CPNSA: Cascade Prediction with Network Structure Attention

Chaochao Liu[1], Wenjun Wang[1], Pengfei Jiao[2], Yueheng Sun[1](✉), Xiaoming Li[3], and Xue Chen[4]

[1] College of Intelligence and Computing, Tianjin University, Tianjin 300350, China
{chaochaoliu,wjwang,yhs,lxm696}@tju.edu.cn

[2] Center of Biosafety Research and Strategy, Tianjin University, Tianjin 300072, China
pjiao@tju.edu.cn

[3] School of International Business, Zhejiang Yuexiu University of Foreign Languages, Zhejiang, China
lxm696@tju.edu.cn

[4] Law School, Tianjin University, Tianjin 300072, China
xuechen@tju.edu.cn

Abstract. Online social medias provide convenient platforms for information spread, which makes the social network structure plays important role on online information spread. Although online social network structure can be obtained easily, few researches use network structure information in the cascade of the resharing prediction task. In this paper, we propose a cascade prediction method (named by CPNSA) involves the network structure information into cascade prediction of resharing task. The method is based on the recurrent neural network, and we introduce a network structure attention to incorporates the network structure information into cascade representation. In order to fuse network structure information with cascading time series data, we use network embedding method to get the representations of nodes from the network structure firstly. Then we use the attention mechanism to capture the structural dependency for cascade prediction of resharing. Experiments are conducted on both synthetic and real-world datasets, and the results show that our approach can effectively improve the performance of the cascade prediction of resharing.

Keywords: Cascade prediction · Deep learning · Network structure influences · Recurrent neural network · Cascade behavior

1 Introduction

Online social networks provide a new type of means for information spreading, which makes information spreading is affected by the online social networks. The exposed users are more likely to spread information than the not exposed one,

H. Gao et al. (Eds.): CollaborateCom 2020, LNICST 349, pp. 64–79, 2021.
https://doi.org/10.1007/978-3-030-67537-0_5

and weak ties play a more important role than strong ties [2]. The user more likely takes retweet action when he received information from multiple social neighbors, and the information spread farther and faster across clustered-lattice networks than across random networks [5].

The research of modeling information spreading data can be used in many fields, such as online social topic detection [4], online post's influence evaluation [37], public opinion monitoring [22], and marketing [23]. Information cascade modeling and prediction is one of these important researches. In this paper, we focus on the research of modeling information cascade, and improve the precision of cascade prediction.

There are already so many methods have been proposed from different perspectives to solve the cascade modeling issues. The wide used models are independent cascade model [6,8], linear threshold model [13], and their variants [3,9]. These models assume that the underlying diffusion pattern is known a priori, while the real-world data may not be like this. The optimization methods of the models are so complex that these models handle large-scale data difficultly. Point process based models are proposed to learn the dependency between the users who spread information [35,41], those models suppose the information spread cascade sequence data following the point process. However, the effectiveness of these models heavily depends on the carefully designed expressions of the point process, and the optimization methods are complex and difficult to compute in parallel. To solve these issues, researchers proposed neural network based models, which do not require an explicit priori assume and can process large-scale data by using GPU Computing. According to the researches [19,24,29,34,40], the neural network based models achieve better performances than the non-neural-network based sequential approaches. Thus, we propose our model based on the neural network to modeling and predicting information spreading cascade sequence data in this paper.

Recently, the neural network based model have been proposed to process the information spreading cascade modeling and predicting task. Wang *et al.* [29] proposed an attention-based recurrent neural network model to learn the cross-dependence influences between the users in information cascades' sequence data. But they did not actually use social network structure information. Wang *et al.* [30] proposed a sequential neural network based model with social network structure attention, to incorporate the social network structural information in information spreading cascade predicting. But they only consider users' neighborhood information. Due to the sparsity of the network, many legitimate links are missing [28]. As a result, their model is not sufficient to involve the network structure information. And because their model make social network adjacency matrix as input, the model can hardly process information spreading cascade data across large-scale social network.

To solve these issues, in this paper, we use the social network embedding method SDNE [28] to get the representations of nodes firstly. SDNE is proposed by Tang *et al.*, which can well maintain the local and global network structure by using the first-order and second-order proximity [27]. In this way, we can solve the missing links of social network issue, and reduce the calculations of training by reducing the representation dimension of social network nodes. Drawing on previous researches, we use the recurrent neural network to model the information spreading cascade sequence data. Thus, we can get the hidden representations of information spreading cascade sequences. To merge the social network structure information with information spreading cascade sequential information, we introduce a social network structure attention neural network layer before the output neural network layer. The proposed model is recognized as CPNSA, which is based on the recurrent neural network for cascade prediction with social network structure attention.

