DeepHawkes: Bridging the Gap between Prediction and Understanding of Information Cascades

Qi Cao, Huawei Shen, Keting Cen, Wentao Ouyang, Xueqi Cheng
{caoqi,cenketing}@software.ict.ac.cn,{shenhuawei,ouyangwt,cxq}@ict.ac.cn
CAS Key Laboratory of Network Data Science and Technology, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China

ABSTRACT

Online social media remarkably facilitates the production and delivery of information, intensifying the competition among vast information for users' attention and highlighting the importance of predicting the popularity of information. Existing approaches for popularity prediction fall into two paradigms: feature-based approaches and generative approaches. Feature-based approaches extract various features (e.g., user, content, structural, and temporal features), and predict the future popularity of information by training a regression/classification model. Their predictive performance heavily depends on the quality of hand-crafted features. In contrast, generative approaches devote to characterizing and modeling the process that a piece of information accrues attentions, offering us high ease to understand the underlying mechanisms governing the popularity dynamics of information cascades. But they have less desirable predictive power since they are not optimized for popularity prediction. In this paper, we propose DeepHawkes to combat the defects of existing methods, leveraging end-to-end deep learning to make an analogy to interpretable factors of Hawkes process — a widely-used generative process to model information cascade. DeepHawkes inherits the high interpretability of Hawkes process and possesses the high predictive power of deep learning methods, bridging the gap between prediction and understanding of information cascades. We verify the effectiveness of DeepHawkes by applying it to predict retweet cascades of Sina Weibo and citation cascades of a longitudinal citation dataset. Experimental results demonstrate that DeepHawkes outperforms both feature-based and generative approaches.

KEYWORDS

popularity prediction, information cascade, Hawkes process, end-to-end deep learning, interpretable factors

1 INTRODUCTION

The emergence of online social platforms, e.g., Twitter, Facebook, WeChat and Sina Weibo, has brought unprecedented convenience for the production and delivery of information. Every day, tens of millions of messages are generated on these platforms. Such a large amount of messages not only cause serious information overload problems to users of these platforms, but also bring great difficulties to traffic management and trending topic tracking for the owners of these platforms. Being able to predict the future popularity of online content and to detect popular messages is useful for both users and the owners of these platforms. However, since these social platforms are generally large-scale open systems and may be affected by exogenous factors, it's challenging to accurately make **popularity prediction** [1] of theses messages with complex popularity dynamics.

Existing methods for popularity prediction mainly fall into two categories: feature-based approaches and generative approaches. Feature-based approaches [2–7] extract various types of features, including temporal features, structural features, content features and features defined on early adopters, and generally work in a supervised framework with machine learning algorithms. The performance of this type of approaches heavily depends on the extracted features, which are often hand-crafted based on human's prior domain knowledge and may be specific to particular platform or particular type of information being diffused. This poses a series of challenges for feature-based methods: How can we systematically design those sophisticated features? How do we know whether the extracted features sufficiently capture the relevant information? As a remedy, deep learning methods were recently introduced to automatically learn features for popularity prediction [8]. However, these learned features lack clear interpretability and pose difficulty to understand what make a piece of information popular. In contrast, generative approaches [9–11] devote to characterizing and modeling the process that a piece of information accrues attentions, offering us high ease to understand the underlying mechanisms governing the popularity dynamics of information. Among them, models based on Hawkes self-exciting point process [11, 12] are state-of-the-art, which can well characterize several important phenomena in social networks, i.e., influence of users, self-exciting mechanism of each retweet and the time decay effect in information diffusion. Due to these interpretable factors, such methods can make a pretty meaningful prediction. However, generative approaches has less desirable predictive power since they are not optimized for popularity prediction.

In this paper, we propose DeepHawkes to combat the defects of existing methods, leveraging the end-to-end deep learning framework to make an analogy to the interpretable factors of Hawkes process. Specifically, we embed user identity into a low-dimensional space to represent the interest-aware influence of users, instead of some hand-crafted quantities used in previous work, like the number of fans [11, 12]. As for the self-exciting mechanism of each

CIKM'17, November 6–10, 2017, Singapore.
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-4918-5/17/11…$15.00
http://dx.doi.org/10.1145/3132847.3132973

retweet, we explicitly exploit the influence of the entire retweet path instead of only the retweet user itself in Hawkes process, and use a sum pooling to model the overall contribution of each retweet. This offers us flexibility to separately model the effect of users' identity and their structural positions in cascade, reflected by their occurrence frequency in cascade paths. Finally, we propose to use a non-parametric way to learn a temporal kernel that characterizes the time decay effect in information diffusion, increasing the flexibility to capture complex popularity dynamics. It is worth noting that all these three interpretable factors of Hawkes process are now learned by the guide of future popularity. On the whole, DeepHawkes not only inherits the high interpretability of Hawkes process, but also possesses the high predictive power of deep learning methods, bridging the gap between prediction and understanding of information cascades. We verify the effectiveness of DeepHawkes by applying it to predict retweet cascades of Sina Weibo and citation cascades of a longitudinal citation dataset. Experimental results demonstrate that DeepHawkes not only outperforms both feature-based and generative approaches, but also outperforms state-of-the-art deep learning methods for popularity prediction.

