0.6157	0.7212	0.7963	0.8787
	MC2	0.6260	0.5280	0.6396	0.7393	0.8007	0.8513
	MC3	0.5208	0.4462	0.5395	0.6035	0.6332	0.6569
	RNN	0.6355	0.5123	0.6135	0.7153	0.7905	0.8656
	RMTPP	0.6620	0.5482	0.6554	0.7578	0.8271	0.8935
	RNN-TD(c)	0.6663	0.5524	0.6601	0.7628	0.8346	0.8999
	RNN-TD(exp)	0.6635	0.5448	0.6560	0.7638	0.8345	0.8988
	RNN-TD*(c)	**0.6663**	**0.5524**	**0.6602**	0.7628	0.8346	**0.8999**
	RNN-TD*(exp)	0.6635	0.5452	0.6566	**0.7641**	**0.8351**	0.8990

p.s. the experimental results from * are dependent with given time.

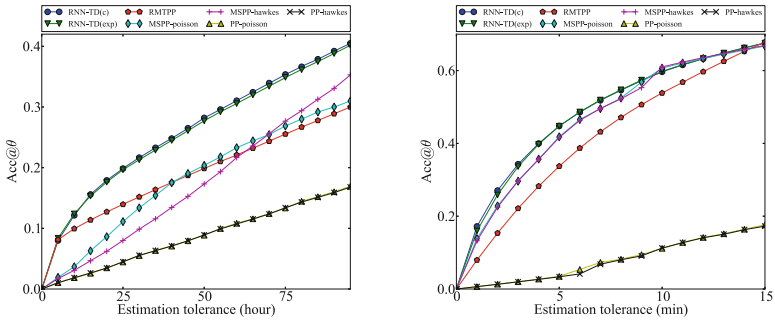
- **Memetracker** [14]**:** Memetracker corpus contains articles from mainstream media and blogs from August 1 to October 31, 2008 with about 1 million documents per day. Contents in the corpus are organized according to topics by the proposed method in [14]. We use top 165 frequent topics and organize the posting sequence about posted blogs and post-time by users. The whole posting sequence of each user is splited into parts as follows, (1) get the statistics of time intervals between two consecutive posted blogs, (2) empirically estimate the period of user's posting behavior, (3) and divide the whole sequence into several parts according to the estimated period. We do not consider the sequences whose length are less than 3. The obtained dataset contains 1,481,491 posting sequences, and the time interval between two consecutive blogs is ranged from 2.77×10^{-4} to 99.68 h.
- **Dianping:** Dianping provides an online restaurant rating service in China, including coupon sales, bill payment, and reservation. We extract transaction coupon sales from top 256 popular stores located in Xidan bussiness district of Beijing from year 2011 to 2015. The consumption sequences of users are divided into segments as the same steps done in memetracker. Because of the existence of sparse shopping records in users, we also limit that time interval between two consecutive consumptions is two months. The processed dataset contains 221,893 event sequences, and the time interval between two consecutive consumptions is ranged from 2.77×10^{-4} to 1440 h.

On both datasets, we randomly pick up 80% of completed sequences in datasets as training, and the rest sequences are divided into two parts equally as validation set and test set respectively.

4.4 Performance of Mark Prediction

The performance of mark prediction is evaluated using metrics Acc@k and MRR. The experimental results are shown in Table 1. Comparing with MC1, MC2, MC3 and RNN, RNN-TD(c) and RNN-TD(exp) achieve significant improvements over all metrics in both datasets. In Memetracker, RNN-TD(exp) outperforms RMTPP in MRR at significance level of 0.1, and achieve a little improvements than RMTPP in Acc@1,3,5,10 and 20. However, the performance of RNN-TD(c) is worse than RMTPP. In Dianping, RNN-TD(c) achieves improvements than RMTPP in metrics of MRR and Acc@5 at significance level of 0.1 and metrics of Acc@10 and Acc@20 at significance level of 0.01. Besides, RNN-TD(exp) achieves improvements than RMTPP in metrics of Acc@20 at significance level of 0.1 and metrics of Acc@5 and Acc@10 at significance level of 0.01. The experimental results indicate that RNN-TD can better learn the mark generation by jointly optimizing mark-specific conditional intensity function with respect to different time shaping function applied in tasks.

We also conduct experiments according to event likelihood on RNN-TD with the given time, marked as RNN-TD*. The results of RNN-TD*(exp) performs little better than RNN-TD(exp) over all metrics in both datasets, However, the performance of RNN-TD*(c) is almost the same as RNN-TD(c). It demonstrates



(a) Experiments on Memetracker (b) Experiments on Dianping

Fig. 2. Performance of timing prediction on two datasets.

the robustness of RNN-TD on mark prediction whether or not given the occuring time. Besides, RNN-TD with exponential form of time shaping function has larger effects on given time than the constant form.