In the proposed model, we use the obtained users' social network embeddings by the SDNE as query vectors. Thus, the model not only considers the historical sequential state of activated users but also various types of social network structure information is transmitted during the cascade behavior modeling. In this way, the model can predict the next activated user precisely. We apply the proposed method on 6 synthetic datasets and the Digg real-world dataset with three baseline methods to compare the prediction performance of next activated users.

It is worthwhile to highlight several contributions of the CPNSA model here:

- We proposed a new model to deal with the modeling and predicting cascades of information resharing tasks, the model is robust by using the neural network.
- We consider social network structure information via attention mechanism in cascades of information resharing prediction task, which makes the method be able to capture nodes' dependency in network structure.
- Recently proposed network embedding method SDNE [28] is used, which makes the method be able to involve the first-order and second-order proximity between nodes. Thus, our model can deal with the missing links in social network.
- Our model utilizes the result of social network embedding, which reduces space complexity and more precise than the model using edges directly.
- CPNSA performs better than the other compared algorithms.

2 Related Work

In this part, we present recent researches related to our work, which mainly contains the researches of network structure's affection on the information diffusion and neural network based models for information spreading cascade modeling. Then we show our point of view on the existing information spreading cascade modeling researches. All these observations indeed motivate the work of this paper.

Social network structure plays important role on the information spreading, recent researchers attempt to involve social network structure information in the information spreading modeling tasks. Huang *et al.* [12] proposed a method utilizes community information for influence maximization task, the experiments' results validate social network structure's influence on the information spreading. Li *et al.* [17] investigate the close correspondence between social tie in information spreading process. Zhang *et al.* researched on the social influence and found that the social users' retweet actions are influenced by their close friends in their social ego networks [39]. Weng *et al.* [31] used social network structure information to predict memes on the social medias, their proposed model showed good performance on the prediction task of future popularity of a meme given its early spreading patterns by using the social network community structure. Su *et al.* [26] worked on the social contagious research and showed that there is an optimal social network community structure can maximizes spreading dynamics. Nematzadeh *et al.* [21] investigated the social network community structure effects on information spreading, and they found that global information spreading speed can be enhanced by the strong communities in online social network. Wu *et al.* [32] worked on the social network multi-community structure effects on information spreading research, and they found that the social network multi-community structure can facilitate the online social information spreading process. Qiu *et al.* [24] designed an end-to-end framework for feature representation learning to predict social influence. However, there are few types of research consider social network structure's influence in online social network information spreading cascade prediction models.

Types of research have been proposed to modeling the social network information spreading, such as Matchbox model [25,38], multiple additive regression trees (MART) model [33], maximum entropy classifier model [1], autoregressive-moving-average (ARMA) model [20], factor graph model [36], conditional random fields model [23] and so on. Because of the specific assumptions of the models, these models can hardly generalized to other datasets. And these models process large-scale data difficultly because of the complex optimization methods. Recently, researchers attempt to use deep neural networks to address above issues inspired by the recent success of deep neural networks in many other applications. These works mostly use the recurrent neural network (RNN) to learn the hidden dependences between the retweet users and time patterns of spreading action. Finally, the models can get the representations of information spreading cascade. Since these models do not require knowing the underlying information spreading model, they can process real-world data robust. In this paper, we also use the deep neural networks to build our model for dealing with the cascade prediction task.

There are already few models based on the neural networks. Xiao *et al.* [34] used two recurrent neural networks to build their model, one recurrent neural network is used to interprets the conditional intensity function of a point process, and the other recurrent neural network is used to learn the time patterns.

Zhang *et al.* [40] used neural network embedding method to transport the tweet contents, the social network user interests, the similarity information between the tweet content and social user interests, social user information and author of tweet information into neural network representations. They proposed an attention mechanism to encode the interests of the social users. Their model finally to predict whether a tweet will be retweeted by a user. Wang *et al.* [29] showed that each cascade generally corresponds to a diffusion tree, causing cross-dependence in cascade, so they proposed an attention-based recurrent neural network to capture the cross-dependence in cascade. Liu *et al.* [19] followed independence cascade model, they defined parameters for every social user with a latent influence vector and a susceptibility vector. The proposed model can be used to learn information spreading cascade dynamics. However, these models not actually use social network structure information. Wang *et al.* [30] proposed a diffusion model based on recurrent neural network, and involved the social network structure information by proposed a social network structure attention. But their model take social network adjacent matrix as input, which make the model can hardly process large-scale social network and difficult to handle the missing links in social networks. Liu *et al.* [18] proposed a cascade prediction model with community structure enhanced, but their model only focused on the community structure information.

Thus, in this paper, we use social network embedding tool to get a low dimensional representation for users, which can solve the missing links issue. We choose SDNE [28] to get social network embedding, since SDNE can learn local and global social network structure information. To merge the social network structure information with the information spreading cascade sequence data, we propose an social network structure attention layer to restrict the information spreading cascade representation.

