Liu *et al. Hum. Cent. Comput. Inf. Sci.* (2020) 10:13
https://doi.org/10.1186/s13673-020-00218-w

Human-centric Computing and Information Sciences

RESEARCH Open Access

Information cascades prediction with attention neural network

Yun Liu[1*], Zemin Bao[1,2], Zhenjiang Zhang[1], Di Tang[3] and Fei Xiong[1]

*Correspondence:
liuyun@bjtu.edu.cn
[1] Key Laboratory of Communication and Information Systems,Beijing Municipal Commission of Education, Beijing Jiaotong University, Beijing 100044, China
Full list of author information is available at the end of the article

Abstract

Cascade prediction helps us uncover the basic mechanisms that govern collective human behavior in networks, and it also is very important in extensive other applications, such as viral marketing, online advertising, and recommender systems. However, it is not trivial to make predictions due to the myriad factors that influence a user's decision to reshare content. This paper presents a novel method for predicting the increment size of the information cascade based on an end-to-end neural network. Learning the representation of a cascade in an end-to-end manner circumvents the difficulties inherent to blue the design of hand-crafted features. An attention mechanism, which consists of the intra-attention and inter-gate module, was designed to obtain and fuse the temporal and structural information learned from the observed period of the cascade. The experiments were performed on two real-world scenarios, i.e., predicting the size of retweet cascades on Twitter and predicting the citation of papers in AMiner. Extensive results demonstrated that our method outperformed the state-of-the-art cascade prediction methods, including both feature-based and generative approaches.

Keywords: Information diffusion, Deep learning, Attention network, Cascade prediction

Introduction

Online social networks are very popular among people, and they are changing the way people communicate, work, and play, mostly for the better. One of the things that fascinates us most about social network sites is the resharing mechanism that has the potential to spread information to millions of users in a matter of few hours or days. For instance, a user can share the content (e.g., videos on YouTube, tweets on Twitter, and photos on Flickr) with her set of friends, who subsequently can potentially reshare the content, resulting in the development of a cascade of resharing. Such information cascades play a significant role in almost every social network phenomenon, which include, but are not limited to, the diffusion of innovation, persuasion campaigns, and spreading rumors. Information cascade prediction is to infer some key properties of information cascades, such as their sizes and shapes, which indicate the extent to which the information can reach in the social network. This prediction task can be valuable, and it can be applied in an array of areas, such as content recommender systems and monitoring the consensus opinion. However,

Springer Open

© The Author(s) 2020. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

cascade prediction is challenging due to the myriad factors that influence a user's decision to reshare content.

The problem of cascade prediction has been studied extensively [1–3], but most of the studies either depended heavily on the quality of the carefully designed hand-crafted features or made various strong assumptions about the generative processes of the resharing events and oversimplified reality, leading to impaired predictive power. On the other hand, deep learning methods, such as convolutional neural networks (CNNs) [4] and recurrent neural networks (RNNs) [5], have achieved great success in various complicated tasks [6–8], and some studies have used neural networks as a transformer to leverage various informative features for cascade prediction [9]. Nevertheless, these methods ignore the temporal properties for cascade prediction, which are regarded as the valuable information that is needed to improve cascade prediction in traditional works.

In this paper, we propose to predict the information cascade within a neural network framework, by incorporating an attention mechanism using temporal and structural information learned from the observed period of the cascade. Our proposed method consists of three layers. In the first layer, the structure embedding is obtained by representing the cascade graph as a set of random walk paths that carry information about the propagator of the message and the local and global topologies among them. Inspired by the recent successes of the point process model in a cascade dynamic modeling task [10], temporal embedding is a series of hidden representations of reshared events ordered ascendingly by time. The challenge is how to assemble paths or events into the effective representation of each factor. Thus, in the second layer, we designed a novel attention mechanism that contains intra-attention and inter-gate modules. The assembly problem is solved via the intra-attention mechanism with respect to (w.r.t.) the topological structure and the temporal properties. Further, a gate mechanism is proposed to fuse the structure and temporal representation by capturing the importance of the two factors for cascade prediction. Finally, the top layer introduces a multi-layer perceptron (MLP) to output the prediction (increment size of the cascade in our case). We performed extensive experiments on two representative real-world datasets, a Twitter dataset and an AMiner citation network dataset. Our results indicated that our proposed method outperformed state-of-the-art cascade prediction models.

The remainder of this paper is organized as follows. "Related work" section presents a survey of the related work. "Preliminaries" section formulates the cascade prediction problem and introduces the recurrent neural network. "Approach" section presents the details of the proposed model. The experimental results are presented in "Experiments" section, and conclusions and plans for future work are reported in "Conclusions" section.

Related work

We reviewed and presented relevant studies to our work from two aspects, i.e, cascade prediction and attention mechanism.

Cascade prediction

Information cascade prediction has been explored in recent years and is still an open problem. Existing methods for cascade prediction can be categorized into two broad types, i.e., feature-based and model-based approaches.

Feature-based approaches [2, 3, 11–15] make the connection between the prediction and various types of hand-crafted features that are extracted from the information cascade, including the structural features of the social network, content features, temporal features, and user features. To predict the popularity of news articles in Yahoo News, Arapakis et al. [16] used 10 different features that they extracted from the content of the news articles as well as external sources. To predict the popularity of online videos in YouTube and Facebook, Trzcinski et al. [17] utilized both the visual clues and the early popularity patterns of the videos once they were released. Instead of predicting the total volume or level of popularity, Kong et al. [18] focused on the popularity evolution of online contents and consider the dynamic factors that influenced how the popularity evolved. Nevertheless, there is no principled way to design and extract these features, and the accuracy of the predictions is sensitive to the quality of the extracted features.

