Attention Network for Information Diffusion Prediction

Zhitao Wang, Chengyao Chen and Wenjie Li
Department of Computing, The Hong Kong Polytechnic University, Hong Kong
{csztwang,cscchen,cswjli}@comp.polyu.edu.hk

ABSTRACT

In this paper, we propose an attention network for diffusion prediction problem. The developed diffusion attention module can effectively explore the implicit user-to-user diffusion dependency among information cascade users. Besides, the user-to-cascade importance and the time-decay effect are captured and utilized by the model. The superiority of the proposed model over state-of-the-art methods is demonstrated by experiments on real diffusion data.

CCS CONCEPTS

• **Information systems** → **Data mining**; • **Computing methodologies** → **Neural networks**;

KEYWORDS

Information Diffusion, Attention Network

ACM Reference Format:
Zhitao Wang, Chengyao Chen and Wenjie Li. 2018. Attention Network for Information Diffusion Prediction. In *WWW '18 Companion: The 2018 Web Conference Companion, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3184558.3186931

1 INTRODUCTION

Information diffusion has been studied with a long history. Early researches focus on the expansion of the fundamental theoretical models, such as independent cascade (IC) [3, 4]. Recently, with the development of online media, information cascades of diffusion are massively traced. This provides great opportunities for applied studies, e.g., predicting real diffusion process. Different from theoretical assumption, real cascades are often recorded as sequences, where the underlying graph of user relationship is not available. This drives researchers to solve the problem from a data-driven view. They formulated diffusion prediction as the sequence prediction problem, which is to predict the future affected user given the previously affected users. A family of sequential models were proposed and their effectivenesses were demonstrated on the real diffusion data [5].

However, due to the constraints of the underlying graph, the information cascades in real data do not strictly follow the assumptions of sequential models. For example, given a cascade $\{(A, t_A), (B, t_B), (C, t_C), (D, t_D)\}$ and a underlying graph with edges $A \rightarrow B$, $A \rightarrow C$ and $B \rightarrow D$, the sequential models assume that the infections of C and D are related to the sequential state at t_B and t_C, but in fact, C and D are directly dependent on A and B respectively. Though the gating mechanisms in existing sequential models can selectively drop the information of B when generating C at t_C, it also leads to the loss of dependency from D to B at next time step. The compressed state is not expressive enough for such non-sequential diffusion dependency, thus the prediction ability is limited.

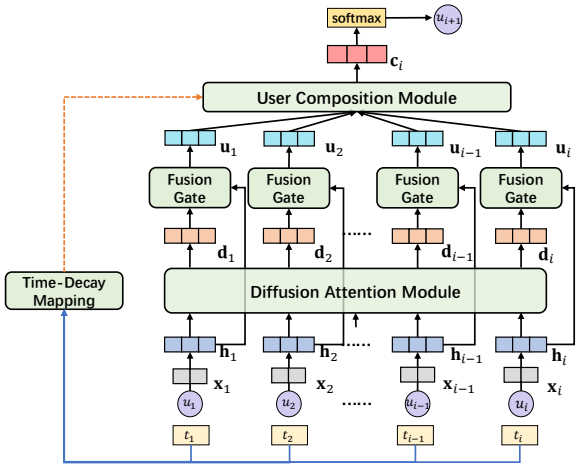

Figure 1: Model Overview

We propose an attention network to explore the diffusion dependency among cascade users. Specifically, we develop a diffusion attention module to capture user dependency and extract dependency-aware information in embedding space. A fusion gate is designed to integrate user self embedding and its dependency-aware information. Given the fused user embedding, the cascade embedding for prediction is generated through user information composition, where both the user-to-cascade importance and the time-decay effect are considered. We evaluate the proposed model on the real diffusion data. The better performance than popular graph-based models and state-of-the-art sequential models indicates that the proposed model effectively capture the underlying diffusion dependency with attention mechanisms.

2 METHOD

A cascade can be represented as $c = \{(u_1, t_1), (u_2, t_2), ..., (u_c, t_c)\}$, where element denotes user u_i is infected at time t_i. The diffusion prediction problem is to predict next user u_{i+1} with given the previously infected users $\{(u_1, t_1), ..., (u_i, t_i)\}$.