3 Proposed Method

3.1 Problem Definition

In this paper, we focus on the task of further take retweet action users prediction and retweet time of the user prediction. The input data of our proposed model are social network structure data and information spreading cascade sequence data. The social network and information spreading cascade are defined as the following:

Definition 1. **Social Network.** A social network contains nodes and edges, and the edges represent the relationships between the nodes. We denote the set of nodes as V, the set of edges as E and the social network as $G = (V, E)$. Thus the number of nodes is $|V|$ and the number of edges is $|E|$, where $|\cdot|$ represent the size of the set.

Definition 2. **Information Spreading Cascade.** A information spreading cascade is a set of sequence information spreading data, it contains retweet users and retweet time. We denote the cascade as $S = \{(t_i, v_i)|v_i \in V, t_i \in [0, +\infty), t_i \leq t_{i+1}, i = 1, 2, ..., N\}$, it start from a original post user and the post time, and the retweet users and the corresponding retweet time are ascendingly ordered by time. We denote N as the number of users take part in the information spreading of the post, and denote the retweet data at time t_i as (t_i, v_i), where v_i represents the $i-$th retweet user.

Given the above definitions, we formulate our problem in the following part. The input data of our proposed model contains a social network $G = (V, E)$, and a collection of F information spreading cascades denote as $Q = \{S_f\}_{f=1}^{F}$. For one of the observed cascade, we denote the sequence data up to the $k-$th retweet behavior as $S_{\leq k}$. Our proposed model take the data input and learn the information spreading patterns. The trained model can be used to calculate the probability of the next take retweet action user v_{k+1} and the action time $t_{v_{k+1}}$. Thus, the probability of the next take retweet action user can be represented as $p(v_{k+1}|S_{\leq k})$.

3.2 Model Framework

Our proposed model is based on recurrent neural network, which can process the large-scale data easily. And we proposed an new social network structure attention to merge the social network structure information with the information spreading cascade sequence data. The proposed model is named by CPNSA, to solve the information spreading cascade prediction problem. The rationale of our model is that we use the recurrent neural network to learn the historical cascade sequential state of retweet users and retweet time. At the same time, we use the social network structure attention to learn the social network structure's effects on the information spreading actions. Based on these ideas, we propose a new deep learning-based cascade behavior modeling framework containing social network structure information. The system framework of the method is shown in Fig. 1. CPNSA mainly uses recurrent neural network (RNN) to model sequence dependence. In order to incorporate the impact of network structure, we propose a social network structure attention model to involve both local and global network structure.

Sequence Modeling. Our model employs two recurrent neural networks (RNNs) to model the user sequence and time sequence respectively. In each RNN, a hidden state is used to memorize the summarized history. In each step k of a cascade, the node v_k is represented as a low-dimensional vector $\mathbf{v}_k \in R^{d_v}$ through a mapping matrix $\mathbf{W}_{emv}$. The node embedding vector is represented as $\mathbf{v}_k = \mathbf{W}_{emv}^T v_k$ with the dimension d_v. Then, the hidden state representation of nodes' activity at step k can be $\mathbf{h}_k^{(0)} = RNN(\mathbf{v}_k, \mathbf{h}_{k-1}^{(0)})$ by using RNN. In addition, for the timing input t_k, we using inter-event duration $t_k - t_{k-1}$ as the temporal features $\mathbf{t}_k$. The hidden state representation of time sequence at step k can be $\mathbf{h}_k^{(1)} = RNN(\mathbf{t}_k, \mathbf{h}_{k-1}^{(1)})$ by using RNN. $\mathbf{h}_0^{(0)}$ and $\mathbf{h}_0^{(1)}$ are initialized as all zero vector.

Social Network Structure Attention. Considering that the influences of nodes in a social network are different, it is important to identify those key users and help extract representations of cascades. Wang *et al.* [30] proposed a network structure attention based on RNN. However, the query vector in [30] was