Model-based approaches [1, 19–22] are devoted to directly characterizing and modeling the formation of an information cascade in the network. These approaches often are optimized to provide intuitive explanations for the prediction due to the interpretable factors that are incorporated in them. Yu et al. [21] proposed a novel NEtworked Weibull Regression model for modeling microbehavioral dynamics that significantly improved the interpretability and generalizability of traditional survival models. Bao et al. [23] modeled the popularity dynamics of the tweet in Twitter using the Hawkes process. They also proposed a method for exploring an adaptive peeking window for each tweet, which can synthesize all of the global dynamic information within the observed period into the predicted peek point. However, using the model-based approach for cascade prediction often is sub-optimal, because strong assumptions often are made about the process of information flow during a diffusion, and they lack the size of the future cascade as a guide.

Inspired by the recent success of deep learning in various complicated tasks, several studies [9, 24] have adopted deep learning methods to leverage various features for cascade prediction, which achieves satisfactory results. Our work is closely related to the above works. While in our work, learning the representation of cascade in an end-to-end manner circumvents the difficulties inherent to the hand-crafted features design step. We also incorporate the temporal properties, which has been ignored in previous work [9].

Attention mechanism

The concept of attention was first introduced in Neuroscience and Computational Neuroscience [25, 26]. For instance, visual attention is the process by which humans focus on specific portion of their visual inputs for computing the adequate responses. Similarly, in training neural networks, the attention mechanism allows models to learn alignments between different parts of the input. Attention mechanism has gained popularity recently in various tasks, such as neural machine translation [27], image caption [28], image/video popularity prediction [24, 29], and question answering [30, 31]. To predict video popularity, Bielski et al. [29] proposed a model with self-attention mechanism to hierarchically attend both video frames and textual modalities. To the best of our knowledge, we are the first to propose the attention mechanism into cascade prediction by fusing temporal and structural information.

Preliminaries

In this section, we first present a formal definition of the cascade prediction problem ("Problem definition" section), and then we briefly describe the recurrent neural network that is used in our proposed method ("Recurrent neural network" section).

Problem definition

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a social network (e.g., Twitter or the academic paper network), where $\mathcal{V}$ is the set of vertices of $\mathcal{G}$, and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges of $\mathcal{G}$. A vertex $u \in \mathcal{V}$ represents a user in the social network and an edge $(u, v) \in \mathcal{E}$ represents that there exists a feedback relationship (e.g., using a like, comment, share, or cite) between user u and user v.

Suppose we have M cascades that start in $\mathcal{G}$ after time t_0. At time t, we denote the i-th cascade as $g_t^i = (\mathcal{V}_t^i, \mathcal{T}_t^i, \mathcal{E}_t^i)$, where $\mathcal{V}_t^i$ is the subset of $\mathcal{V}$ who have taken part in the cascade, $\mathcal{T}_t^i = \{t_1^i, \ldots, t_{|\mathcal{V}_t^i|}^i\}$ represents the time when a user in $\mathcal{V}_t^i$ takes part in the cascade, and $\mathcal{E}_t^i = \mathcal{E} \cap (\mathcal{V}_t^i \times \mathcal{V}_t^i)$ represents the feedback relationships between users in $\mathcal{V}_t^i$.

In this work, we first obtain g_t^i's detailed representation as $\{\boldsymbol{S^i}, \boldsymbol{H^i}\}$, where $\boldsymbol{S^i}$ and $\boldsymbol{H^i}$ correspond to structure representation and temporal representation, respectively. We denote the cascade size of g_t^i as $R_t^i = |\mathcal{V}_t^i|$. Thus, our aim is to predict the incremental size $\Delta R_t^i = |\mathcal{V}_\infty^i| - |\mathcal{V}_t^i|$. In other words, the target is to learn a function f that maps $\{\boldsymbol{S^i}, \boldsymbol{H^i}\}$ to ΔR_t^i, $f : \boldsymbol{S^i}, \boldsymbol{H^i} \rightarrow \Delta R_t^i$.

Note that throughout this paper, we denote vectors by bold lowercase letters and matrices by bold capital Roman letters. In what follows, we will omit the superscript i of related notations for simplicity.

Recurrent neural network

Recurrent neural network (RNN) [5, 32] is a type of deep neural network with cycle and internal memory units that capture sequential information, which is a more general model than the feed-forward network. In practice, RNN has been shown to be a powerful tool for modeling sequences [33]. Long short-term memory (LSTM) [34] and gated recurrent unit (GRU) [35] are recurrent mechanisms that are used extensively. According to Chung et al. [35], GRU has been shown to exhibit better performance with less computation, and it is used as the basic recurrent unit in our proposed approach. The updating formulation of GRU is as follows:

$$\begin{aligned} \boldsymbol{u_i} &= \sigma(\boldsymbol{W_u x_i} + \boldsymbol{U_u h_{i-1}} + \boldsymbol{b_u}) \\ \boldsymbol{r_i} &= \sigma(\boldsymbol{W_r x_i} + \boldsymbol{U_r h_{i-1}} + \boldsymbol{b_r}) \\ \tilde{\boldsymbol{h}}_{\boldsymbol{i}} &= \tanh(\boldsymbol{W_h x_i} + \boldsymbol{r_i U_h h_{i-1}} + \boldsymbol{b_h}) \\ \boldsymbol{h_i} &= \boldsymbol{u_i} \cdot \tilde{\boldsymbol{h}}_{\boldsymbol{i-1}} + (\boldsymbol{1} - \boldsymbol{u_i}) \cdot \boldsymbol{h_{i-1}} \end{aligned} \tag{1}$$

where $\boldsymbol{x_i}$ is current input, $\boldsymbol{h_{i-1}}$ is previous hidden state, $\sigma(\cdot)$ is the sigmoid activation function, $\cdot$ denotess element-wise multiplication, $\boldsymbol{W_u}$, $\boldsymbol{W_r}$, $\boldsymbol{W_h}$, $\boldsymbol{U_u}$, $\boldsymbol{U_r}$, $\boldsymbol{U_h}$,and $\boldsymbol{b_u}$, $\boldsymbol{b_r}$, $\boldsymbol{b_h}$ are GRU parameters learned during training, and $\boldsymbol{h_i}$ is the updated hidden state. The above system can be reduced into an GRU equation: $\boldsymbol{h_i} = GRU(\boldsymbol{x_i}, \boldsymbol{h_{i-1}})$

Approach

In this section, we introduce our proposed method (presented in Fig. 1). It consists of three major components: (1) input embedding ("Input embedding" section); (2) attention mechanism ("Attention mechanism" section); and (3) output layer ("Output layer" section).