4.5 Performance of Time Prediction

We evaluate the performance of time prediction by Acc@θ. The predictions of RNN-TD and MSPP are based on true marks. Fig. 2(a) and (b) show the experimental results of RNN-TD and baselines on memetracker and dianping. As shown in Fig. 2, without considering any mark information, PP-poisson and PP-hawkes are unable to handle the temporal dynamics well on both Memetracker and Dianping. MPP can discriminate mark-specific time-cost, leading to better performance than PPs. In memetracker dataset, although RMTPP has better performance than PP, it does not overbeat MSPP-poisson and MSPP-hawkes. In dianping dataset, RMTPP(c) and RMTPP(exp) achieve better performance than MSPP-hawkes when tolerance $\theta \leq 65\,\mathrm{h}$, and also achieve better performance than MPP-poisson when tolerance $\theta \leq 15\,\mathrm{h}$. It is seen that RNN-TD(c) and RNN-TD(exp) achieve the best performance than all the baselines in the most cases on two datasets. The improvements achieved by RNN-TD indicate that our proposed method can well model marked temporal dynamics by learning mark-specific intensity functions, while RMTPP share the same intensity function for all the marks. Note that the variance of time distribution is quite larger in Dianping than Memetracker. Thus we need to give a smaller α in Dianping when training PP-hawkes and MSPP-hawkes model, leading to the similar performance than PP-poisson and MSPP-poisson shown in Fig. 2(b).

4.6 Case Study on Event Prediction

To explore the capability of event prediction of RNN-TD, we randomly choose one specific event sequence from memetracker and dianping respectively, and estimate the next events in the sequence. In RNN-TD, we select top 3 events in

Table 2. Case study on event prediction

(a) One specific event sequence prediction on memetraker				
i-th event: mark,time (mins)		1th	2nd	3rd
RMTPP	c#1	Europe debt, 22.32	Europe debt, 12.29	Europe debt, 60.31
	c#2	LinkedIn IPO, 22.32	Dominique Strauss, 12.29	Amy Winehouse, 60.31
	c#3	Amy Winehouse, 22.32	LinkedIn IPO, 12.29	Dominique Strauss, 60.31
RNN-TD	c#1	Europe debt, 1.07	Dominique Strauss, 1.34	Dominique Strauss, 3.63
	c#2	Dominique Strauss, 0.45	Europe debt, 1.29	Europe debt, 3.12
	c#3	LinkedIn IPO, 0.44	LinkedIn IPO, 0.56	attack, 2.93
Ground truth		Dominique Strauss, 6.37	attack, 83.78	attack, 18.18
(b) One specific event sequence prediction on dianping				
i-th event: mark,time (days)		1th	2nd	3rd
RMTPP	c#1	bibimbap, 2.34	bibimbap, 2.90	Sichuan cuisine, 3.02
	c#2	tea restaurnt, 2.34	cookies, 2.90	cookies, 3.02
	c#3	Yunnan cuisine, 2.34	Sushi, 2.90	tea restaurnt, 3.02
RNN-TD	c#1	bibimbap, 2.93	barbecue, 0.65	barbecue, 0.96
	c#2	Yunnan cuisine, 0.88	bibimbap, 0.85	Sichuan cuisine, 0.81
	c#3	bread, 0.92	Vietnamese cuisine, 0.51	bread, 0.48
Ground truth		barbecue,0.14	Sichuan cuisine,1.03	barbecue,1.06

descending order of event likelihood as candidates of next event, called c#1, c#2 and c#3. In RMTPP, we choose the most probable mark and expectation time independently and combine them as the candidates of next event. Table 2 lists the performance of RMTPP and RNN-TD. We can see that the predicted marks on RNN-TD are more accurate and relevant to ground truth than compared methods on both cases. Then, we categorize most relevant marks by empirical knowledge to evaluate the estimated time on mark-specific methods when marks are mismatched in all 3 candidates. For example, we consider bibimbap and barbecue belong to same regional cuisine, and Dominique Strauss is related to Europe debt. In this way, the average error of time prediction to ground truth for RNN-TD is 34.55 min, and the average error is up to 43.19 min for RMTPP in the case of Memetrack. In the case of Dianping, the average error of time prediction to ground truth for RNN-TD is 1.13 days, and the average error is nearly doubled to 2.04 days for RMTPP. Indeed, RNN-TD can provide more options according to possible event predictions which has more general applications, e.g., recommendation systems.