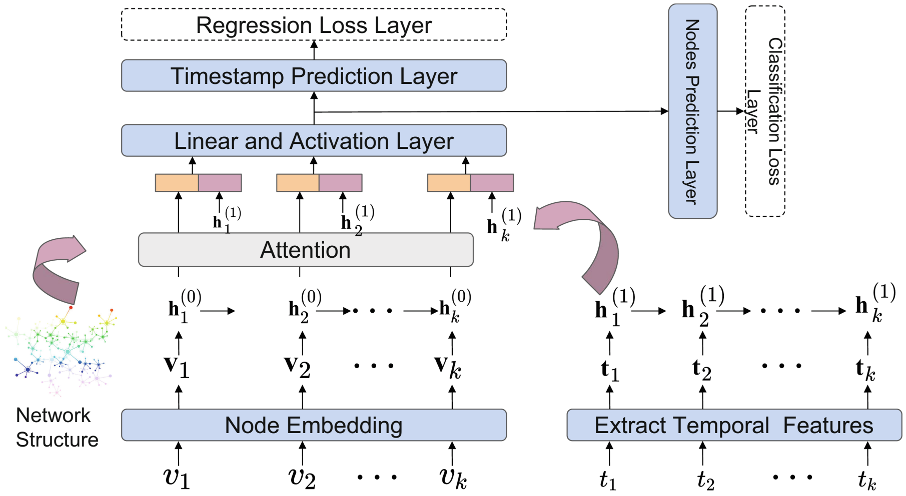

Fig. 1. Our recurrent neural network based model can be trained end-to-end. Users are converted to low-dimensional vectors by the user embedding layer, then feed into RNN network, and we can get hidden state vectors $\mathbf{h}_1^{(0)}, \mathbf{h}_2^{(0)}, ..., \mathbf{h}_h^{(0)}$. Time series are extracted as inter-event duration, and feed into recurrent neural network, finally we can get hidden state vectors $\mathbf{h}_1^{(1)}, \mathbf{h}_2^{(1)}, ..., \mathbf{h}_h^{(1)}$. We can get users' embedding vectors via SDNE [28]. Then the hidden state vectors $\mathbf{h}_1^{(0)}, \mathbf{h}_2^{(0)}, ..., \mathbf{h}_h^{(0)}$ can be transformed by using the social network structure attention layer, and concatenated with the hidden state vectors of time features feed into linear and activation layers, we can get the cascade's representation. Then two prediction layers are used to output the predicted activated node, and the associated timestamp. Cross-entropy and square loss are respectively used for event type and timestamp prediction.

computed complexity and can only utilize oversimplified neighbor information of network structure. Also that, many legitimate links are missing due to the sparsity of networks, which causing the Wang's model hardly model the data precisely. As a result, our model CPNSA is developed to consider both local and global social network structure information.

Thus, we use the social network embedding method SDNE [28] to represent the frist-order and second-order proximity of users into low-dimensional vectors. In this way, the global and local social network structure information can be involved in the social network embeddings. Given the network G of the training set and testing set, composed of N nodes and E edges, SDNE can learn highly non-linear network structure by using multiple nonlinear mapping functions to map the input data to a highly nonlinear latent space. SDNE also uses both the second-order and first-order proximity to capture the global and local network structure. Due to the sparsity of networks, SDNE imposes more penalty to the reconstruction error of the non-zero elements than that of zero elements. Following SDNE, we minimize the loss function to obtain embeddings of nodes $\mathbf{H}^{(e)}$:

$$\mathcal{L} = \mathcal{L}_{2nd} + \alpha\mathcal{L}_{1st} + \beta\mathcal{L}_{reg} \tag{1}$$

following the SDNE's definitions, we denote the $\mathcal{L}_{2nd}$ as the second-order proximity loss, and the $\mathcal{L}_{1st}$ as the first-order proximity loss. We denote the $\mathcal{L}_{reg}$ as an $\mathcal{L}2$-norm regularizer term in this paper. The α and the β represent the balance parameters.

Given the social network nodes' embeddings, we compute the effection of node at step i on the node at step k by using attention with the embedding of $\mathbf{h}_k^{(0)}$ as query vectors, like the Eq. (2):

$$\gamma_{k,i} = \frac{exp(\mathbf{h}_k^{(0)} A(\mathbf{h}_i^{(e)})^T)}{\sum_{j=1}^{k} exp(\mathbf{h}_k^{(0)} A(\mathbf{h}_j^{(e)})^T))} \tag{2}$$

where $\mathbf{h}_i^{(e)}$ is the social network structure embedding of node at step i, A is a parameter matrix, $\gamma_{k,i}$ is effection of node at step i on node at step k. We denote $\mathbf{h}_k^{(2)}$ as the final hidden representation of node sequence at step k, just like Eq. (3):

$$\mathbf{h}_k^{(2)} = \sum_{i=1}^{k} \gamma_{k,i}\mathbf{h}_i^{(0)} \tag{3}$$

Next Activated Node and Time Generation. The hidden representation of cascade at step k is given by merge the hidden representations $\mathbf{h}_k^{(1)}$ and $\mathbf{h}_k^{(2)}$:

$$\mathbf{h}_k^{cas} = \delta(\mathbf{W}_h^T(\mathbf{h}_k^{(1)} \oplus \mathbf{h}_k^{(2)}) + \mathbf{b}_h) \tag{4}$$

where $\mathbf{W}_h^T$ is the weight, $\mathbf{b}_h$ is the bias, $\oplus$ is the connection operation and $\delta()$ represents the activation function, and in this paper we use the PReLU activation function.

