Cascade Dynamics Modeling with Attention-based Recurrent Neural Network

Yongqing Wang[1,2,*], **Huawei Shen**[1,2,†], **Shenghua Liu**[1,2,†], **Jinhua Gao**[1,2,*] **and Xueqi Cheng**[1,2,†]
[1]CAS Key Laboratory of Network Data Science and Technology,
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
[*]{wangyongqing,gaojinhua}@software.ict.ac.cn, [†]{shenhuawei,liushenghua,cxq}@ict.ac.cn

Abstract

An ability of modeling and predicting the cascades of resharing is crucial to understanding information propagation and to launching campaign of viral marketing. Conventional methods for cascade prediction heavily depend on the hypothesis of diffusion models, e.g., independent cascade model and linear threshold model. Recently, researchers attempt to circumvent the problem of cascade prediction using sequential models (e.g., recurrent neural network, namely RNN) that do not require knowing the underlying diffusion model. Existing sequential models employ a chain structure to capture the memory effect. However, for cascade prediction, each cascade generally corresponds to a diffusion tree, causing cross-dependence in cascade—one sharing behavior could be triggered by its non-immediate predecessor in the memory chain. In this paper, we propose to an attention-based RNN to capture the cross-dependence in cascade. Furthermore, we introduce a *coverage* strategy to combat the misallocation of attention caused by the memoryless of traditional attention mechanism. Extensive experiments on both synthetic and real world datasets demonstrate the proposed models outperform state-of-the-art models at both cascade prediction and inferring diffusion tree.

1 Introduction

The emergence of social media platform has revolutionized the dissemination of information via its great ease in information delivery, accessing and filtering. In social media, online content or a piece of information could reach a large number of people by being shared and reshared among them following their social relationship, and a cascade of resharing is developed during this process. Modeling and predicting such cascade dynamics is fundamental to understanding information propagation [Huang *et al.*, 2012], launching campaign of viral marketing [Cheng *et al.*, 2014], and popularity prediction [Shen *et al.*, 2014].

Existing methods fall into two main paradigms according to whether they require a diffusion model. Conventional methods for cascade prediction generally assume that the underlying diffusion model is known a priori. Typical examples include discrete-time or continuous-time independent cascade model [Cheng *et al.*, 2013; Gomez-Rodriguez *et al.*, 2013], discrete-time or continuous-time linear threshold model [Kempe *et al.*, 2003], and their variants with certain constraint confined by network structure [Gomez-Rodriguez *et al.*, 2010]. The key of these models lies in how to estimate the interpersonal influence or parameters that are used to characterize the influence and susceptibility of individuals [Wang *et al.*, 2015]. The estimation is achieved either by exploiting the structure of social network [Kossinets *et al.*, 2008; Gomez-Rodriguez *et al.*, 2013] or by maximizing the likelihood of observed cascades [Tang *et al.*, 2013]. However, the effectiveness of these methods heavily depends on the hypothesis of the underlying diffusion model, which is hard to specify or verify in practice. Consequently, although these methods gains success at characterizing the diffusion process of information, they are inappropriate for cascade prediction.

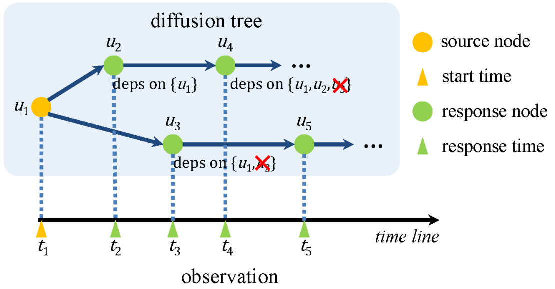


Figure 1: An example of cross-dependence problem in sequence modeling.

Sequential models are proposed to circumvent the problem of cascade prediction, without requiring an explicit underlying diffusion model. Sequential models focus on modeling how the historical sharing behaviors in a cascade affect the future sharing behavior, i.e., the dependence among sharing behaviors in the same cascade. For example, Manavoglue et al. [Manavoglu *et al.*, 2003] proposed a user behavior generation method based on maximum entropy and Markov mixture model. Recently, researchers found that Recurrent Neural Network (RNN) offers a convenient and effective tool for cascade modeling [Mikolov *et al.*, 2010; Wang *et al.*, 2017]. In particular, Du et al. [Du *et al.*, 2016] proposed a RNN framework to model and predict marked cascade, called RMTPP, where timing and mark information are embedded to parameterize the generation process of cascades. The benefits of sequence modeling are two-fold: 1) It avoids strong prior knowledge on diffusion model; 2) It is flexible to capture sequential dependence or memory effect in cascades.