The main contributions of this paper are two-fold:

- **Bridging the gap between prediction and understanding of information cascades:** The proposed DeepHawkes model integrates the high prediction power of end-to-end deep learning technique into interpretable factors of Hawkes process for popularity prediction. The marriage between deep learning technique and a well-established interpretable process for modeling cascade dynamics bridges the gap between prediction and understanding of information cascades. This practice leads a new trend to develop new methods for popularity prediction. This is the first key contribution of this paper.
- **Data sets:** We collect and construct two different data sets, one from the most popular social platform in China, i.e., Sina Weibo, and the other from the publicly available academic platform, i.e., American Physical Society. Different from previous public-available datasets for popularity prediction, these two datasets include detailed cascade paths, recording the diffusion process over social networks. With such information, the two datasets offer us a great playground to investigate the effect of cascade tree on popularity prediction. We believe such a type of datasets could promote the development of new methods to leverage diffusion tree for popularity prediction.

The rest of this paper is structured as follows. In Section 2 we briefly review the related works. Section 3 gives the formal definition of popularity prediction and introduction of Hawkes process, while Section 4 details the proposed DeepHawkes model. In Section 5 we describe how the two data sets are collected and constructed, and then comprehensive experiments are conducted on these two data sets in Section 6. We conclude our work in Section 7.

2 RELATED WORK

Existing approaches for popularity prediction mainly fall into two categories: feature-based approaches and generative approaches. Recently, methods based on representation learning emerge with impressive predictive power. In this section, we briefly review these approaches, placing our method in appropriate context and helping readers identify the main contributions of this paper.

Feature-based approaches. Feature-based approaches generally consider the popularity prediction task as a regression problem [2, 3, 13–15] or a classification problem [4, 5, 16]. These approaches extracted various hand-crafted features for popularity prediction, which are often binded to specific datasets or social media platforms, e.g., Twitter [13, 16], Digg [2, 17], and Youtube [2, 3]. These features mainly include temporal features [2, 3], structural features [16, 18, 19], content features [15, 20–22] and features defined on early adopters [13, 23, 24]. Szabo et al. [2] observed that the future popularity of online content is linearly correlated with its early popularity. Later, Pinto et al. [3] extended this model by replacing the early cumulative popularity with multiple incremental popularity in equal-sized time intervals within observation time. In fact, the most effective features are temporal features, i.e., the timing when early adopters involve the retweet of cascades [4]. For structural features, Weng et al. [19] demonstrated that the future popularity of a meme can be predicted by quantifying its early spreading pattern in terms of community concentration. Prediction error was further reduced when combining content features with temporal and structural features [15]. Bakshy et al. [13] studied the features related to early adopters and found that user features, particularly user's influence in past and the number of followers, are informative predictors. Recently, in spite of exploring informative features, Martin et al. [14] also explored the limits of prediction in complex social systems, finding that their best performing models integrating users, content and past success features, can explain less than half of the variance in cascade sizes. In sum, feature-based approaches provide us a general understanding of popularity prediction, demonstrating the effectiveness of the features about structural, temporal, content and early adopters. However, their performance heavily depends on the extracted features, which are hard to design and measure. There is an urgent need for models that can automatically learn the representations of the input data based on an explicit and interpretable principle, which is exactly what this paper did.

Generative approaches. This line of literature generally regards the popularity cumulation of online content as an arrival process of retweets and models the intensity function in the arrival process for each message independently [10, 11, 25–28]. Shen et al. [9] employed Reinforced Poisson Processes (RPP) to model the three phenomena in social networks: attractiveness of an item, a temporal relaxation of attractiveness, and the rich-get-richer mechanism. Gao et.al extended the RPP model with various temporal relaxation functions and applied it to the prediction of retweet dynamics [26]. Later, instead of using the number of retweets as done in RPP to model the rich-get-richer phenomenon, Hawkes self-exciting point process was employed to directly model the specific contribution of each retweet, characterized by user influence and temporal relaxation function [29]. Hawkes process offers us a general framework to model the process about how a piece of information accrues attention, offering us high ease to understand the underlying mechanism governing the popularity dynamics of information cascades. Deep learning technique was also employed to learn the function of arrival rate in point process [30, 31]. However, these methods generally are not directly optimized for future popularity and learn parameters for each message independently. Consequently, they cannot fully leverage the information implied in the popularity dynamics of all cascades for popularity prediction. There still remains a gap between the interpretability and predictability. To solve this dilemma, we propose to learn the interpretable factors of Hawkes process, i.e., user influence, self-exciting mechanism and time decay effect under an end-to-end supervised framework.