Next Activated Node Generation. The model adds a linear layer and an activation layer to project the hidden representation into the same space with the node embedding showed in Eq. (5):

$$\mathbf{h}_k^{node} = \delta(\mathbf{W}_{node}^T\mathbf{h}_k^{cas} + \mathbf{b}_{node}) \tag{5}$$

where $\mathbf{W}_{node}^T$ is the weight, and $\mathbf{b}_{node}$ is the bias. Finally, we calculate cosine similarities between the hidden vector $\mathbf{h}_k^{node}$ with the embedding vectors of all the nodes, and use a softmax layer to generate the probability distribution of next infected node as follow:

$$\mathbf{p}_k^{node} = softmax(\mathbf{h}_k^{node}\mathbf{W}_{emv}^T) \tag{6}$$

where $\mathbf{p}_k^{node} \in R^N$, N is the number of nodes.

Next Activated Time Generation. Based on the hidden representation of cascade at step k, we can generate the time inter-event duration between step $k+1$ and step k by adding a linear layer following Eq. (7)

$$t_{k+1} - t_k = \mathbf{W}_t^T\mathbf{h}_k^{cas} + \mathbf{b}_t \tag{7}$$

where $\mathbf{W}_t^T$ is the weight, and $\mathbf{b}_t$ is the bias.

3.3 Optimization

We introduce our learning process of the model as bellow. Given a collection of cascades $Q = \{S_f\}_{f=1}^{F}$, we treat the cascades are independent on each other. Thus, we can learn the model by maximizing the joint log-likelihood of observing Q in Eq. (8):

$$\mathbf{Loss}(Q) = \sum_{f=1}^{F} \sum_{i=1}^{N_f - 1} logp((t_{k+1}, v_{k+1}) | c_{v_k}, S_{\leq k}) \tag{8}$$

which is the sum of the logarithmic likelihood for all the individual cascades. We exploit backpropagation through time (BPTT) for training our model. In each training iteration, we vectorize activated nodes' information as inputs, including nodes' embedding, nodes' embedding calculated from network structure information and inter-event duration temporal features. At last, we apply stochastic gradient descent (SGD) with mini-batch and the parameters are updated by Adam [14]. To speed up the convergence, we use orthogonal initialization method in training process [10]. We also employ clips gradient norm for the parameters to prevent overfitting.

4 Experimental Setup

In this section, we introduce the data sets, the comparison methods, and the evaluation metrics used in the experiments to quantitatively evaluate the proposed framework.

4.1 Data Sets

Our experiments are conducted on two types of data sets—synthetic data and real-word data.

Synthetic Data. The data generation consists of two parts: network generation and cascade generation. We use two network generation tools to generate networks. The first network generation tool is following from previous work [29,30], we apply Kronecker graph model [16] to generate random network (RD) with the parameter matrix [0.5 0.5; 0.5 0.5]. We construct a network with default 1024 users and avenge 20°. The second network generation tool is the LFR benchmark proposed by Lancichinetti *et al.* [15], which is the popular used synthetic networks containing community structure generator. We set the average degree of nodes to be 20, the maximum degree of nodes to be 50, power-law exponent for the degree distribution to be 2, power-law exponent for the community size distribution to be 1. Then we generate two networks contain 500 nodes (LFR500) and 1000 (LFR1000) nodes separately. In the cascade generation part, for each activated node, we set the activation time of an activated user following a certain time distribution. Similar with Wang's setup, we choose two-time distributions for sampling: 1) mixed exponential (Exp) distributions, controlled by rate parameters in [0.01, 10]; 2) mixed Rayleigh (Ray) distribution, controlled by scale parameters in [0.01, 10] [29]. The cascade generation progress uses breadth-first to search for next activated node, and the progress will stop until the overall time exceeds the threshold ω or no node is activated. We set $\omega = 100$. Finally, six synthetic data sets are generated by different combinations of network scale and propagation time distributions, denoted by (RD, Exp), (RD, Ray), (LFR500, Exp), (LFR500, Ray), (LFR1000, Exp), and (LFR1000, Ray). We generate 20 cascades per node in each dataset and randomly pick up 80% of cascades for training and the rest for validation and test.

Real World Data. The Digg dataset proposed by Nathan *et al.* is used in this paper. The dataset contains diffusions of stories as voted by the users, along with friendship network of the users [11]. We drop the cascades with size larger than 1,000, as the large cascade rarely occurs in practice and may dominate the training process [29]. We randomly pick up 80% of cascades for training and the rest for validation and test.

4.2 Comparison Methods

For comparison with the proposed model, we evaluate the following methods on the data sets.