Input embedding

Extracting structure representation

The cascade graph g_t is first represented as a set of cascade paths that are sampled through multiple random walk processes. Each of the cascade paths not only carry the information about who are the information propagators, but they also capture the information flow. Thus, we then feed them into a gated recurrent neural network to obtain the hidden representation.

We follow previous work [9, 36] and use a fixed path length L and a fixed number of sequences K. Concisely speaking, for each random walk process, we first sample the starting node with a probability by the following equation:

$$P(u) = \frac{\deg_c(u) + \alpha}{\sum_{s \in V_c}(\deg_c(s) + \alpha)} \tag{2}$$

where α is a smoother, $deg_c(u)$ is the out-degree of vertex u in $\mathcal{G}$, and V_c is the set of nodes in g_t. Following the starting node, the neighbor node is sampled with the probability:

$$P(u \in N_c(v) \mid v) = \frac{\deg_c(u) + \alpha}{\sum_{s \in N_c(v)}(\deg_c(s) + \alpha)} \tag{3}$$

The sampling of one selected sequence stops either when we reach the predefined length L or when we reach a vertex that has no outgoing neighbors. Whenever the length of one sequence is smaller than T, the sequence is padded by a special vertex '+'. This process of sampling sequences continues until we sample K sequences.

Each node in the sequence is represented as a one-hot vector, $\boldsymbol{q} \in \mathbb{R}^N$, where N is the total number of nodes in $\mathcal{G}$. Before we feed the one-hot vector into GRU, we first covert each of them into a low-dimensional dense vector $\boldsymbol{x}$ by a embedding matrix $\boldsymbol{W_x} \in \mathbb{R}^{H \times N}$: $\boldsymbol{x} = \boldsymbol{W_x q}$ where H is an adjustable dimension of embedding.

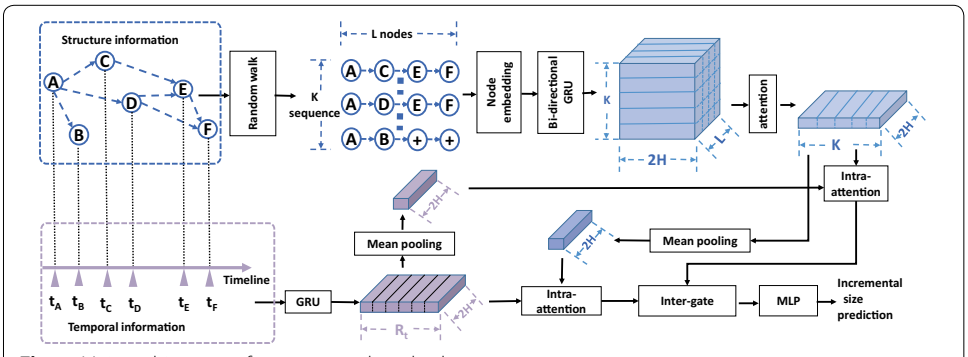

Fig. 1 Main architecture of our proposed method

Then we feed the sequence into GRU to generate sequential hidden states. We adopt the bi-directional GRU [37], where a forward GRU reads the sequence node by node, from left to right, and generates a sequence of forward hidden vectors $[\overrightarrow{\boldsymbol{h}}_{\boldsymbol{i}}^{\boldsymbol{k}}]$. Similarly, a backward GRU reads from right to left, node by node and generates a sequence of backward hidden vectors $[\overleftarrow{\boldsymbol{h}}_{\boldsymbol{i}}^{\boldsymbol{k}}]$. This encoder can be used to simulate the process of information flow during a diffusion. For the i-th node in the sequence, the updated hidden state is computed as the concatenation of the forward and backward hidden vectors:

$$\overleftrightarrow{\boldsymbol{h}}_{\boldsymbol{i}}^{\boldsymbol{k}} = \overrightarrow{\boldsymbol{h}}_{\boldsymbol{i}}^{\boldsymbol{k}} \oplus \overleftarrow{\boldsymbol{h}}_{\boldsymbol{i}}^{\boldsymbol{k}} \tag{4}$$

where $\oplus$ denotes the concatenation operation.

Hence, we can obtain the k-th sequence's representation $[\overleftrightarrow{\boldsymbol{h}}_{\boldsymbol{i}}^{\boldsymbol{k}}]$. We assume multinomial distribution $\alpha_1, \ldots, \alpha_L$ over L nodes so that $\sum_{i=1}^{L}(\alpha_i) = 1$. Thus, the k-th sequence is represented as:

$$\boldsymbol{s_k} = \sum_{i=1}^{L} \alpha_i \overleftrightarrow{\boldsymbol{h}}_{\boldsymbol{i}}^{\boldsymbol{k}} \tag{5}$$

Note that the weight α_i is also learned through the deep learning process.

Finally, from the perspective of topological structure, a cascade graph can be expressed as $\boldsymbol{S} = [\boldsymbol{s_1}, \ldots, \boldsymbol{s_K}]$, $\boldsymbol{s_k} \in \mathbb{R}^{2H}$.

Extracting temporal representation

When we consider about the temporal information of cascade graph g_t, the adoption process is either a time series or a point process. The former series is indexed with fixed and equal time intervals, which can be used to capture the dependence in the time-varying features in a timely manner. The latter are generated asynchronously with random timestamps, and the precise time interval between two adoption events carries a great deal of information about the underlying dynamics. Capturing this information will be crucial for predicting the increment size of the cascade graph. Thus, as Fig. 1 shows, we used the point process form. The effectiveness of the point process form is demonstrated in "Experiments" section.