Traditional implementation of RNN (e.g., LSTM and GRU) employ a chain structure to capture the memory effect. However, in the scenario of cascade prediction, each cascade generally corresponds to a diffusion tree, causing cross-dependence in cascade—one sharing behavior could be triggered by its non-immediate predecessor in the memory chain. As shown in Figure **??**, the resharing behavior of u_3 depends on the resharing behavior of u_1 rather that the resharing behavior of its immediate predecessor u_2. This cross-dependence cannot be captured by chain-structured sequential models. For example, for standard RNNs and RNNs with LSTM, u_3 either both depend on u_1 and u_2 or u_3 does not depend on both of them, modulated by specific memory mechanism. Taken together, we lack an effective method to capture the cross-dependence in cascades.

In this paper, we propose an attention-based RNN to capture the cross-dependence in cascade. Furthermore, we introduce a *coverage* strategy [Tu *et al.*, 2016] to combat the misallocation of attention caused by the memoryless of traditional attention mechanism. Our contributions are summarized as:

- We explore the cross-dependence problem existed in RNN when being applied to model cascade dynamics, and we propose an attention-based RNN to capture cross-dependence in cascade;
- We further introduce *coverage* in proprosed attenion-based RNN to adjust allocation of attention, allowing alignments to better reflect the structure of propagation;
- The learned alignments in the proposed models can reflect true diffusion structure and the proposed models consistently outperform competitive baselines in two cascade prediction tasks.

2 Models

The input data is a collection of M cascades $\mathcal{C} = \{S_m\}_{m=1}^{M}$. A cascade $S = \{(t_k, u_k) | u_k \in U, t_k \in [0, +\infty)$ and $k = 1, \ldots, N\}$ is a sequence of resharing behaviors ascendingly ordered by time, where U refers to user set in cascade. The k-th resharing behavior is recorded as a tuple (t_k, u_k), referring to a pair of activation time and activated user. Let the history $\mathcal{H}_k$ be the list of activation time and activated user pairs up to the k-th resharing behavior. The objective of sequence modeling in cascade dynamics is to formulate the conditional probability of next resharing behavior $p((t_{k+1}, u_{k+1})|\mathcal{H}_k)$.

2.1 Background

Firstly, we introduce RNN in cascade dynamics modeling. RNN is a feed-forward neural network, which can be used to generate a cascade by $N-1$ steps sequentially. At step k, we vectorize the k-th resharing behavior into x_k as input. The input is fed into hidden units of RNN by nonlinear transformation f, jointly with the outputs from the last hidden units, updating the hidden state $h_k = \mathit{RNN}(x_k, h_{k-1})$. The representation of hidden state h_k can be considered as embedding of the k-th resharing behavior, and the output is trained to predict the next resharing behavior (t_{k+1}, u_{k+1}) given h_k. In other words, we use RNN to maximize the likelihood of cascade,

$$p(S) = \prod_{k=1}^{N-1} p((t_{k+1}, u_{k+1})|\mathcal{H}_k) = \prod_{k=1}^{N-1} p((t_{k+1}, u_{k+1})|h_k).$$

The conditional probability $p((t_{k+1}, u_{k+1})|h_k)$ can be decomposed into two parts, assuming that the activation time and activated user are conditionally independent with each other,

$$\begin{aligned} p((t_{k+1}, u_{k+1})|h_k) =& p(t_{k+1}|h_k) \cdot p(u_{k+1}|h_k) \\ =& f(t; h_k) \cdot \text{softmax}(g(h_k)), \end{aligned}$$

where g is a non-linear function. The output of softmax function is regarded as the transition proabilities to each possible next activated user. The function $f(t; h_k)$ refers to a temporal point process parameterized by h_k. Based on sufficient observed cascades, RNN can find an optimal solution for the conditional probability of next resharing behavior in a huge functional space, avoiding the bias on diffusion model and the constraint of diffusion network. Thus, RNN offers us a promising and flexible method to capture the complex propagation patterns in cascade dynamics modeling.

2.2 CYAN-RNN

RNN suffers cross-dependence problem caused by tree-structured propagation in cascade, as shown in Fig. **??**. One possible solution is to construct a pooling layer above the hidden units at each step in order to build the direct dependence between the next resharing behavior and all previous resharing behaviors, i.e., $p((t_{k+1}, u_{k+1})|\text{pooling}(h_1, \ldots, h_k))$. The general way of pooling is to calculate a context vector

$$s_k = \sum_{i=1}^{k} \alpha_{k,i} h_i, \quad \text{s.t.} \sum_{i=1}^{k} \alpha_{k,i} = 1, \qquad (1)$$

where the weight $\alpha_{k,i}$ refers to the extent to which the k-th resharing behavior depends on the i-th resharing behavior. Mean pooling and max pooling are two popular choices for setting weights which takes the mean or element-wise max value of all hidden states. However, these two methods still ignore the structure information in cascades. Thus we propose attention mechanism to implicitly model the structure information by automatically learning the pooling weights from the cascade data. Next we introduce the detail of attention mechanism.