5 Conclusions

In this paper, we proposed a general model for marked temporal dynamics modeling. Based on RNN framework, the representation of hidden layer in RNN-TD learns the history embedding through a deep structure. The generation of marks and times is dependent on history embedding so that we can avoid strong prior knowledge on dependency of history. We observe that the generation processes of next event are significant different with respect to marks. To capture the dependence between marks and times, we unfolded the joint probability of mark

and time and parameterized the mark transition probability and mark-specific conditional intensity function based on history embedding. We evaluate the effectiveness of our proposed model on two real-world datasets from memetracker and dianping. Experimental results demonstrate that our model consistently outperforms existing methods at mark prediction and time prediction tasks. Moreover, we conduct case study on event prediction demonstrating that our proposed model is well applicable in marked temporal dynamics modeling.

Acknowledgments. This work was funded by the National Basic Research Program of China (973 Program) under Grant Numbers 2013CB329602 and 2014CB340401, and the National Natural Science Foundation of China under Grant Numbers 61472400, 61572467, 61433014. H. W. Shen is also funded by Youth Innovation Promotion Association CAS and the CCF-Tencent RAGR (No. 20160107).

References

1. Arjas, E., Keiding, N., Borgan, O., Andersen, P.K., Natvig, B.: Survival models and martingale dynamics. Scand. J. Stat. **16**(3), 177–225 (1989)
2. Bao, P., Shen, H.W., Jin, X., Cheng, X.Q.: Modeling and predicting popularity dynamics of microblogs using self-excited hawkes processes. In: Proceedings of the 24th International Conference on World Wide Web, pp. 9–10 (2015)
3. Barzel, B., Liu, Y.Y., Barabási, A.L.: Constructing minimal models for complex system dynamics. Nat. Commun. **6**, Article no. 7186 (2015)
4. Cao, L., Ou, Y., Yu, P.S., Wei, G.: Detecting abnormal coupled sequences and sequence changes in group-based manipulative trading behaviors. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 85–94 (2010)
5. Chauvin, Y., Rumelhart, D.E.: Backpropagation: Theory, Architectures, and Applications. Psychology Press, Abingdon (1995)
6. Crane, R., Sornette, D.: Robust dynamic classes revealed by measuring the response function of a social system. Proc. Natl. Acad. Sci. **105**(41), 15649–15653 (2008)
7. Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., Song, L.: 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, pp. 1555–1564 (2016)
8. Engle, R.F., Russell, J.R.: Autoregressive conditional duration a new model for irregularly spaced transaction data. Econometrica **66**, 1127–1162 (1998)
9. Gao, S., Ma, J., Chen, Z.: Modeling and predicting retweeting dynamics on microblogging platforms. In: Proceedings of the 8th ACM International Conference on Web Search and Data Mining, pp. 107–116 (2015)
10. Gomez-Rodriguez, M., Leskovec, J., Schlkopf, B.: Modeling information propagation with survival theory. In: Proceedings of the 30th International Conference on Machine Learning, vol. 28, pp. 666–674 (2013)
11. Gunawardana, A., Meek, C.: Universal models of multivariate temporal point processes. In: Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, pp. 556–563 (2016)
12. Isaacson, D.L., Madsen, R.W.: Markov Chains, Theory and Applications, vol. 4. Wiley, New York (1976)

13. Kingma, D.P., Adam, J.B.: Adam: A method for stochastic optimization. In: International Conference on Learning Representation (2015)
14. Leskovec, J., Backstrom, L., Kleinberg, J.: Meme-tracking and the dynamics of the news cycle. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 497–506 (2009)
15. Liu, Q., Wu, S., Wang, L., Tan, T.: Predicting the next location: a recurrent model with spatial and temporal contexts (2016)
16. Vaz de Melo, P.O.S., Faloutsos, C., Assunção, R., Loureiro, A.: The self-feeding process: a unifying model for communication dynamics in the web. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 1319–1330 (2013)
17. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
18. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. Adv. Neural Inf. Process. Syst. **26**, 3111–3119 (2013)
19. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710 (2014)
20. Pinto, H., Almeida, J.M., Gonçalves, M.A.: Using early view patterns to predict the popularity of youtube videos. In: Proceedings of the 6th ACM International Conference on Web Search and Data Mining, pp. 365–374 (2013)
21. Shen, H., Wang, D., Song, C., Barabási, A.L.: Modeling and predicting popularity dynamics via reinforced Poisson processes. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence, pp. 291–297 (2014)
22. Szabo, G., Huberman, B.A.: Predicting the popularity of online content. Commun. ACM **53**(8), 80–88 (2010)
23. Tibshirani, R.: Regression shrinkage and selection via the Lasso. J. R. Stat. Soc. Ser. B (Methodol.) **58**, 267–288 (1996)
24. Voorhees, E.M.: The TREC8 question answering track report. In: Text REtrieval Conference (1999)
25. Wang, Y., Shen, H., Liu, S., Cheng, X.: Learning user-specific latent influence and susceptibility from information cascades. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence, pp. 477–483 (2015)