Specifically, for adoption event i, we can extract the associated temporal features (e.g., the inter-event duration $d_i = t_i - t_{i-1}$) and obtain the corresponding temporal sequence $\mathcal{T}_t = \{d_1, \ldots, d_{|\mathcal{V}_t|}\}$. Then, we feed the sequence, $\mathcal{T}_t$, into GRU, where the hidden state of adoption event i (denoted as $\boldsymbol{h_i}$) can be updated by:

$$\boldsymbol{h_i} = \mathrm{GRU}(d_i, \boldsymbol{h_{i-1}}) \tag{6}$$

We should emphasize that, in this case, the current input vector degenerates into a scalar. After recurrent computation for each time step, we gather a series of hidden states $\boldsymbol{T} = [\boldsymbol{h_1}, \ldots, \boldsymbol{h_{R_t}}]$, $\boldsymbol{h_m} \in \mathbb{R}^{2H}$.

In summary, we have a structure representation $\boldsymbol{S}$ and a temporal representation $\boldsymbol{T}$ as inputs for the attention mechanism to be proposed below.

Attention mechanism

Our attention mechanism consists of two parts: intra-attention module and inter-gate module. Through these we can obtain a more suitable representation of cascade g_t for prediction.

Intra-attention mechanism

Attention computation for topological structure Intra-attention w.r.t. topological structure (presented in Fig. 2) aims at assembling the sampled cascade paths into the effective representation of the structure information of g_t. First, we convert the temporal embedding matrix into a vector representation $\bar{\boldsymbol{h}}$ via a mean pooling mechanism:

$$\bar{\boldsymbol{h}} = \frac{1}{R_t}\sum_{m=1}^{R_t} \boldsymbol{h_m} \tag{7}$$

The weight α_k is formalized as

$$\alpha_k = \frac{exp(\omega(\boldsymbol{s_k}, \bar{\boldsymbol{h}})}{\sum_{k=1}^{K} exp(\omega(\boldsymbol{s_k}, \bar{\boldsymbol{h}}))} \tag{8}$$

where α_k is the attention to the hidden state representation of the k-th sequence in the graph g_t, and $\omega(\boldsymbol{s_k}, \bar{\boldsymbol{h}})$ is set using the following function

$$\omega(\boldsymbol{s_k}, \bar{\boldsymbol{h}}) = \boldsymbol{A_S} \tanh(\boldsymbol{W_S s_k} + \boldsymbol{U_S}\bar{\boldsymbol{h}}) \tag{9}$$

where the parameter matrices of intra-attention satisfy $\boldsymbol{A_S} \in \mathbb{R}^{1\times 2H}$, $\boldsymbol{W_S}$ and $\boldsymbol{U_S} \in \mathbb{R}^{2H\times 2H}$. The above equation essentially is used to calculate the relevance of each sequence in graph g_t to temporal embedding. The intuition lies in the aspect that different temporal properties have diverse influences on the topological structure of the cascade. For instance, when compared with adoption events that occur occasionally, intensive adoption events will bring more potential adoption base for the selected message, which in turn leads to a more complex cascade network. Hence, here we used temporal embedding to guide the combined weights learning of sequences extracted in the cascade graph. Consequently, we can get the attended whole structure embedding $\dot{\boldsymbol{s}}$ via the weighted sum pooling mechanism:

$$\dot{\boldsymbol{s}} = \sum_{k=1}^{K} \alpha_k \cdot \boldsymbol{s_k} \tag{10}$$

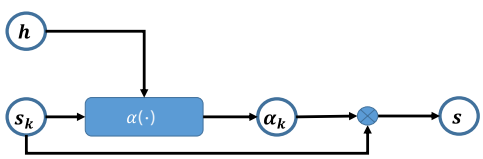

Fig. 2 Architecture of the intra-attention mechanism w.r.t. topological structure

Attention computation for temporal properties Intra-attention w.r.t. temporal properties (presented in Fig. 3) aims to assemble event into the effective representation of the temporal information of g_t. Similarly, we first convert the structure embedding matrix into a vector representation $\bar{\boldsymbol{s}}$ via a mean pooling mechanism:

$$\bar{\boldsymbol{s}} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{s_k} \tag{11}$$

The attention weight α_m for the m-th hidden vector $\boldsymbol{h_m}$ is formalized as:

$$\alpha_m = \frac{\exp(\omega(\boldsymbol{h_m}, \bar{\boldsymbol{s}})}{\sum_{m=1}^{R_t} \exp(\omega(\boldsymbol{h_m}, \bar{\boldsymbol{s}}))} \tag{12}$$

where

$$\omega(\boldsymbol{h_m}, \bar{\boldsymbol{s}}) = \boldsymbol{A_T} \tanh(\boldsymbol{W_T}\boldsymbol{h_m} + \boldsymbol{U_T}\bar{\boldsymbol{s}}) \tag{13}$$

scores the extent of the dependence between the i-th adoption behavior and the structure embedding, and the parameter matrices satisfy $\boldsymbol{A_T} \in \mathbb{R}^{1\times 2H}$, $\boldsymbol{W_T}$ and $\boldsymbol{U_T} \in \mathbb{R}^{2H\times 2H}$. Complex cascade network topology will improve the reception and visibility of the message, and thus promote the occurrence of adoption events. Reflected in the time dimension is the aggregation of adoption events, which is also called bursting diffusion of the message. In our previous work [23], we demonstrated that different parts of the diffusion history have diverse influences on the future cascade size, and we proposed a method for obtaining the most effective part of the history to make an accurate prediction. Analogically, the pooling weights for the temporal property of different adoption events are automatically learned based on the structural embedding of the cascade graph g_t to optimize the prediction of cascade growth.