Attention Mechanism

Attention mechanism is orginally used in neural machine translation (NMT). In the senario of attention-based NMT, the target words are translated by the words in source sequence and attention mechanism can automatically learn the alignment between source words and target words. For modeling cascade dynamics, we construct both source and target sequence from the observed cascade, restricting that the

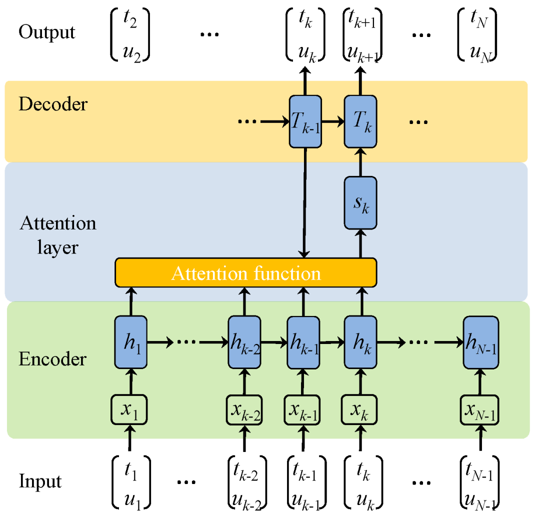


Figure 2: The architecture of CYAN-RNN. The figure presents the case when modeling the generation of the $(k+1)$-th resharing behavior. The sequence at bottom is the observed cascade and the sequence at top is the predictive resharing behaviors. The blue rectangles refer to representations of hidden units. in source sequence, attention layer, and hidden units in target sequence. The yellow rectangle is a general form of attention function $s_k = \mathit{AttentionFunc}(T_{k-1}, \{h_1, \ldots, h_k\})$.

$(k+1)$-th resharing behavior is the output of the k-th resharing behavior.

We propose a dynamic attention mechanism for CYAN-RNN. The proposed architecture is shown in Fig. 2. According to the architecture, we rewrite the conditional probability of next resharing behavior

$$\begin{aligned} p((t_{k+1}, u_{k+1})|\mathcal{H}_k) &= p((t_{k+1}, u_{k+1})|x_k, T_k, s_k) \\ &= f(t; x_k, T_k, s_k) \cdot \text{softmax}(g(x_k, T_k, s_k)). \end{aligned}$$

The time distribution follows

$$\begin{aligned} &f(t; x_k, T_k, s_k) = \lambda(t) \exp\left(-\int_{t_k}^{t} \lambda(\tau) d\tau\right) \\ &\textit{s.t.}\ \ \lambda(t) = \exp(wt + W^{(t)}T_k + U^{(t)}h_k + Z^{(t)}x_k) \end{aligned} \tag{2}$$

where w is a scalar and $W^{(t)}, U^{(t)}, Z^{(t)}$ are parameter matrices. The expectation of Eq. (2) with respect to time t can be regarded as prediction of next activation time. The decoding state T_k for the k-th step in target sequence is computed by

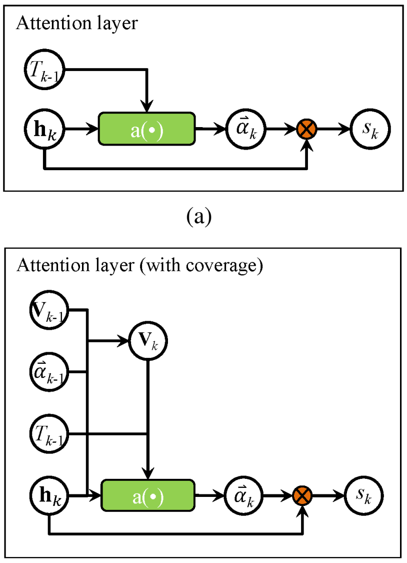
$$T_k = \sigma(x_k, T_{k-1}, s_k), \tag{3}$$

where σ is a non-linear activation function, which can be either a *tanh* or a *sigmoid* function. The context vector s_k is calculated by Eq. (1) where the alignment weights $\alpha_{k,\cdot}$ is updated by the context $\{h_1, \ldots, h_k\}$ and T_{k-1}. The weight $\alpha_{k,i}$ is formalized as

$$\alpha_{k,i} = \frac{\exp(e_{k,i})}{\sum_{j=1}^{k} \exp(e_{k,j})}, \tag{4}$$

where

$$e_{k,i} = a(T_{k-1}, h_i) = v^T \tanh(W^{(a)}T_{k-1} + U^{(a)}h_i) \tag{5}$$



(a)