RNNPP [34]: The method of RNNPP takes a recurrent neural network (RNN) perspective to point process and models its background and history effect. The model can be used to predict event timestamp, main-type event and sub-type event. In this paper, we consider nodes as main-types and do not use the sub-type event prediction layer.

Recurrent Marked Temporal Point Processes (RMTPP) [7]: The method of RMTPP views the intensity function of a temporal point process as a nonlinear function of the history, and uses a recurrent neural network to automatically learn a representation of influences from the event history. The RMTPP can be applied in activated nodes timestamp and activated nodes prediction for information cascade.

Sequential Neural Network with Structure Attention (SNNSA) [30]: The SNNSA is a recently proposed method to model information diffusion, which can capture the structural dependency among users by using attention mechanism. However, the method only considers local network structure.

4.3 Evaluation Metrics

Our task is predicting next activated node, and next activated timestamp, given previously cascade information. Since the number of potential nodes is huge, we can regard the prediction task as a ranking problem with users' transition probabilities as their scores [29]. Each model outputs the infection probability distribution over all users and the actual infected user is expected to get the highest probability [30]. Thus, we evaluate the proposed method and comparison method by accuracy on top k (Acc@k) and mean reciprocal rank (MRR) which are the widely used metrics. For timestamp prediction, we use the root-mean-square error (RMSE) which measures the difference between the predicted time point and the actual one.

5 Experimental Results

5.1 Synthetic Data Results

Table 1, Table 2, and Table 3 show the prediction comparisons of next activated node on baselines and our proposed model respectively for ACC@5, ACC@10, and MRR metrics. As we can see, our proposed method CPNSA performs consistently and significantly better than other baselines on Acc@5, Acc@10, and MRR in all datasets. The results indicate that our proposed method can better predict next activated node. It is interesting to see that SNNSA performs better on (Rd, Exp) and (RD, Ray) datasets. This phenomenon demonstrates that although SNNSA involves network structure information via attention mechanism, the method does not fully consider network structure information, such as community structure.

Table 1. Predictive performance ACC@5 for predictions of next activation node on baselines and our proposed model named by CPNSA.

Method	500, Exp	500, Ray	1000, Exp	1000, Ray	Rd, Exp	Rd, Ray
RNNPP	0.0793	0.1796	0.0756	0.1388	0.0800	0.1337
RMTPP	0.3486	0.3308	0.3443	0.3094	0.6807	0.7144
SNNSA	0.3466	0.2851	0.2835	0.2357	0.7135	0.6566
CPNSA	**0.8268**	**0.7912**	**0.8930**	**0.8903**	**0.7428**	**0.8569**

Table 2. Predictive performance ACC@10 for predictions of next activation node on baselines and our proposed model named by CPNSA.

Method	500, Exp	500, Ray	1000, Exp	1000, Ray	Rd, Exp	Rd, Ray
RNNPP	0.1253	0.2142	0.0937	0.1604	0.0395	0.1389
RMTPP	0.4600	0.4508	0.4533	0.4117	0.7208	0.7925
SNNSA	0.4790	0.4217	0.4090	0.3515	**0.8116**	0.7792
CPNSA	**0.8367**	**0.8144**	**0.8950**	**0.8938**	0.7825	**0.8773**

Table 4 shows the prediction comparisons of next activation timestamp on baselines and our proposed model. The evaluation metric is RMSE. We can see that all the methods perform similarity. The available implementation does not allow SNNSA to compute the next activation time.

Table 3. Predictive performance MRR for predictions of next activation node on baselines and our proposed model named by CPNSA.

Method	500, Exp	500, Ray	1000, Exp	1000, Ray	Rd, Exp	Rd, Ray
RNNPP	0.0768	0.1566	0.0757	0.1215	0.0254	0.1312
RMTPP	0.2303	0.2208	0.2263	0.2063	0.4982	0.4736
SNNSA	0.2339	0.1960	0.1910	0.1579	0.4764	0.4309
CPNSA	**0.8222**	**0.7921**	**0.8958**	**0.8896**	**0.7055**	**0.8359**

Table 4. Predictive performance RMSE for predictions of next activation time on baselines and our proposed model named by CPNSA.

Method	500, Exp	500, Ray	1000, Exp	1000, Ray	Rd, Exp	Rd, Ray
RNNPP	7.0813	0.2359	3.1739	0.1958	12.1832	0.6288
RMTPP	**6.0980**	**0.2344**	**2.7998**	**0.1944**	11.6363	**0.5842**
SNNSA	-	-	-	-	-	-
CPNSA	6.5452	0.2829	3.0073	0.2317	**10.9942**	0.6866

5.2 Real Data Results

Table 5 shows the prediction comparisons of next activated node on Digg dataset. We perform all the algorithms on NVIDIA Tesla 32G GPU server. SNNSA algorithm runs out of memory. As we can see, CPNSA performs consistently best. SNNSA uses the whole adjacency matrix of the network, while our model uses the result of network embedding. As we know, the dimension of network embedding is far less than the adjacency matrix of the network. Thus, the space complexity of SNNSA is much larger than ours.