Hence we can obtain the attended whole temporal embedding $\dot{\boldsymbol{h}}$ via the following equation:

$$\dot{\boldsymbol{h}} = \sum_{m=1}^{R_t} \alpha_m \cdot \boldsymbol{h_m} \tag{14}$$

Inter-gate mechanism

Having obtained the attended whole structure embedding $\dot{\boldsymbol{s}}$ and temporal embedding $\dot{\boldsymbol{h}}$, we can feed these two embeddings into the inter-gate mechanism to effectively combine these two factors. The proposed inter-gate mechanism can capture the different

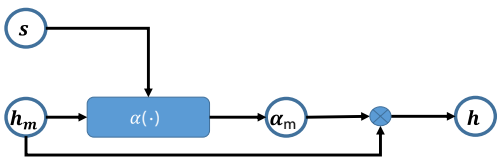

Fig. 3 Architecture of the Intra-attention Mechanism w.r.t. temporal properties

importance of the two factors when predicting the cascade growth. Instead of setting a fixed weight, the proposed inter-gate mechanism can adaptively tune the combination weight. Specifically, the final representation $\boldsymbol{c}$ of cascade graph g_t when combing temporal and structure factor is assembled by:

$$\boldsymbol{c} = \beta \cdot \dot{\boldsymbol{h}} + (1 - \beta) \cdot \dot{\boldsymbol{s}} \tag{15}$$

where the adaptive combination weight $\beta \in (0, 1)$ is computed by:

$$\beta = \sigma(\boldsymbol{W_C}\dot{\boldsymbol{h}} + \boldsymbol{U_C}\dot{\boldsymbol{s}}) \tag{16}$$

where the parameter matrices satisfy $\boldsymbol{W_C}$ and $\boldsymbol{U_C} \in \mathbb{R}^{2H \times 2H}$, and they are both learned through the deep learning process.

Output layer

Finally, our output module consists of a multi-layer perceptron (MLP), taking the cascade representation $\boldsymbol{c}$ as input and generating the final incremental size prediction:

$$\Delta R = MLP(\boldsymbol{c}) \tag{17}$$

The benefit of this fully connected layer is that it does not incur much model complexity and ensures the capacity of nonlinear modeling.

Experiments

This section presents the experiment setup ("Experiment setup" section) and results analysis ("Experiment results" section).

Experiment setup

Dataset and processing

Twitter The dataset contains tweets and retweets on Twitter from September 1 to October 1, 2016. Here we focus on a subset of popular tweets that have at least 50 retweets for easier calibration in our model. For each retweet cascade, the datasets include the publish time of the original tweet, time of retweet, and ID of users who participated in the cascade. The global social network $\mathcal{G}$ was constructed using the same tweet stream from July and August 2016. To evaluate the performance of our model, we split the original data chronologically into a training dataset, a validation dataset and a test dataset. Specifically, cascades whose original tweets were published during the first 11 days were used for training, cascades that originated on September 12 were used for validation, and cascades that originated from September 13 to September 15 were used for testing. The rest of the days were used for unfolding the twitter cascade over the network.

AMiner AMinerThe scientific paper datasets were publicly available in https://www.ami-ner.cn/citation. We constructed the global network $\mathcal{G}$ using the citations between 1985 and 1995. Specifically, we drew an edge from author A to author B if B ever cited A's paper. A citation cascade of a given paper thus contains all authors who have written or cited the paper. We also split the datasets in chronological order. Papers published

between 1996 and 2000 were included in the training set. Papers published in 2001 and 2002 were used for validation and testing, respectively.

In summary, Table 1 gives an overview of the basic statistics of the Twitter dataset and the AMiner dataset.

Evaluation metrics

We used the mean squared error (MSE) and mean absolute errors (MAE), two standard measurements for regression tasks, to evaluate the prediction performance:

$$MSE = \frac{1}{M}\sum_{i=1}^{M}(y_i - \hat{y}_i)^2 \tag{18}$$

$$MAE = \frac{1}{M}\sum_{i=1}^{M}\left|y_i - \hat{y}_i\right| \tag{19}$$

where $\hat{y}_i$ and y_i are the predicted value and ground truth value of cascade i, respectively. Note that, following the practice of [9], we also predict a scaled version of the actual increment of the cascade size, i.e. $y_i = log_2(\Delta R^i + 1)$.

Comparison methods

The comparison methods are as follows:

Features-linear We extract a bag of hand-crafted features that were used in previous work [3, 38–41] and which can better represent the temporal factor and structure factor for cascade prediction. There features are then fed into a linear regression with L2 regularization. These features include:

- *Temporal feature* This type of feature has to do with the speed of adoptions during the prefix cascade. We extract the five point summary (min, median, max, 25-th and 75-th percentile) of waiting times between reshare events, the First Half Rate (mean time between adoptions for the first half of the adoptions), Second Half Rate [38], and the cumulative popularity [42].
- *Structural features* This type of feature includes the structural features of the entire social network around early adopters and the structural features of the cascade. Thus, we extracted the indegree of the each node, connection between g_t and $\mathcal{G}$, number of edges in g_t, number of leaf nodes in g_t, and average and max length of reshare path [38].

Table 1 Statistics of the data set

Data	#node in $\mathcal{G}$	#cascades (train)	#cascades (val)	#cascades (test)	Avg. cascade size
Twitter	429,347	23,786	2604	6275	182.3
AMiner	126,422	31,257	6139	6071	19.1

Support vector regression (SVR) We follow previous work [17, 43] and adopt SVR model using linear kernel to predict cascade size with time series data as features.

SEISMIC [44] This is one of state-of-the-art generative models on cascade prediction. The model is based on a self-exciting point process producing final cascade size forecasts using the early adoption activity of a selected message. Note that its predictor is based on a branching process, and thus this method can only be applied to predict the final size of the retweet cascade. In contrast, our proposed end-to-end method can be easily extended to predict the dynamic of the retweet cascade.

DeepCas [9] This is the first end-to-end, deep learning method for information cascades prediction. It mainly utilizes the information of the structure of the cascade graph and node identities for prediction. The attention mechanism is designed to assemble a cascade graph representation from a set of random walk paths.