(b)

Figure 3: Two kinds of implementation on attention layer. (a) The attention mechanism applied in CYAN-RNN; (b) The attention mechanism with coverage applied in CYAN-RNN (cov). Note that $\mathbf{h}_k = (h_1, \ldots, h_k)$ is matrix assembled by all embeddings of historical resharing behaviors at step k, and $\mathbf{V}_k = (V_1, \ldots, V_k)$ is a coverage martix containing all coverage vectors at step k.

scores the extent of the dependence between the i-th resharing behavior and the output at the k-th step, and $W^{(a)}, U^{(a)}$ are the parameter matrices. The implementation of attention mechanism is depicted in Fig. 3(a).

With the attention mechanism, the alignments $\alpha_{k,\cdot}$ can be directly updated through the cost function and exploit an expected representation s_k over all historical resharing behaviors for each step k.

Coverage

In the proposed attention-based model, a majority of "attention" is dominated by a handful of influentials. It indicates that most of users are directly motivated by those influentials in cascade. However, it is generally assumed that users are actually triggered by chronologically adjacent users in cascade [Kossinets *et al.*, 2008]. Therefore we propose coverage to adjust the misallocation of attention, leading the alignments to better reflect the true structure of propagation.

The misallocation of attention is caused by memoryless characteristic in attention mechanism. Inspired by linguistic coverage model, we formulate the general form of coverage in cascade dynamics modeling, keeping historical alignments so as to release misallocation of attention. The k-th step of coverage is defined as

$$V_{k,i} = \sigma\left(V_{k-1,i}, \alpha_{k-1,i}, T_{k-1}, h_i\right). \tag{6}$$

Remarkably, as the increasing context and alignments, $V_{k,k}$ and $\alpha_{k,k}$ have no corresponding values in $V_{k-1,\cdot}$ and $\alpha_{k-1,\cdot}$. Instead we fill up with zeros in our work. At step k, the k-th coverage serves an additional input to the attention mechanism, providing complementary information about how likely the dependence of resharing behaviors has already been explored in the past. The rewritten alignment calculation in Eq. (5) with coverage can be formalized [1] as

$$\begin{aligned} e_{k,i} &= a(T_{k-1}, h_i, V_{k,i}) \\ &= v^T \tanh(W^{(a)} T_{k-1} + U^{(a)} h_i + Z^{(a)} V_{k,i}), \end{aligned} \tag{7}$$

where $W^{(a)}$, $U^{(a)}$ and $Z^{(a)}$ are parameter matrices. We assume that the alignments would focus more on recent resharing behaviors. The assumption will be validated in section 4.

2.3 Length of Dependence

In practice, a cascade may last for a long time and the propagation length would be huge, causing an extreme computation cost when applying dynamic attention mechanism proposed in CYAN-RNN. According to the observation that users' interests concentrated more on recent resharing behaviors, we consider a hyper-parameter, *length of dependence l*, limiting the size of alignments so that the output can only depend on last l resharing behaviors.

3 Optimization

In this section, we introduce the learning process of our proposed models. Given a collection of cascades $\mathcal{C} = \{S_m\}_{m=1}^{M}$, we suppose that each cascade is independent on each other. As a result, the logarithmic likelihood of a set of cascades is the sum of logarithmic likelihood of individual cascade. In this way, the negative logarithmic likelihood of the set of cascades can be estimated as

$$\mathcal{L}(\mathcal{C}) = -\sum_{m=1}^{M} \sum_{k=1}^{N_m - 1} \log p((t_{k+1}, u_{k+1}) | x_k, T_k, s_k), \tag{8}$$

and we can learn parameters of the proposed model by minimizing the negative logarithmic likelihood $\arg\min_\theta \mathcal{L}(\mathcal{C})$, where θ is the parameter set in the model. We exploit back-propagation through time (BPTT) for training. In each training iteration, we vectorize resharing behaviors as inputs, including user embedding and temporal features. The embedding matrix of users is learned along with the training process. The temporal features are assembled by logarithm time interval $\log(t_k - t_{k-1})$ and discretization of numerical attributes on year, month, day, week, hour, mininute and second. We adopt GRU [Chung *et al.*, 2014] to encode the k-th inputs to h_k. We apply stochastic gradient descent (SGD) with mini-batch and the parameters are updated by Adam [Kingma and Adam, 2015]. To speed up the convergence, we use orthogonal initialization method [Henaff *et al.*, 2016] in training process. We also employ early stopping method [Prechelt, 1998] to prevent overfitting.