Table 5. Predictive performance of next activated node on Digg dataset.

Method	ACC@5	ACC@10	MRR
RNNPP	0.0088	0.0107	0.0076
RMTPP	0.0179	0.0298	0.0161
SNNSA	-	-	-
CPNSA	**0.7712**	**0.7644**	**0.7437**

Table 6 shows the prediction comparisons of next activated time on Digg dataset. We use RMSE as the evaluation metric. All the methods perform similarity. The results of SNNSA is not shown because of the not available implementation.

Table 6. Predictive performance of next activated time on Digg dataset.

Method	RNNPP	RMTPP	SNNSA	CPNSA
	30863.83	30862.12	-	30862.53

6 Conclusion

In this paper, we work on the cascade prediction task involving network structure information in the recurrent neural network framework (RNN) via attention mechanism. Different from traditional modeling methods, RNN is a convenient and effective tool for cascade predicting, avoiding strong prior knowledge on diffusion model and being flexible to capture complex dependence in cascades. Besides, recent researches find that network structure, such as community structure, always effects cascade behaviors. Thus we first embedding local and global network structure into nodes' representation vectors and using an attention mechanism in RNN to involve network structure information for capture the network structure effects in cascade.

We evaluate the effectiveness of our proposed model on both synthetic and real datasets. Experimental results demonstrate that our proposed model outperforms state-of-the-art modeling methods at the next activated node prediction task. Additionally, CPNSA performs better than SNNSA on both synthetic and real datasets, implying that our method not just involves neighborhood structure information, our method can also capture community structure effects.

Acknowledgement. This work was supported in part by the National Key Research and Development Plan of China under grant 2018YFC0831005 and the Supreme People's Court 2019 Judicial Research Major Project (ZGFYZDKT201916-01).

References

1. Artzi, Y., Pantel, P., Gamon, M.: Predicting responses to microblog posts. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 602–606. Association for Computational Linguistics (2012)
2. Bakshy, E., Rosenn, I., Marlow, C., Adamic, L.: The role of social networks in information diffusion. In: Proceedings of the 21st International Conference on World Wide Web, pp. 519–528 (2012)
3. Barbieri, N., Bonchi, F., Manco, G.: Topic-aware social influence propagation models. Knowl. Inf. Syst. **37**(3), 555–584 (2013). https://doi.org/10.1007/s10115-013-0646-6
4. Cataldi, M., Di Caro, L., Schifanella, C.: Emerging topic detection on twitter based on temporal and social terms evaluation. In: Proceedings of the Tenth International Workshop on Multimedia Data Mining, p. 4. ACM (2010)
5. Centola, D.: The spread of behavior in an online social network experiment. Science **329**(5996), 1194–1197 (2010)

6. Cheng, S., Shen, H., Huang, J., Zhang, G., Cheng, X.: Staticgreedy: solving the scalability-accuracy dilemma in influence maximization. In: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, pp. 509–518. ACM (2013)
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. ACM (2016)
8. Gomez-Rodriguez, M., Leskovec, J., Schölkopf, B.: Modeling information propagation with survival theory. In: International Conference on Machine Learning, pp. 666–674 (2013)
9. Goyal, A., Bonchi, F., Lakshmanan, L.V.: Learning influence probabilities in social networks. In: Proceedings of the third ACM International Conference on Web Search and Data Mining, pp. 241–250. ACM (2010)
10. Henaff, M., Szlam, A., LeCun, Y.: Recurrent orthogonal networks and long-memory tasks. arXiv preprint arXiv:1602.06662 (2016)
11. Hogg, T., Lerman, K.: Social dynamics of digg. EPJ Data Sci. **1**(1), 5 (2012)
12. Huang, H., Shen, H., Meng, Z., Chang, H., He, H.: Community-based influence maximization for viral marketing. Appl. Intell. **49**(6), 2137–2150 (2019). https://doi.org/10.1007/s10489-018-1387-8
13. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137–146. ACM (2003)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
15. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. Phys. Rev. E **78**(4), 046110 (2008)
16. Leskovec, J., Chakrabarti, D., Kleinberg, J., Faloutsos, C., Ghahramani, Z.: Kronecker graphs: an approach to modeling networks. J. Mach. Learn. Res. **11**, 985–1042 (2010)
17. Li, K., Lv, G., Wang, Z., Liu, Q., Chen, E., Qiao, L.: Understanding the mechanism of social tie in the propagation process of social network with communication channel. Front. Comput. Sci. **13**(6), 1296–1308 (2019). https://doi.org/10.1007/s11704-018-7453-x
18. Liu, C., Wang, W., Sun, Y.: Community structure enhanced cascade prediction. Neurocomputing **359**, 276–284 (2019)
19. Liu, S., Zheng, H., Shen, H., Cheng, X., Liao, X.: Learning concise representations of users' influences through online behaviors. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, pp. 2351–2357 (2017)
20. Luo, Z., Wang, Y., Wu, X.: Predicting retweeting behavior based on autoregressive moving average model. In: Wang, X.S., Cruz, I., Delis, A., Huang, G. (eds.) WISE 2012. LNCS, vol. 7651, pp. 777–782. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35063-4_65
21. Nematzadeh, A., Ferrara, E., Flammini, A., Ahn, Y.Y.: Optimal network modularity for information diffusion. Phys. Rev. Lett. **113**(8), 088701 (2014)
22. Nip, J.Y., Fu, K.W.: Networked framing between source posts and their reposts: an analysis of public opinion on china's microblogs. Inf. Commun. Soc. **19**(8), 1127–1149 (2016)