Platform and parameter setting

For the length t of the observed initial period of the information cascade, we consider three settings, i.e., $t = 1, 2, 3$ hours for Twitter and $t = 1, 2, 3$ months for AMiner. To instantiate our models, we used the high-level neural network library Keras [45] with Theano [46] as the computational back-end. The code is running on a Linux server with 32G memory, 2 CPUs with 4 cores for each: Inter® $Core^{TM}$ i7-7700K CPU @4.50 GHz. The GPU in use is the $Nvidia^{TM}$ GeForce GTX TITAN 1080 Ti.

Experiment results

We evaluated our proposed model with the comparison methods on the Twitter and AMiner dataset to present the performance of our method. The prediction results are reported in Table 2, which shows that irrespective of the dataset (Twitter and AMiner) and prefix cascade (1, 2, 3 h for Twitter, and 1, 2, 3 months for AMiner), our proposed method outperformed other comparison methods, since it achieved a lower MSE.

Table 2 shows that Features-linear provides worse results than our proposed method, which indicates the limitation of hand-crafted features. The Features-linear method selects the most predictive features for cascade prediction, which was demonstrated in past studies [38]. This is especially obvious when compared with our proposed method, which automatically learns joint and effective representation from temporal and structural factors.

Table 2 also shows that our proposed method outperformed SEISMIC, a state-of-the-art generative model, since our method uses more powerful attention mechanisms and is likely to yield better performance. Specifically, our model uses an attention mechanism to automatically learn the pooling weights for the temporal properties of different adoption events, while SEISMIC uses a constant peeking period within a prefix cascade for different messages when making predictions. In addition, SEISMIC lacks the future cascade size as a guide and makes various stronger assumptions about the diffusion process, which are common disadvantages of generative prediction methods.

Table 2 Overall prediction performance

	t = 1*		t = 2*		t = 3*	
	MSE	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
Twitter						
Features-linear	3.821	1.536	3.511	1.479	3.423	1.42
SVR	3.798	1.529	3.028	1.384	3.382	1.411
SEISMIC	3.770	1.527	2.954	1.313 1.462	3.319	1.408
DeepCas	3.725	1.493	3.496		3.308	1.395
Proposed	2.609	1.265	2.349	1.167	2.300	1.14
AMiner						
Features-linear	2.429	1.197	2.136	1.089	1.880	1.067
SVR	2.419	1.194	2.195	1.123	1.865	1.066
SEISMIC	2.417	1.193	2.282	1.136	1.852	1.061
DeepCas	2.239	1.127	1.987	1.072	1.674	1.056
Proposed	2.172	1.116	1.672	1.042	1.534	1.003

p.s. t = 1*, where '*' denotes hour for Twitter (year for AMiner)

Among all of the methods that were compared, DeepCas had the best performance because it benefits from end-to-end learning from the data to the prediction. Our proposed method leads to a certain reduction of prediction errors when compared with DeepCas, due to the introduction of temporal information, which is ignored in DeepCas.

Comparing the performance of using different prefix t, we can make the conclusion that applies to all methods for both twitter cascade and citation cascade: As we increased the observation time, the prediction errors tended to decrease, suggesting that more accessible information will make prediction easier. In addition, we can observe that prediction errors are much bigger in Twitter (the top-half of the Table 2) than that in AMiner (the bottom-half of Table 2), which indicates that predicting the twitter cascade size is a more difficult scenario of information cascade prediction.

To study the effects of temporal factor and structural factor on cascade prediction in more detail, we compared the proposed method and the Feature-linear method and their variants that do not consider one of these factors. We also ran these methods on the two datasets and aimed to predict the incremental size of information cascade using a fixed observation window ranging from 1 to 3 h (months for AMiner). For ease of results presentation, we denote temporal factor as $\boldsymbol{T}$ and structural factor as $\boldsymbol{S}$, respectively. Thus "no $\boldsymbol{T}$" means removing temporal factor for corresponding methods, and it is similar for "no $\boldsymbol{S}$".

The prediction results of these methods are summarized in Table 3. This results show that our proposed method and Feature-linear both outperform their variants, which indicates the usefulness of these factors. For instance, by testing "Proposed (no $\boldsymbol{T}$)", we can see a notable decrease in performance compared with our proposed method, with MSE = 3.772 and 2.609 when observing for 1 h on Twitter. This phenomenon shows that feeding temporal features into deep neural networks is indeed meaningful.

We also found that Feature-linear (no $\boldsymbol{S}$) performs better than Feature-linear (no $\boldsymbol{T}$), which is consistent with previous research [38]. However, "Proposed (no $\boldsymbol{S}$)" and "Proposed (no $\boldsymbol{T}$)" have very similar performances for most situations, which suggests that

Table 3 Effects of temporal factor and structural factor on cascade prediction

	MSE (t = 1*)	MSE (t = 2*)	MSE (t = 3*)
Twitter			
Features-linear (no $\boldsymbol{T}$)	4.106	3.823	3.715
Features-linear (no $\boldsymbol{S}$)	3.976	3.640	3.524
Features-linear	3.821	3.511	3.423
Proposed (no $\boldsymbol{T}$)	3.772	3.503	3.328
Proposed (no $\boldsymbol{S}$)	3.716	3.540	3.407
Proposed (time series $\boldsymbol{T}$)	3.809	3.621	3.463
Proposed	2.609	2.349	2.300
AMiner			
Features-linear (no $\boldsymbol{T}$)	2.621	2.407	2.092
Features-linear (no $\boldsymbol{S}$)	2.561	2.312	1.986
Features-linear	2.429	2.136	1.880
Proposed (no $\boldsymbol{T}$)	2.411	2.050	1.799
Proposed (no $\boldsymbol{S}$)	2.307	2.186	1.838
Proposed (time series $\boldsymbol{T}$)	2.457	2.129	1.906
Proposed	2.172	1.672	1.534

p.s. t = 1*, where '*' denotes hour for Twitter (year for AMiner)