4 Experiments

In experiments, we compare our CYAN-RNN to the state-of-the-art modeling methods of cascade prediction on both synthetic and real data. The results show that CYAN-RNN performs the best in modeling cascade dynamics. Moreover, we adjust the learned alignments to the underlying social network on synthetic data, demonstrating the capability of our proposed model in network inference.

4.1 Baselines

Previous methods can seldom predict next propagations completely including both next activated users and activation time. To better illustrate the performance of our proposed model, we conduct experiments on two separated tasks, i.e., next activated user prediction task and next activation time prediction task. The choosen baseline models have at least one of the prediction abilities on these two prediction tasks.

- **RMTPP** [Du *et al.*, 2016]: Recurrent marked temporal point process (RMTPP) is a method which can model both the next activated user and the next activation time based on RNN.
- **CT Bernoulli** and **CT Jaccard** models [Goyal *et al.*, 2010]: They are continuous time propagation models. The propagation probabilities between two users are defined by Bernoulli or Jaccard distribution and the probabilities are decayed over time. The two models can be used to predict next activated users.
- **MC-1** Model: The markov chain model is a classic sequence modeling method, depicting the generation of activated users. Here we compare with markov chain with one-order dependency.
- **Poisson process** model [Vere-Jones, 1988]: It is a stochastic point process model, depicting the time consuming from one resharing behavior to another. The intensity function is parameterized by a constant.
- **Hawkes process** model [Hawkes, 1971]: It is a stochastic point process model where the intensity function is parameterized by

  $$\lambda(t) = \lambda_0 + \alpha \sum_{t_i < t} \exp\left(-\frac{t - t_i}{\sigma}\right), \tag{9}$$

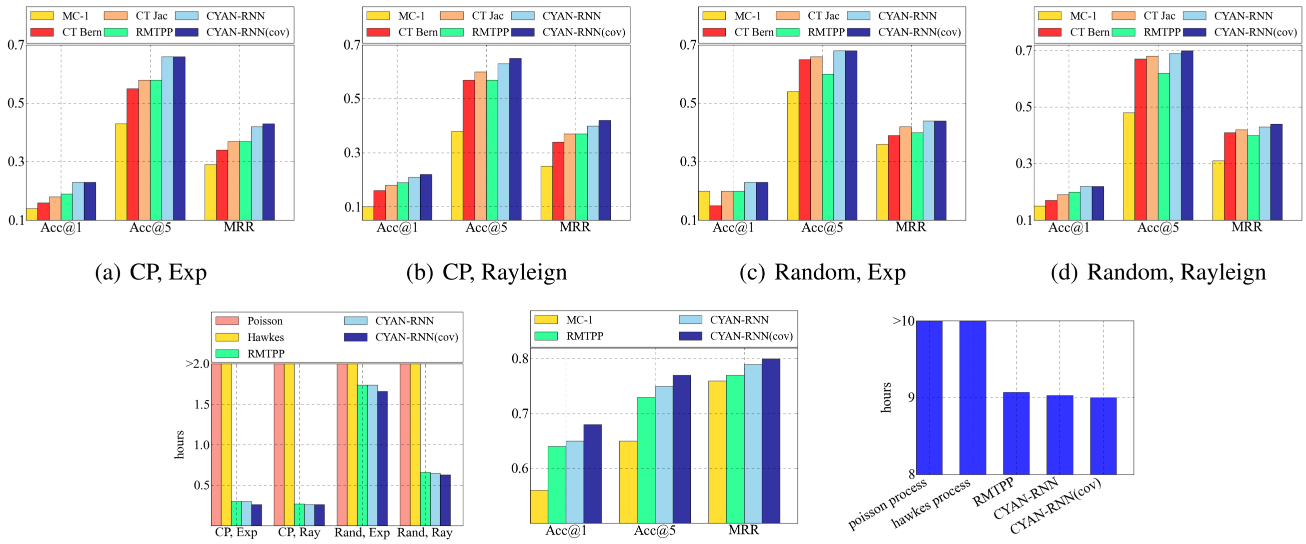
  where λ_0 is the base rate. We set $\sigma = 1$ in our experiments.

4.2 Synthetic Data and Results Anlaysis

The goal of the experiments on synthetic data is to validate the effectiveness of our proposed models in cascade prediction tasks under different underlying network structure and different diffusion models.