Methods based on representation learning. Popularity prediction is difficult due to the openness of social networks. There may be various factors affecting the future popularity of online content. In order to deal with such complex popularity dynamics and automatically extract the useful information from raw data, methods based on representation learning are emerging in popularity prediction [12, 17, 32–34]. Mishra et al. [12] proposed a two-stage model, learning the parameters of Hawkes point process as representations, and feeding them together with other extracted features into machine learning model to finally make the popularity prediction. Hoang et al. [34] proposed to group users into cohesive clusters and then adopt tensor decomposition to make predictions. This method is actually learning the representation of each messages in the space spanned by the dimensions corresponding to user groups. The success of deep learning in different fields also inspires some end-to-end representation learning framework for cascade prediction. Li et al. [8] proposed DeepCas, which transforms the cascade graph as node sequences through random walk and learns the representation of each cascade under a deep learning framework. This line of works can automatically learn the valuable representations from data and is emerging with the success of representation learning in other areas. However, existing works either followed a two-stage way [12, 34], lacking the future popularity as a guide for representation learning, or ignored the predictive information revealed by traditional works, e.g., DeepCas [8] ignores the temporal information for popularity prediction. Moreover, deep learning methods lack clear interpretability to help us understand the underlying mechanisms governing the popularity dynamics of information cascades. In this paper, we take advantage of both the effective information demonstrated by traditional methods and the power of end-to-end representation learning framework for popularity prediction. The proposed DeepHawkes model inherits the high interpretability of Hawkes process and possesses the high predictive power of deep learning methods, bridging the gap between prediction and understanding of information cascades. This is the key contribution of this paper.

3 PRELIMINARIES

In this section, we start with giving the formal definition of popularity prediction studied in this paper, and then offer an introduction of Hawkes Process before presenting the proposed model.

3.1 Problem Definition

Existing works generally regard popularity prediction task either as a regression problem, i.e., predicting the exact future popularity [13, 14], or a relatively easier classification problem, i.e., predicting whether the messages will reach double size or surpass a certain threshold in future [4, 5]. In this paper, we focus on predicting the exact future popularity of messages.

Suppose we have M messages, denoted by $\mathcal{M} = \{m^i\}(1 \leq i \leq M)$. For each message m^i, we use a cascade $C^i = \{(u^i_j, v^i_j, t^i_j)\}$ to record the diffusion process of message m^i, where the tuple (u^i_j, v^i_j, t^i_j) corresponds to the jth retweet, meaning that user v^i_j retweets message m^i from user u^i_j, and the time elapsed between the original post and the jth retweet is t^i_j. For example, in Figure 1, the cascade can be represented as $\{(\oslash, A, t_A = 0), (A, B, t_B), (B, C, t_C), (B, D, t_D), (D, E, t_E)\}$. The popularity R^i_t of message m^i up to time t is defined as the number of its retweets, i.e., $|\{(u^i_j, v^i_j, t^i_j)|t^i_j \leq t\}|$. The problem of popularity prediction to be solved in this paper is then formulated as:

Popularity Prediction: Given the cascades in the observation time window $[0, T)$, it predicts the incremental popularity ΔR^i_T between observed popularity R^i_T and final popularity R^i_∞ of each cascade C^i.

We predict the *incremental* popularity instead of the final popularity to avoid the intrinsic correlation between the observed popularity and the final popularity. This forms a more difficult scenario of popularity prediction.

3.2 Hawkes Process

Hawkes self-exciting point process, a stochastic event model, is proved to be effective for modeling popularity dynamics [11, 12, 25]. The key to such process-based methods is to model the arrival rate of new events, which can be written as

$$\rho^i_t = \sum_{j:t^i_j<=t} \mu^i_j \phi(t - t^i_j), \tag{1}$$

where ρ^i_t is the arrival rate of new retweets for message m^i at time t, t^i_j is the time elapsed between the original post and the jth retweet, μ^i_j is the number of potential users who will be directly influenced by the jth retweet, $\phi(t)$ is a temporal decay function. There are three key factors captured in Hawkes process: **influence of users**—influential users contributes more to the arrival rate of new retweets, indicating that tweets retweeted by influential users tend to get retweeted more; **self-exciting mechanism**—each retweet contributes to the arrival rate of new retweets in the future; **time decay effect**—the influence of retweet decays with the increase of time.

Though Hawkes process can well explain the observed retweets, it learns parameters for each cascade independently and is not optimized for future popularity. Next, we will make an analogy to interpretable factors of Hawkes process under an end-to-end supervised deep learning framework.

4 MODEL

We now introduce the proposed DeepHawkes model. The framework of DeepHawkes model takes the cascade as input and outputs the incremental popularity (shown as Figure 1). It first transforms the cascade into a set of diffusion paths, each one depicting the process of information propagation to each retweet user within observation time. The main part of the DeepHawkes model contains