The framework of the proposed **D**iffusion **A**ttention **N**etwork (**DAN**) is illustrated in Fig.1. Each user u in cascade is initially embedded into a low-dimensional vector $\mathbf{x} \in \mathbb{R}^{d_x}$ as the input of model. The model then transforms the input as hidden user embeddings with a feed-forward layer: $\mathbf{h} = \text{ELU}(\mathbf{W}_x\mathbf{x} + \mathbf{b}_x)$, where $\mathbf{W}_x \in \mathbb{R}^{d_x \times d_h}$, $\mathbf{h}, \mathbf{b}_x \in \mathbb{R}^{d_h}$ and ELU represents the Exponential

WWW '18 Companion, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5640-4/18/04.
https://doi.org/10.1145/3184558.3186931

Linear Unit non-linear activation function. Given the user embeddings $\mathbf{h}$, a dependency-aware information vector is constructed for each affected user through the Diffusion Attention Module. The module employs user-to-user attention mechanism to extract diffusion dependency between each user and its previously affected users. For cascade user $u_j \in \{u_1, ..., u_i\}$, it defines the attention of u_j to its previous user $u_k \in \{u_1, ..., u_{j-1}\}$ as follow:

$$\alpha_{jk} = \frac{\exp(\langle \mathbf{W}_h^1 \mathbf{h}_j, \mathbf{W}_h^2 \mathbf{h}_k \rangle)}{\sum_{l=1}^{j-1} \exp(\langle \mathbf{W}_h^1 \mathbf{h}_j, \mathbf{W}_h^2 \mathbf{h}_l \rangle)} \quad (1)$$

where $\mathbf{W}_h^1, \mathbf{W}_h^2 \in \mathbb{R}^{d_h \times d_h}$ and $\langle , \rangle$ represents inner product. Based on above attention, the dependency-aware information vector of u_j is computed as: $\mathbf{d}_j = \sum_{k=1}^{j-1} \alpha_{jk} \mathbf{h}_k$.

To integrate user representation $\mathbf{h}_j$ and its dependency-aware information $\mathbf{d}_j$, a fusion gating mechanism is designed as follow:

$$\mathbf{g}_j = \text{sigmoid}(\mathbf{W}_g^1 \mathbf{h}_j + \mathbf{W}_g^2 \mathbf{d}_j + \mathbf{b}_g) \quad (2)$$

$$\mathbf{u}_j = \mathbf{g}_j \odot \mathbf{h}_j + (1 - \mathbf{g}_j) \odot \mathbf{d}_j \quad (3)$$

where $\mathbf{W}_g^1, \mathbf{W}_g^2 \in \mathbb{R}^{d_h \times d_h}$ and $\mathbf{g}, \mathbf{b}_g \in \mathbb{R}^{d_h}$.

Because all previously infected users may trigger the next infection at time t_{i+1}, we compose each extracted user information $\mathbf{u}$ in cascade at t_i through User Composition Module. It considers both the user-to-cascade importance and the time-decay effect. The importance of u_j to cascade is calculated by an attention mechanism as follow:

$$\beta_j = \frac{\exp(\mathbf{w}^T \text{ELU}(\mathbf{W}_u \mathbf{u}_j + \mathbf{b}_u))}{\sum_{k=1}^{i} \exp(\mathbf{w}^T \text{ELU}(\mathbf{W}_u \mathbf{u}_k + \mathbf{b}_u))} \quad (4)$$

where $\mathbf{W}_u \in \mathbb{R}^{d_h \times d_h}$, $\mathbf{w}, \mathbf{b}_u \in \mathbb{R}^{d_h}$. As for capturing time-decay effect on u_j at time t_i, we transform the time interval $t_i - t_j$ to a one hot vector $\mathbf{t}^j \in \mathbb{R}^L$, where $\mathbf{t}_n^j = 1$ if $t_{n-1} < t_i - t_j < t_n$. The critical time points t_{n-1}, t_n are defined by splitting the time range $(0, T]$ into L disjoint intervals $\{(0, t_1], ...(t_{n-1}, t_n], ..., (t_{L-1}, T]$, where T is max observation time in dataset. In the experiments, we set T as 120 hours and the number of intervals L as 40. Given the $\mathbf{t}^j$, the model derives the time-decay weight as: $\lambda_j = \boldsymbol{\lambda} \cdot \mathbf{t}^j$, where $\boldsymbol{\lambda} \in \mathbb{R}^L$ is the parameter vector to be learned. Based on the extracted user-to-cascade importance and time-decay weight, the module finally generates the embedding of cascade at time t_i as: $\mathbf{c}_i = \sum_{j=1}^{i} \beta_j \lambda_j \mathbf{u}_j$.