Data generation. The data generation consists of two parts: network generation and cascade generation. We use Kroneck generator [Leskovec and Faloutsos, 2007] to construct two types of networks with directed edges: 1) the Core-Periphery (CP) network [Leskovec *et al.*, 2008] (Kroneck parameter matrix $[0.962, 0.535; 0.535, 0.107]$), mimicking real-world social networks; 2) the Erdős-Rényi random (Random) network ($[0.5, 0.5; 0.5, 0.5]$). In terms of cascade, we randomly choose a root user as the source of the cascade and set its activation time as zero. For each activated user, the

[1] If we use the last coverage $V_{k-1,\cdot}$ instead of $V_{k,\cdot}$ to update $e_{k,\cdot}$ at step k, we will lose certain coverage information and cause unbalanced calculation on k-th resharing behavior. This is proved by our preliminary experiments.



(a) CP, Exp

(b) CP, Rayleign

(c) Random, Exp

(d) Random, Rayleign

(e) RMSE on synthetic data

(f) Real data

(g) RMSE on real data

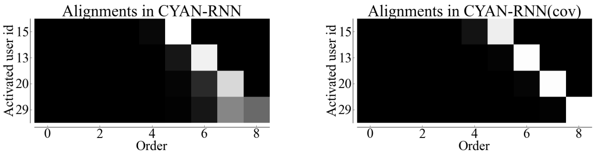
Figure 4: Comparisons on baselines and our proposed models. (a)∼(e) The predictions of next activated user and activation time on synthetic data produced from different networks and diffusion models; (f) and (g) The predictions of next activated user and activation time on real data.

activation time of its neighbors is sampled from a certain time distribution. The generation process is repeated in the breadth-first fashion on the network until the overall time exceeds $\mathcal{T}$ or no user is activated. 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 $[0.01, 10]$. In our experiments, we set the total number of users $|U| = 32$ and the latest time $\mathcal{T} = 100$.

At the end, four datasets are generated by different combinations of network generators and propagation time distributions, denoted by (CP, Exp), (CP, Ray), (Random, Exp) and (Random, Ray). We generate 20,000 cascades in each dataset, and we randomly pick up 80% of cascades for training and the rest for validation and test by an even split.

Evaluation results. We regard the prediction task on next activated user as a ranking problem with users' transition probabilities as their scores. The prediction performance is evaluated by *Accuracy on top k* (Acc@k) and *Mean Reciprocal Rank* (MRR). The larger values in Acc@k and MRR indicate the better performance. In terms of time prediction, we use Root Mean Square Error (RMSE) between the estimated time and the ground truth. A good model should have small values in RMSE.

We first compare the prediction results on next activated user prediction task, shown in Fig. 4(a)∼ 4(d). As we can see, CYAN-RNN and CYAN-RNN(cov) perform consistently and significantly better than other baselines on Acc@1, Acc@5 and MRR in all datasets. The results indicate that our proposed methods can better predict next activated user. It is interesting to see that RMTPP has lower accuracy and MRR values than CT Bern and CT Jac in some cases, while CYAN-RNN and CYAN-RNN(cov) consistently performs the best. It clearly demonstrates that the proposed attention mechanism has the ability to directly capture past propagation information, which may be "forgotten" by sequential transitions in RNN, i.e., cross-dependence problem in cascade dynamics. Fig. 4(e) compares the predictive results on RMSE. We can observe that Poisson and Hawkes processes have the lowest performance, with errors larger than 2 hours. Meanwhile, the RMSE values of RMTPP and CYAN-RNN are equivalently good, while our CYAN-RNN(cov) can perform slightly better than RMTPP and CYAN-RNN. Moreover, we can observe that CYAN-RNN(cov) consistently performs better than CYAN-RNN in the two prediction tasks, implying that the coverage can release the misallocation of attention. Next we will explore how the coverage can help to boost predictive performance and better reflect the structure of propagation.



(a)

(b)

Figure 5: Sampled alignments from a fragement of cascade. The y-axis is the users who will be activated next sequentiallly from top to bottom. The x-axis is the activation order in the cascade. Each pixel shows the alignment $\alpha_{k,i}$ related to the i-th propagation at step k in grayscale (0:black, 1:white). (a) the alignments learned by CYAN-RNN; (b) the alignments learned by CYAN-RNN(cov).