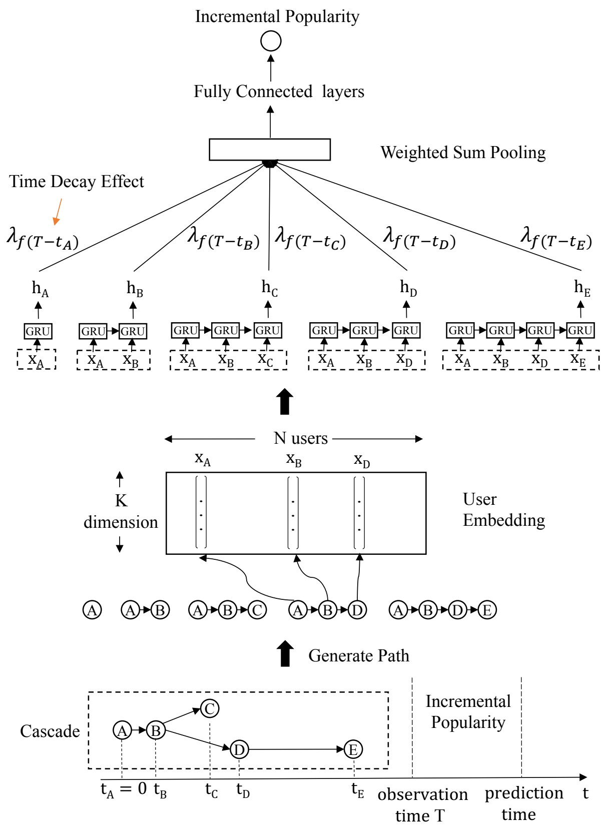


Figure 1: Framework of DeepHawkes model

three components: (1) **user embedding**, embedding user identity into a low-dimensional space to represent the influence of users; (2) **path encoding and sum pooling**, feeding the retweet path into recurrent neural network and sum up the value of the last hidden layer for all paths to model the self-exciting mechanism of each retweet; (3) **non-parametric time decay effect**, using a non-parametric way to learn the time decay effect in information diffusion. These three components of the DeepHawkes model exactly correspond to the three key interpretable factors of Hawkes process. At last, the representation of cascade is connected by a dense layer to prediction target, i.e., incremental popularity. Next, we give a detailed introduction to these components one by one.

4.1 User Embedding

As indicated by Hawkes process, the future popularity of online content is related to the participating users—the more influential the participants are, the more retweets it will receive. In previous works, only the number of fans are considered as an indication of user influence [11, 12], which means that as long as two users have the same number of fans, their impact on the future popularity are exactly the same. This is obviously a simplification of the real case. Indeed, users with different positions in social networks, or different interests or characteristics may have different influence on cascades diffusion, even though they have the same number of fans. Feature-based approaches also demonstrate that user features, like age, gender, past success and so on, is an effective type of features for popularity prediction [4, 5].

The above two methods, using the number of fans or using other user features, are anticipating to use these characteristics to represent and distinguish users or users’ influence, yet there is no principle to tell us which quantity is the best. In fact, different characteristics may be effective for different situations. If this is the case, why don’t we directly learn users’ representations from history data which are suitable and customized for specific scenario? Therefore, in this paper, we propose to directly use users’ identity and learn their representations under a supervised framework for popularity prediction. In other words, we use these learned representations as indications of users’ influence.

Specifically, each user in the generated retweet path is represented as a one-hot vector, $q \in R^N$, where N is the total number of users. All users share an embedding matrix $A \in R^{K \times N}$, where K is an adjustable dimension of embedding. This user embedding matrix converts each user into its representation vector

$$x = Aq, \tag{2}$$

where $x \in R^K$. Note that the user embedding matrix A is learned during the training process, supervised by the future popularity. In this way, the learned user embeddings are optimized for popularity prediction.

4.2 Path Encoding and Sum Pooling

The second key factor of Hawkes process is the mechanism of self-exciting, i.e., each retweet increases the arrival rate of new retweet in the future. However, in real situation, not only the current retweet user itself promotes the bloom of future popularity, but also the entire retweet path makes contribution. Consequently, in this paper, instead of only modeling the retweet user itself, we proposed to encode the influence of the entire retweet path through recurrent neural network. Indeed, there are two intuitions behind such practice: (1) **Transitivity of influence.** The effect of influence transitivity means that the previous participants not only influence its direct retweeters, but also cause influence on indirect retweeters by the way of transitivity. Taking the scene of Sina Weibo as an example, user A published a message m^i, user B retweeted this message from user A, and user C retweeted this message from user B. For other users, e.g., user D, who saw the message m^i and the retweet path $A \rightarrow B \rightarrow C$, the retweet probability of user D may increase when he/she saw the participants of this message, i.e., user B or user A, are authoritative. In other words, each user in the entire retweet path may have an impact on subsequent retweet after current retweet. (2) **Structural position.** We characterize the importance of users’ structural positions in the cascade tree by their occurrence frequency in cascade paths. For example, in Figure 1, since the rest retweets are all directly or indirectly caused by user B, the structural position of user B in this cascade is quite important. This importance of structural position is naturally captured in the DeepHawkes model by user’s occurrence frequency in multiple retweet paths, i.e., user B will appear in each retweet path of user B, C, D and E. In sum, we model the influence of the entire retweet path for each retweet and then sum them up to achieve the mechanism of self-exciting.