23. Peng, H.K., Zhu, J., Piao, D., Yan, R., Zhang, Y.: Retweet modeling using conditional random fields. In: 2011 11th IEEE International Conference on Data Mining Workshops, pp. 336–343. IEEE (2011)
24. Qiu, J., Tang, J., Ma, H., Dong, Y., Wang, K., Tang, J.: DeepInf: modeling influence locality in large social networks. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2018) (2018)
25. Stern, D.H., Herbrich, R., Graepel, T.: Matchbox: large scale online bayesian recommendations. In: Proceedings of the 18th International Conference on World Wide Web, pp. 111–120. ACM (2009)
26. Su, Z., Wang, W., Li, L., Stanley, H.E., Braunstein, L.A.: Optimal community structure for social contagions. New J. Phys. **20**(5), 053053 (2018)
27. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web, pp. 1067–1077. International World Wide Web Conferences Steering Committee (2015)
28. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1225–1234. ACM (2016)
29. Wang, Y., Shen, H., Liu, S., Gao, J., Cheng, X.: Cascade dynamics modeling with attention-based recurrent neural network. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence,pp. 2985–2991. AAAI Press (2017)
30. Wang, Z., Chen, C., Li, W.: A sequential neural information diffusion model with structure attention. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 1795–1798. ACM (2018)
31. Weng, L., Menczer, F., Ahn, Y.Y.: Predicting successful memes using network and community structure. In: ICWSM (2014)
32. Wu, J., Du, R., Zheng, Y., Liu, D.: Optimal multi-community network modularity for information diffusion. Int. J. Modern Phys. C **27**(08), 1650092 (2016)
33. Wu, Q., Burges, C.J., Svore, K.M., Gao, J.: Ranking, boosting, and model adaptation. Technical report, Microsoft Research (2008)
34. Xiao, S., Yan, J., Yang, X., Zha, H., Chu, S.M.: Modeling the intensity function of point process via recurrent neural networks. In: AAAI, vol. 17, pp. 1597–1603 (2017)
35. Xu, H., Wu, W., Nemati, S., Zha, H.: Patient flow prediction via discriminative learning of mutually-correcting processes. In: 2017 IEEE 33rd International Conference on Data Engineering (ICDE), pp. 37–38. IEEE (2017)
36. Yang, Z., Guo, J., Cai, K., Tang, J., Li, J., Zhang, L., Su, Z.: Understanding retweeting behaviors in social networks. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 1633–1636. ACM (2010)
37. Ye, S., Wu, F.: Measuring message propagation and social influence on twitter.com. Int. J. Commun. Netw. Distrib. Syst. **11**(1), 59–76 (2013)
38. Zaman, T.R., Herbrich, R., Van Gael, J., Stern, D.: Predicting information spreading in twitter. In: Workshop on computational Social Science and the Wisdom of Crowds, nips, vol. 104, pp. 17599–601. Citeseer (2010)

39. Zhang, J., Liu, B., Tang, J., Chen, T., Li, J.: Social influence locality for modeling retweeting behaviors. In: IJCAI, vol. 13, pp. 2761–2767 (2013)
40. Zhang, Q., Gong, Y., Wu, J., Huang, H., Huang, X.: Retweet prediction with attention-based deep neural network. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 75–84. ACM (2016)
41. Zhou, K., Zha, H., Song, L.: Learning social infectivity in sparse low-rank networks using multi-dimensional Hawkes processes. In: Artificial Intelligence and Statistics, pp. 641–649 (2013)