Evaluation on propagation structure. We expect to check if coverage can release the misallocation problem mentioned in section 2.2. Fig. 5(a) and 5(b) show the results. Each row

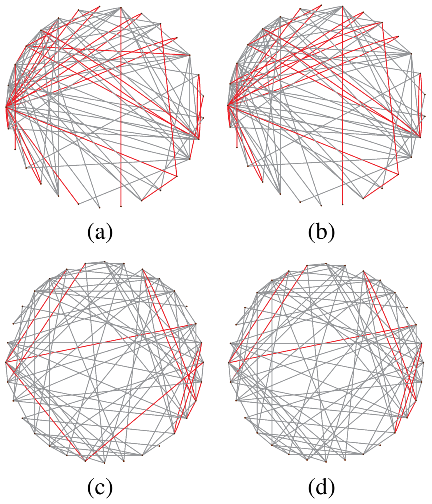


Figure 6: Visualization of network inference. Edges in grey are the correct inferred edges, while edges highlighted in red are either missed or estimated falsely. (a) and (b) CP network inferred by CYAN-RNN and CYAN-RNN(cov) respectively; (c) and (d) Random network inferred by CYAN-RNN and CYAN-RNN(cov) respectively.

in the figure corresponds to the next activated user, with its grids indicating the alignments related to already activated users. The brighter grid refers to the larger alignment. From the figure we can tell which positions in the past propagation are considered more important when predicting the next. Comparing to the alignments in CYAN-RNN, we can see that the alignments in CYAN-RNN(cov) concentrate more on unfollowed resharing behaviors, which helps to release misallocation problem.

Moreover, we wonder if the learned alignments are homologous with true propagation structure. Thus we attempt to infer the underlying network from the alignments. For u_k we take the user with largest alignment as its activation user and mark them as a pair. All the pairs are accumulated in a user-user matrix. We conduct binarization on the matrix through a threshold, thus providing a 0-1 matrix which serves as the adjacency matrix of inferred network. The visualization of network inference is depicted in Fig. 6(a)$\sim$ 6(d). High accuracy of network inference is achieved by CYAN-RNN and CYAN-RNN(cov) in both CP network and Random network. The results indicate that our proposed alignment mechanism can be natrually used in inferring hidden propagation structure, which may have some potential applications in practice, e.g., advertisement and recommendation.

4.3 Real Data and Result Analysis

Experimental setup. The real data is from Sina Weibo, a Chinese microblog website. The data is from June 1st, 2016 to June 30th, 2016. We choose the records on June 1st and extract users whose posting counts are in the range of $(100, 200]$. Then we filtered all the posts by those users in 30 days and extract their cascades. We drop the cascades with size larger than 1,000, as the large cascade rarely occurrs in practice and may dominate the training process. Finally, the processed data contains 2,964 users and 596,088 cascades. We use 536,240 sequences for training, 29,758 for validation and 30,005 for testing.

Prediction results. The results are shown in Fig. 4(f) and 4(g). The hyper-parameters of CYAN-RNN and CYAN-RNN(cov) are set as follows: learning rate is 0.0001; hidden layer size of encoder is 20; hidden layer size of decoder is 10; length of dependence is 200; coverage size is 10; and batch size is 128. We have no social networks in extracted real data, so we cannot compare our proposed models with CT Bern and CT Jac. CYAN-RNN and CYAN-RNN(cov) outperform other baselines with higher MRR values on next activated user prediction task and lower RMSE values on next activation time prediction task. Comparing to RMTPP, CYAN-RNN(cov) achieves 6.25%, 5.48% and 3.90% relative increase on Acc@1, Acc@5 and MRR respectively, and reduces 0.78% relative errors on RMSE. Comparing to CYAN-RNN, CYAN-RNN(cov) achieves 4.62%, 2.67% and 1.27% relative increase on Acc@1, Acc@5 and MRR respectively, and reduces 0.33% relative errors on RMSE.

5 Conclusion

In this paper, we present the cascade dynamics modeling with attention-based RNN. As we know, it is a prior attempt on cascade dynamics modeling based on RNN. Different from traditional modeling methods, RNN is a convenient and effective tool for cascade modeling, avoiding strong prior knowledge on diffusion model and being flexible to capture complex dependence in cascades. However, RNN suffers cross-dependence problem when applying in cascade dynamics modeling. Thus we propose to an attention-based RNN to capture the cross-dependence in cascade. Furthermore, we introduce a coverage strategy to combat the misallocation of attention caused by memoryless of traditional attention mechanism, leading the alignments to better reflect the true structure of propagation.