We encode the entire retweet path p_j^i for each retweet user u_j^i, $1 \le j \le R_T^i$, of message m^i through a Gated Recurrent Unit (GRU) [35, 36], a specific type of recurrent neural network (RNN) [37]. Specifically, denoting step k the k-th user in retweet path p_j^i, let's describe how GRU computes the k-th hidden state $h_k = \text{GRU}(x_i, h_{k-1})$, where the output $h_k \in R^H$, the inputs $x_k \in R^K$ is the user embedding, $h_{k-1} \in R^H$ is the previous hidden state, k is the dimension of user embedding, and H is the dimension of hidden state.

First, the reset gate $r_k \in R^H$ is computed by

$$r_k = \sigma(W^r x_k + U^r h_{k-1} + b^r), \tag{3}$$

where $\sigma(\cdot)$ is the sigmoid activation function, $W^r \in R^{H \times K}$, $U^r \in R^{H \times H}$ and $b^r \in R^H$ are GRU parameters learned during training.

Similarly, the update gate $z_k \in R^H$ is computed by

$$z_k = \sigma(W^z x_k + U^z h_{k-1} + b^z), \tag{4}$$

where $W^z \in R^{H \times K}$, $U^z \in R^{H \times H}$ and $b^z \in R^H$.

The actual activation of hidden state h_k is then computed by

$$h_k = z_k \odot h_{k-1} + (1 - z_k) \odot \tilde{h}_k, \tag{5}$$

where

$$\tilde{h}_k = \tanh(W^h x_k + r_k \odot (U^h h_{k-1}) + b^h), \tag{6}$$

$\odot$ represents element-wise product, $W^h \in R^{H \times K}$, $U^h \in R^{H \times H}$ and $b^h \in R^H$.

For each retweet path p_j^i, we use the last hidden states as the representation of the entire diffusion path, denoted as h_j^i. The representation c^i of cascade C^i is then assembled by the sum pooling mechanism:

$$c^i = \sum_{j=1}^{R_T^i} h_j^i, \tag{7}$$

where $c^i \in R^H$.

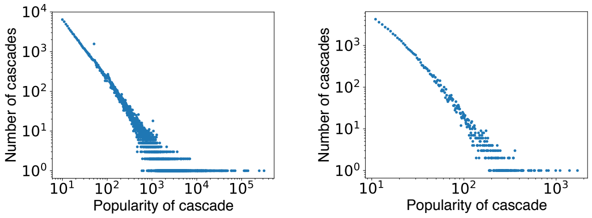
4.3 Non-parametric Time Decay Effect

The influence of retweet decays with time passing, coming into being the last key factor of Hawkes process, i.e., time decay effect. Existing works generally predefine the shape of the time decay function [12], e.g., power-law functions $\phi^p(t) = (t + c)^{-(1+\theta)}$ and exponential functions $\phi^e(t) = e^{-\theta t}$, based on prior domain knowledge. However, in different scenerios, it's actually hard to decide which one of the above functions should be used .

To avoid the above problem suffered by existing works, we propose to directly learn the time decay effect through a non-parametric way in this paper. Assuming that we observe the diffusion of all messages within time T, then the unknown time decay effect $\phi(t)$ is a function, changing continuously over $[0, T)$. We now approximate this time decay effect function by splitting the length of time range T into L disjoint intervals $\{[t_0 = 0, t_1), [t_1, t_2), ..., [t_{L-1}, t_L = T)\}$ and learning the corresponding discrete variable of time decay effect $\lambda_l, l \in (1, 2, ..., L)$. The mapping function f from continuous time to time interval are defined as

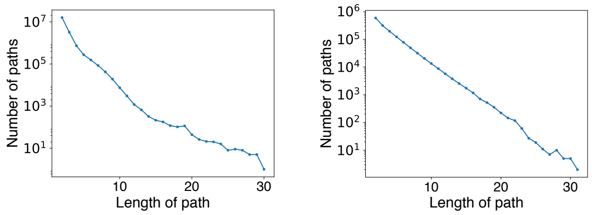
$$f(T - t_j^i) = l, \qquad if \quad t_{l-1} \le T - t_j^i < t_l \tag{8}$$

where t_j^i is the time elapsed between the original post and the jth retweet of message m^i, $f(T - t_j^i)$ is the corresponding time interval of time decay effect for jth retweet.



(a) Sina Weibo (b) APS

Figure 2: Distribution of popularity



(a) Sina Weibo (b) APS

Figure 3: Distribution of the length of paths

For cascade C^i of message m^i within observation time window $[0, T)$, each retweet path is denoted as p_j^i while the retweet time of the last user in this retweet path is t_j^i. Then the representation c^i of cascade C^i when considering time decay effect is assembled by a weighted sum pooling mechanism:

$$c^i = \sum_{j=1}^{R_T^i} \lambda_{f(T - t_j^i)} h_j^i. \tag{9}$$

4.4 Output Layer

The last part of our DeepHawkes model consists of a multi-layer perceptron (MLP), taking the cascade representation c^i as input and outputting one final unit:

$$\Delta \hat{R}_T^i = MLP(c^i). \tag{10}$$

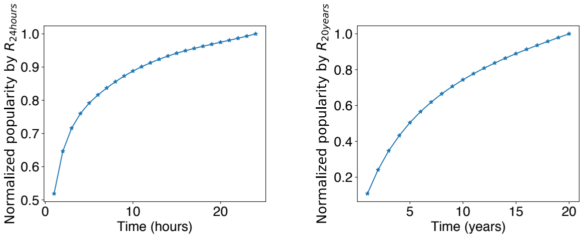
The objective function to be minimized is defined as

$$obj = \frac{1}{M} \sum_{i=1}^{M} (\log \Delta \hat{R}_T^i - \log \Delta R_T^i)^2, \tag{11}$$

where $\Delta \hat{R}_T^i$ is the predicted incremental popularity for cascade C^i, ΔR_T^i is the true incremental popularity and M is the total number of cascades. We take log-transformation for the incremental popularity since the original square loss can be easily affected by outliers while the transformed objective function behaves similar to MAPE (mean absolute percentage error) and is easier to be optimized.

5 DATA SETS

We evaluate the DeepHawkes model by applying it to two scenarios of popularity prediction. One scenario is predicting the future size of retweet cascades in Sina Weibo. The other scenario is predicting the citation count of papers. The two scenarios offer us a great



(a) Sina Weibo (b) APS

Figure 4: Normalized popularity

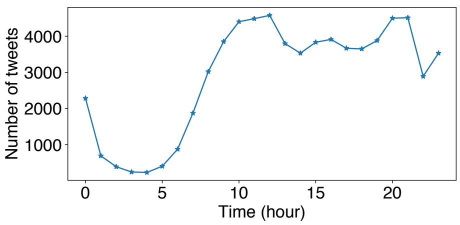


Figure 5: Diurnal rhythm of users in Sina Weibo.

common playground to compare the proposed model with state-of-the-art methods. In particular, using two scenarios could verify the generality of the proposed model and avoid the risk of binding to specific platforms. Next we describe the two scenarios and the dataset statistics that are relevant to the experimental settings in Section 6.

5.1 Sina Weibo Dataset

The first dataset is from Sina Weibo, the most popular microblogging platform in China. In Sina Weibo, social network among users is formed by their following relationships, and tweets are spread conveniently along such relationships. Everyday, more than 10 million original messages are posted, and these messages receive nearly one hundred million retweets, comments, and likes. In this paper, we focus on the retweets of messages, considering that retweets form a natural information cascade while comments and likes mainly reflect the direct interaction among users.

We collect all the original messages produced on June 1, 2016, and track the retweets of each message within the next 24 hours. Messages with less than 10 retweets are filtered out, remaining 119,313 messages in total. Figure 2(a) shows the distribution of cascade popularity—the number of retweets of each message, following a power-law distribution. We obtain the retweet cascade of each message by parsing the text of its retweets. Specifically, we use “//@username:” to identify the retweeting path from the text of retweets. For example, when user A retweets a message originally posted by user D and the text of retweet is “...//@B:...//@C:...”, we obtain a retweet path $D \rightarrow C \rightarrow B \rightarrow A$. As shown in Figure 3(a), the distribution of path lengths follows an exponential distribution, consistent with reported results [38].

Now we introduce the scenario of popularity prediction in Sina Weibo. For each cascade, we use the number of tweets within 24 hours, i.e., $R_{24hours}$ as an approximation to the final popularity R_∞.

Table 1: Statistics of the data sets

		Sina Weibo			APS		
T		1 h	2 h	3 h	5 y	7 y	9 y
	train	29,531	35,403	38,576	16,299	21,171	24,658
M	val	6,328	7,586	8,266	3,582	4,507	5,254
	test	6,327	7,586	8,266	3,475	4,589	5,279
Avg.	train	1.24	1.26	1.27	2.12	2.28	2.41
path	val	1.27	1.30	1.30	2.07	2.27	2.41
length	test	1.25	1.27	1.28	2.14	2.30	2.42
Avg.	train	62.7	68.5	70.7	17.8	19.2	20.2
	val	66.2	72.3	75.5	17.3	18.9	20.0
R^i_T	test	61.5	66.7	70.8	17.9	19.1	20.3

As shown in Figure 4(a), the popularity of cascade saturates when approaching 24 hours after publication, indicating $R_{24hours}$ offers a good approximation to the final popularity. For the length T of observation time window, we consider three settings, i.e., $T = 1$ hour, 2 hours and 3 hours, corresponding to the timing that the popularity reaches about 50%, 60%, 70% of the final popularity respectively. For each T, we only consider the cascades with no less than 10 retweets and no more than 1000 retweets in observation time window. Next, we split the dataset into training, validation, and test set. To avoid the effect of diurnal rhythm of users in Sina Weibo (Figure 5), we only consider the cascades with the publication time between 8:00 and 18:00. With such a setting, each cascade has 6 hours in active period (8:00-24:00) to accrue retweets. Indeed, as shown in Figure 4(a), on average, a message receives about 80% retweets within 6 hours. Finally, we sort all the remaining cascades by their publication time, and take the first 70% as training set, the middle 15% as validation set, and the last 15% as test set. Statistics of the preprocessed dataset used for popularity prediction are shown in Table 1.