there potentially is still room to improve the utilization of temporal factors (the most predictive information) in our proposed method. Thus, we examined the effects of different ways to integrate temporal information. The method of "Proposed (time series $\boldsymbol{T}$)" is to form a time series of the cascade size for each message and to feed the time series into our neural network, instead of temporal embedding of individual nodes. Table 3 shows that "Proposed (time series $\boldsymbol{T}$)" performs worse than "Proposed (no $\boldsymbol{S}$)". This is consistent with our expectation, since the precise time interval between two adoption events is more informative than a time series dataset. Note that when making predictions at the beginning of the information cascade, "Proposed (no $\boldsymbol{T}$)" performed worse than "Proposed (no $\boldsymbol{S}$)", which may be due to the fact that a "simple" topology is inadequate for providing an effective forecast. Finally, our proposed method had the best performance, suggesting that temporal information and structural information are complimentary for cascade prediction.

To demonstrate the effectiveness of the components of attention mechanism and gate mechanism in the proposed method, we compare the proposed method and its variants that remove one of the components. For ease of results presentation, we denote attention mechanism as ***attention*** and gate mechanism as ***gate***, respectively. The corresponding results are presented in Table 4. We find that our proposed method outperforms its variants, which demonstrates the positive contribution of each component.

Conclusions

In this paper, we proposed a novel method for information cascade prediction based on an end-to-end neural network. Learning the representation of a cascade in an end-to-end manner circumvented the difficulties inherent to hand-crafted features design. To efficiently obtain and fuse the temporal and structural information, we carefully designed an attention mechanism, which involves intra-attention and inter-gate

Table 4 Contribution of different components of our proposed method

	MSE (t = 1*)	MSE (t = 2*)	MSE (t = 3*)
Twitter			
Proposed (no ***gate***)	2.956	2.811	2.513
Proposed (no ***attention***)	3.226	3.124	2.825
Proposed (no***attention*** + ***gate***)	3.726	3.419	3.376
Proposed	2.609	2.349	2.300
AMiner			
Proposed (no ***gate***)	2.285	1.706	1.592
Proposed (no ***attention***)	2.339	1.816	1.763
Proposed (no ***attention*** + ***gate***)	2.462	1.928	1.809
Proposed	2.172	1.672	1.534

p.s. t = 1*, where '*' denotes hour for Twitter (year for AMiner)

modules. We conducted experiments on two scenarios, i.e., predicting the size of cascade of Tweet on Twitter and predicting the citation of papers in AMiner. Compared with the other three state-of-the-art prediction methods, our proposed method offered small prediction error. Future works include the incorporation of other predictive information within the attention framework. Cascade dynamics modeling with our attention neural network is also of interest.

Abbreviations

CNN: Convolutional neural network; RNN: Recurrent neural network; MLP: Multi-layer perceptron; LSTM: Long short-term memory; GRU: Gated recurrent unit; SVR: Support vector regression; MSE: Mean squared error; MAE: Mean absolute errors.

Acknowledgements

Not applicable.

Authors' contributions

YL carried out design of the proposed framework, managed and supervised this paper. ZB conducted the experiments, analyzed the results and drafted the document. ZZ and DT provided valuable suggestions on improving the standards of the manuscript. All authors read and approved the final manuscript.

Funding

This research was funded by the National Key Research and Development Program of China (Grant No. 2018YFC0832304), the National Science Foundation for Young Scientists of China (Grant No. 61801125) and the Fundamental Research Funds for the Central Universities (Grant No. 2017JBZ107) .

Availability of data and materials

The datasets used in this study are available from the corresponding author on reasonable request.

Competing interests

The authors declare that they have no competing interests.

Author details

[1] Key Laboratory of Communication and Information Systems,Beijing Municipal Commission of Education, Beijing Jiaotong University, Beijing 100044, China. [2] National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China. [3] The Third Research Institute of Ministry of Public Security, Shanghai 200031, China.

Received: 20 April 2019 Accepted: 26 March 2020
Published online: 11 April 2020

References

1. Zaman T, Fox EB, Bradlow ET (2014) A bayesian approach for predicting the popularity of tweets. Ann Appl Stat 8(3):1583–1611
2. Cheng J, Adamic LA, Dow PA, Kleinberg JM, Leskovec J (2014) Can cascades be predicted. In: International world wide web conferences. 925–936