Given the above embedding $\mathbf{c}_i$, the model can output the probability distribution of next infected user as follow:

$$\hat{P}(u_{i+1} | \mathbf{c}_i) = \text{softmax}(\mathbf{W}_c \mathbf{c}_i + \mathbf{b}_c) \quad (5)$$

where $\mathbf{W}_c \in \mathbb{R}^{d_h \times d_h}$ and $\mathbf{b}_c \in \mathbb{R}^{d_h}$. The model is trained by minimizing the cross-entropy loss between the predicted $\hat{P}$ and true probability of u_{i+1}. The backpropagation algorithm is exploited and parameters are updated by Adadelta optimizer with mini-batch.

3 EXPERIMENTS

The representative real diffusion data, i.e., MemeTracker [2], is used to evaluate the performance of the proposed model. The final size of MemeTracker data for experiment is: 1109 nodes, 33992 training cascades, 4250 validation cascades and 4250 testing cascades.

We compare the proposed model with the following state-of-the-art baselines: Continuous Time Independent Cascade (CTIC) [4] is a representative theoretical model; Embedded Independent Cascade (EIC) [1] is a state-of-the-art representation learning model for diffusion prediction; RNN is the basic recurrent neural network sequential model; LSTM is a stronger RNN-based model which employs an effective gating mechanism; CYAN-RNN [5] is the latest sequence-to-sequence method for cascade prediction. The input embedding and hidden embedding sizes are respectively set as 32 and 64 for all neural models. Other parameters of baselines follow the recommended settings in original papers. For the proposed DAN, the learning rate of Adadelta optimizer is set as 0.5.

Table 1: Diffusion Prediction Performance(%)

	Validation			Testing		
	MRR	*A@5*	*A@10*	*MRR*	*A@5*	*A@10*
CTIC	8.27	10.35	13.18	7.83	9.96	12.54
EIC	6.36	9.07	10.29	5.60	8.23	10.77
RNN	22.87	31.08	39.87	23.26	31.17	40.23
LSTM	24.15	32.96	41.44	24.53	33.01	41.87
CYANRNN	11.20	16.78	22.36	10.63	15.24	21.97
DAN	**26.20**	**36.01**	**45.82**	**26.17**	**35.53**	**45.79**

The performance is evaluated by predicting next infected user. Given previously infected users, models can estimate the next infection probability of each user. Due to the large number of potential targets, this prediction task is often regarded as a ranking problem [5]. We employ two widely used ranking metrics for evaluation: Mean Reciprocal Rank (*MRR*) and Accuracy on top k (*A@k*).

The experimental results are shown in Table 1. The representative RNN-based sequential models achieve relatively better performance among baselines. In spite of employing sequential techniques, CYANRNN achieves relatively worse results, possibly because the sequence-to-sequence architecture in CYANRNN may be not well adaptive to the single-chain structure of cascades. The proposed DAN outperforms all baselines in terms of all metrics. It gains a significant improvement over the best baseline, i.e., LSTM. This demonstrates that, compared with the state-of-the-art sequence gating method, the proposed attention architecture is more aware of the non-sequential diffusion dependency among cascade users thus it has better prediction ability for diffusion cascades.

ACKNOWLEDGMENTS

The work in this paper was supported by Research Grants Council of Hong Kong (PolyU 152094/14E), National Natural Science Foundation of China (61272291) and The Hong Kong Polytechnic University (G-YBJP, 4-BCB5).

REFERENCES

[1] Simon Bourigault, Sylvain Lamprier, and Patrick Gallinari. 2016. Representation learning for information diffusion through social networks: an embedded cascade model. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining (WSDM '16)*. ACM, 573–582.

[2] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. 2009. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '10)*. ACM, 497–506.

[3] Manuel Gomez Rodriguez, David Balduzzi, and Bernhard Schölkopf. 2011. Uncovering the Temporal Dynamics of Diffusion Networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML '11)*. ACM, 561–568.

[4] Kazumi Saito, Masahiro Kimura, Kouzou Ohara, and Hiroshi Motoda. 2009. Learning continuous-time information diffusion model for social behavioral data analysis. In *Asian Conference on Machine Learning*. Springer, 322–337.

[5] Shenghua Liu Jinhua Gao Xueqi Cheng Yongqing Wang, Huawei Shen. 2017. Cascade Dynamics Modeling with Attention-based Recurrent Neural Network. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI '17)*. 2985–2991.