5.2 APS Citation Dataset

We now turn to the scenario of predicting the citation count of papers. The dataset used in this paper is from American Physical Society (APS) [9], including all the papers published by the 11 APS journals between 1893 and 2009, and the citations among these papers. In this scenario, the unit of time is day, and for each citation we record the number of days elapsed since the publication of the cited paper. All the citations to a paper form a cascade, and the popularity of cascade is the number of citations. Figure 2(b) shows the popularity of cascade in the APS citation dataset, exhibiting a power-law distribution.

Different from the scenario of retweet prediction where each retweet corresponds to a unique retweeter and the retweet path is explicitly recorded, the scenario of citation prediction requires some tricky transformation and preprocessing, making analogy to the scenario of retweet prediction. First, we take all coauthors of a paper as an author group, acting like a distinct person. In this way, we circumvent the issue of duplicated citations suffered by the way when each author is taken as a distinct person [8]. Second, we propose a method to figure out the implicit cascade path. Similar to

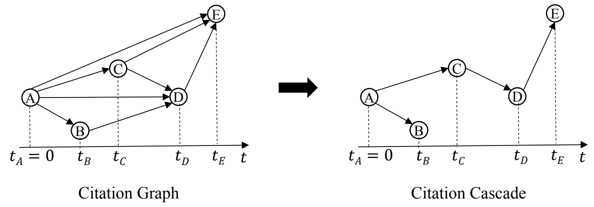


Figure 6: Illustration of citation cascade construction

retweet cascade in Sina Weibo, citation cascade of a paper captures the diffusion process of the idea or impact of the paper. Each citing paper corresponds to a retweet, with its references exposing the potential cascade path, playing the role of "//@username:" in text of retweet. Specifically, each citing paper takes its *latest* referenced paper that also cited the target paper as the source where the citing paper's authors learned the idea of the target paper. Thus, given a target paper, we can identify the cascade path according to the reference of its citing papers. For example, the left part of Figure 6 shows the citation graph of a target paper, where A is the authors of target paper and authors B, C, D, E all cited this target paper. The right part of Figure 6 shows the citation cascade with the citation path identified. When considering the diffusion process of the idea of target paper to authors E, we can regard that authors E heard the idea of target paper from the latest referenced paper which also cited the target paper, i.e., paper of authors D. Following this practice, the entire diffusion path of authors E is $A \rightarrow C \rightarrow D \rightarrow E$. The distribution of path length in the APS dataset is given in Figure 3(b), exhibiting a clear exponential distribution.

Finally, we introduce the setting for the citation prediction scenario. Similar to Sina Weibo, for each paper, we take the citation count within 20 years after publication as an approximation to its final popularity. Accordingly, we only use the papers published between 1893 and 1989, leaving each paper 20 years to develop its cascade. For observation time window, we choose T = 5 years, 7 years and 9 years, corresponding to the year that the popularity reaches about 50%, 60% and 70% of the final popularity. Papers with less than 10 citations within observation time window are ignored. We split the dataset into training set, validation set and test set by 70%, 15%, and 15%. Statistics of the preprocessed dataset are shown in Table 1.

6 EXPERIMENTS

In this section, we compare the prediction performance of the proposed DeepHawkes model with state-of-the-art approaches. Moreover, we perform detailed analysis to understand the role of each component in DeepHawkes for popularity prediction.

6.1 Baselines

As we claimed in Section 2, existing methods for popularity prediction are classified into three categories. Therefore, we select state-of-the-art method in each category as strong baselines. For feature-based approaches, we use the features mentioned by Cheng et al. [4] and Shulman et al. [5] to investigate the predictbility of cascades. For generative approaches, we use SEISMIC [11], a successful implementation of Hawkes process, as baseline. For models based on representation learning, we use the recent work, i.e., DeepCas [8], as baseline.

Feature-linear. As demonstrated by a recent study, temporal features, structural features, and features defined on early adopters are the most informative features for popularity prediction [4, 5]. We extract all the predictive features of these three types that could be generalized across data sets. These features are:

- *Temporal features.* This type of features capture the speed of adoptions within observation time window. We extract the mean time between each retweet [5], the cumulative popularity [2] and incremental popularity [3] every 10 minutes for Sina Weibo and every 3 months for APS.
- *Structural features.* As the entire social network structure is hard to obtain, we only consider the structural features of the cascade. We count the number of leaf nodes, and calculate the average degree [8], average and max length of retweet path [5] as a measure of centrality and density.
- *Features of early adopters.* We use the number of fans and friends of the source user and the average number of fans and friends of all retweet users as features defined on early adopters [4, 5].

After extracting all the predictive features, we feed them into a linear regression model with L2 regularization. Note that the label (incremental popularity) has been logarithmically transformed before fed into linear regression, making the baseline of feature-linear optimizes the same objection function with DeepHawkes.