3. Martin T, Hofman JM, Sharma A, Anderson A, Watts DJ (2016) Exploring limits to prediction in complex social systems. In: International conference on world wide web pp 683–694
4. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
5. Pascanu R, Mikolov T, Bengio Y (2013) On the difficulty of training recurrent neural networks. In: International conference on machine learning. p. 1310–1318.
6. Pandarinath C, O'Shea DJ, Collins J, Jozefowicz R, Stavisky SD, Kao JC, Trautmann EM, Kaufman MT, Ryu SI, Hochberg LR et al (2018) Inferring single-trial neural population dynamics using sequential auto-encoders. Nat Methods 15(10):805–815
7. Cornia M, Baraldi L, Serra G, Cucchiara R (2018) Predicting human eye fixations via an lstm-based saliency attentive model. IEEE Trans Image Process 27(10):5142–5154
8. Afouras T, Chung JS, Senior A, Vinyals O, Zisserman A (2018) Deep audio-visual speech recognition. In: IEEE transactions on pattern analysis and machine intelligence
9. Li C, Ma J, Guo X, Mei Q (2017) Deepcas: An end-to-end predictor of information cascades. In: Proceedings of the 26th international conference on world wide web. pp 577–586 . International World Wide Web Conferences Steering Committee
10. Du N, Dai H, Trivedi R, Upadhyay U, Gomez-Rodriguez M, Song L (2016) 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, New York
11. Aghababaei S, Makrehchi M (2017) Activity-based twitter sampling for content-based and user-centric prediction models. Hum Cent Compu Inf Sci 7(1):3
12. Weng L, Menczer F, Ahn Y-Y (2014) Predicting successful memes using network and community structure. In: ICWSM
13. Loyola-González O, López-Cuevas A, Medina-Pérez MA, Camiña B, Ramírez-Márquez JE, Monroy R (2019) Fusing pattern discovery and visual analytics approaches in tweet propagation. Inf Fusion 46:91–101
14. Jia AL, Shen S, Li D, Chen S (2018) Predicting the implicit and the explicit video popularity in a user generated content site with enhanced social features. Comput Netw 140:112–125
15. Kursuncu U, Gaur M, Lokala U, Thirunarayan K, Sheth A, Arpinar IB (2019) Predictive analysis on twitter: techniques and applications. In: Emerging research challenges and opportunities in computational social network analysis and mining. pp 67–104. Springer, Berlin
16. Arapakis I, Cambazoglu BB, Lalmas M (2017) On the feasibility of predicting popular news at cold start. J Assoc Inf Sci Technol 68(5):1149–1164
17. Trzcinski T, Rokita P (2017) Predicting popularity of online videos using support vector regression. IEEE Trans Multimed 99:1–1
18. Kong Q, Mao W, Chen G, Zeng D (2018) Exploring trends and patterns of popularity stage evolution in social media. IEEE Trans Syst Man Cybern Syst 99:1–11
19. Engelhard M, Xu H, Carin L, Oliver JA, Hallyburton M, McClernon FJ (2018) Predicting smoking events with a time-varying semi-parametric hawkes process model. Proc Mach Learn Res 85:312
20. Li L, Zha H (2014) Learning parametric models for social infectivity in multi-dimensional hawkes processes. In: Twenty-eighth AAAI conference on artificial intelligence. p. 101–107
21. Yu L, Cui P, Wang F, Song C, Yang S (2017) Uncovering and predicting the dynamic process of information cascades with survival model. Knowl Inf syst 50(2):633–659
22. Saito K, Nakano R, Kimura M (2008) Prediction of information diffusion probabilities for independent cascade model. In: International conference on knowledge-based and intelligent information and engineering systems. pp 67–75. Springer, Berlin
23. Bao Z, Liu Y, Zhang Z, Liu H, Cheng J (2019) Predicting popularity via a generative model with adaptive peeking window. Phys A Stat Mech Appl 522:54–68
24. Zhang W, Wang W, Wang J, Zha H (2018) User-guided hierarchical attention network for multi-modal social image popularity prediction. In: Proceedings of the 2018 world wide web conference on world wide web. pp. 1277–1286 . International World Wide Web Conferences Steering Committee
25. Itti L, Koch C, Niebur E (1998) A model of saliency-based visual attention for rapid scene analysis. IEEE Trans Pattern Anal Mach Intell 20(11):1254–1259
26. Desimone R, Duncan J (1995) Neural mechanisms of selective visual attention. Ann Rev Neurosci 18(1):193–222
27. Choi H, Cho K, Bengio Y (2018) Fine-grained attention mechanism for neural machine translation. Neurocomputing 284:171–176
28. Lopez PR, Dorta DV, Preixens GC, Sitjes JMG, Marva FXR, Gonzalez J (2019) Pay attention to the activations: a modular attention mechanism for fine-grained image recognition. IEEE Trans Multimed
29. Bielski A, Trzcinski TP (2018) Understanding multimodal popularity prediction of social media videos with self-attention. IEEE Access 6:74277–74287
30. Xiong C, Merity S, Socher R (2016) Dynamic memory networks for visual and textual question answering. In: International conference on machine learning. p. 2397–2406
31. Kumar A, Irsoy O, Ondruska P, Iyyer M, Bradbury J, Gulrajani I, Zhong V, Paulus R, Socher R (2016) Ask me anything: dynamic memory networks for natural language processing. In: International conference on machine learning. p. 1378–1387
32. Elman JL (1990) Finding structure in time. Cognit Sci 14(2):179–211
33. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. In: Advances in neural information processing systems, pp 3104–3112
34. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
35. Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555

36. Perozzi B, Al-Rfou R, Skiena S (2014) Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International conference on knowledge discovery and data mining. pp 701–710. ACM, New York
37. Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. IEEE Trans Signal Process 45(11):2673–2681
38. Shulman B, Sharma A, Cosley D (2016) Predictability of popularity: gaps between prediction and understanding. In: International conference on weblogs and social media. pp 348–357
39. Ugander J, Backstrom L, Marlow C, Kleinberg J (2012) Structural diversity in social contagion. Proceedings of the national academy of sciences 201116502
40. Mishra S, Rizoiu MA, Xie L (2016) Feature driven and point process approaches for popularity prediction. In: ACM international on conference on information and knowledge management, pp 1069–1078
41. Souri A, Hosseinpour S, Rahmani AM (2018) Personality classification based on profiles of social networks' users and the five-factor model of personality. Hum cent Comput Inf Sci 8(1):24
42. Szabo G, Huberman BA (2010) Predicting the popularity of online content. Commun ACM 53(8):80–88
43. Khosla A, Das Sarma A, Hamid R (2014) What makes an image popular? In: Proceedings of the 23rd international conference on world wide web, pp 867–876
44. Zhao Q, Erdogdu MA, He HY, Rajaraman A, Leskovec J (2015) Seismic: a self-exciting point process model for predicting tweet popularity. In: ACM SIGKDD international conference on knowledge discovery and data mining, pp 1513–1522
45. Chollet F et al (2015) Keras: deep learning library for theano and tensorflow. https://keras.io/k. 7(8)
46. Team TTD, Al-Rfou R, Alain G, Almahairi A, Angermueller C, Bahdanau D, Ballas N, Bastien F, Bayer J, Belikov A, et al (2016) Theano: a python framework for fast computation of mathematical expressions. arXiv preprint arXiv :1605.02688

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen journal and benefit from:

- ► Convenient online submission
- ► Rigorous peer review
- ► Open access: articles freely available online
- ► High visibility within the field
- ► Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com