We evaluate the effectiveness of our proposed models on both synthetic and real datasets. Experimental results demonstrate that our proposed models can consistently outperform state-of-the-art modeling methods at both next activated user prediction task and next activation time prediction task. Addtionallly, CYAN-RNN(cov) performs consistently best on both synthetic and real datasets, implying that the coverage can release the misallocation of attention. Besides, we conduct experiments to explore alignment quality on network inference. The results show that the alignments from our proposed models can reflect true propagation structure, which may have some potential applications in practice, e.g., advertisement and recommendation.

Acknowledgments

This work was funded by the National Basic Research Program of China under grant numbers 2013CB329606 and 2014CB340401, and the National Natural Science Foundation of China under grant numbers 61472400, 61425016, and 61433014. Shenghua Liu is also funded by the Beijing NSF No. 4172059. Huawei Shen is also funded by Youth Innovation Promotion Association CAS.

References

[Cheng *et al.*, 2013] Suqi Cheng, Huawei Shen, Junming Huang, Guoqing Zhang, and Xueqi Cheng. Staticgreedy: Solving the scalability-accuracy dilemma in influence maximization. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, pages 509–518, 2013.

[Cheng *et al.*, 2014] Suqi Cheng, Huawei Shen, Junming Huang, Wei Chen, and Xueqi Cheng. Imrank: Influence maximization via finding self-consistent ranking. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 475–484, 2014.

[Chung *et al.*, 2014] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[Du *et al.*, 2016] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1555–1564, 2016.

[Gomez-Rodriguez *et al.*, 2010] Manuel Gomez-Rodriguez, Jure Leskovec, and Andreas Krause. Inferring networks of diffusion and influence. *Computing Research Repository*, abs/1006.0:1019–1028, 2010.

[Gomez-Rodriguez *et al.*, 2013] Manuel Gomez-Rodriguez, Jure Leskovec, and Bernhard Schlkopf. Modeling information propagation with survival theory. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pages 666–674, 2013.

[Goyal *et al.*, 2010] Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 241–250, 2010.

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

[Henaff *et al.*, 2016] Mikael Henaff, Arthur Szlam, and Yann LeCun. Orthogonal rnns and long-memory tasks. *arXiv preprint arXiv:1602.06662*, 2016.

[Huang *et al.*, 2012] J. Huang, X.Q. Cheng, H.W. Shen, T. Zhou, and X. Jin. Exploring social influence via posterior effect of word-of-mouth recommendations. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 573–582, 2012.

[Kempe *et al.*, 2003] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 137–146, 2003.

[Kingma and Adam, 2015] Diederik P Kingma and Jimmy Ba Adam. Adam: A method for stochastic optimization. In *International Conference on Learning Representation*, 2015.

[Kossinets *et al.*, 2008] Gueorgi Kossinets, Jon M. Kleinberg, and Duncan J. Watts. The structure of information pathways in a social communication network. *Computing Research Repository*, abs/0806.3:435–443, 2008.

[Leskovec and Faloutsos, 2007] Jure Leskovec and Christos Faloutsos. Scalable modeling of real graphs using kronecker multiplication. In *Proceedings of the 24th International Conference on Machine Learning*, pages 497–504, 2007.

[Leskovec *et al.*, 2008] Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th International Conference on World Wide Web*, pages 695–704, 2008.

[Manavoglu *et al.*, 2003] Eren Manavoglu, Dmitry Pavlov, and C. Lee Giles. Probabilistic user behavior models. In *IEEE International Conference on Data Mining*, pages 203–210, 2003.

[Mikolov *et al.*, 2010] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, volume 2, page 3, 2010.

[Prechelt, 1998] Lutz Prechelt. Automatic early stopping using cross validation: quantifying the criteria. *Neural Networks*, 11(4):761–767, 1998.

[Shen *et al.*, 2014] Huawei Shen, Dashun Wang, Chaoming Song, and Albert-Lászlό Barabási. Modeling and predicting popularity dynamics via reinforced poisson processes. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 291–297, 2014.

[Tang *et al.*, 2013] Jie Tang, Sen Wu, and Jimeng Sun. Confluence: Conformity influence in large social networks. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 347–355, 2013.

[Tu *et al.*, 2016] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. In *Proceedings of Association for Computational Linguistics*, 2016.

[Vere-Jones, 1988] D Vere-Jones. An introduction to the theory of point processes. *Springer Ser. Statist., Springer, New York*, 1988.

[Wang *et al.*, 2015] Yongqing Wang, Huawei Shen, Shenghua Liu, and Xueqi Cheng. Learning user-specific latent influence and susceptibility from information cascades. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 477–483, 2015.

[Wang *et al.*, 2017] Yongqing Wang, Shenghua Liu, Huawei Shen, Jinhua Gao, and Xueqi Cheng. Marked temporal dynamics modeling based on recurrent neural network. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2017.