SEISMIC [11] is an implementation of Hawkes self-exciting point process, which uses the number of fans as user's influence and models the self-exciting mechanism of each retweet. In addition, after observing several tweets in the training set as prior knowledge, SEISMIC uses the power-law function to fit the time decay effect in information diffusion. This is one of the state-of-the-art generative approaches for popularity prediction.

DeepCas [8] is state-of-the-art deep learning method for popularity prediction, which learns the representation of cascade graphs in an end-to-end manner and predicts the future incremental popularity. It mainly utilizes the information of structure and node identities for prediction.

6.2 Experimental Setup

For evaluation metric, following the practice of DeepCas, we use the mean square log-transformed error, defined as

$$MSLE = \frac{1}{M} \sum_{i=1}^{M} SLE^i, \tag{12}$$

where M is the total number of messages, SLE^i is the square log-transformed error for a given message m^i defined as $SLE^i = (\log \Delta\hat{R}^i_T - \log \Delta R^i_T)^2$, $\Delta\hat{R}^i_T$ is the predicted incremental popularity for message m^i and ΔR^i_T is the true incremental popularity.

Furthermore, since SEISMIC is sensitive to outlier error, we also use median square log-transformed error (mSLE) as an evaluation metric, which is defined as the 50th percentile of the distribution of SLE over all messages.

For model parameters, we choose L2-coefficient from $\{1,10^{-1},10^{-2},10^{-3},10^{-4}\}$ for linear regression model. For SEISMIC, it estimates

Table 2: Overall prediction performance

	Sina Weibo						APS					
T	1 hour		2 hours		3 hours		5 years		7 years		9 years	
Evaluation Metric	MSLE	mSLE	MSLE	mSLE	MSLE	mSLE	MSLE	mSLE	MSLE	mSLE	MSLE	mSLE
SEISMIC	—	0.657	—	1.084	—	1.575	—	0.874	—	0.717	—	0.712
Feature-linear	3.701	1.058	3.365	1.105	3.328	1.139	1.582	0.679	1.508	0.674	1.456	0.722
DeepCas	3.631	0.808	3.213	0.837	3.107	0.902	1.629	0.671	1.538	0.603	1.462	0.662
DeepHawkes	**2.448**	**0.650**	**2.279**	**0.675**	**2.223**	**0.639**	**1.510**	**0.652**	**1.337**	**0.584**	**1.211**	**0.605**

the temporal decay function $\phi(t)$ with the following form:

$$\phi(t) = \begin{cases} c, & 0 < t \leq s0 \\ c(t/s0)^{-(1+\theta)}, & t > s0 \end{cases} . \tag{13}$$

When implementing this baseline for Sina Weibo, we adopt the setting of parameters used in [11], i.e., setting the constant period $s0$ to 5 minutes and power law decay parameters $\theta = 0.231$ and $c = 6.27 \times 10^{-4}$. Besides, we choose the value of mean degree n^* from $\{10, 20, 50, 100\}$ to minimize the mSLE of training set. As for APS citation dataset, we reestimate the parameters of temporal decay function due to the big differences between the propagation process of papers and messages. The estimated $s0$ equals to 500 days, the power law decay parameters $\theta = 0.788$ and the corresponding $c = 8.81 \times 10^{-4}$. Similarly, we choose the value of mean degree n^* from $\{1, 3, 5, 10, 20\}$ to minimize the error of training set. For deep neural network models, we follow the settings of DeepCas [8], where the embedding size of users equals to 50, the hidden layer of each GRU has 32 units and the hidden dimensions of the two-layer fully connected layers are 32 and 16, respectively. Meanwhile, the learning rate for user embeddings is 5×10^{-4} and the learning rate for other variables is 5×10^{-3}. The batch size for each iteration is set to 32 and the training process stop immediately as long as the loss of validation set doesn't decline for 10 consecutive iterations. The time interval of non-parametric time decay effect is set to 10 minutes for Sina Weibo and 3 months for APS.

6.3 Prediction Performance

We compare the prediction performance of DeepHawkes with three strong baselines on two scenarios, i.e., predicting the future size of retweet cascades in Sina Weibo and predicting the citation count of papers in APS. Table 2 shows that the proposed DeepHakes model outperforms all the baselines, with a significant reduction of prediction error on both scenarios under two evaluation metrics. Note that the mSLE is not optimized as loss function in the training process. Consequently, mSLE is not monotonic with respect to the length of observation time.

For SEISMIC, we only use the mSLE as the evaluation metric. Although SEISMIC is one of the state-of-art Hawkes process based models, it actually performs not well in cascade prediction since it lacks the future popularity as a guide. As shown in Figure 2, when learning the interpretable factors of Hawkes process under an end-to-end supervised framework, our proposed DeepHawkes model significantly reduces the prediction error.

Our DeepHawkes model also significantly outperforms Feature-linear model since the prediction performance of this baseline highly depends on hand-crafted features which are hard to design.

As the state-of-the-art end-to-end deep learning method for popularity prediction, DeepCas has a quite good performce. However, since DeepCas ignores the most predictive information of time, its prediction performance is far worse than the proposed DeepHawkes model.

Besides these three strong baselines, we also consider the prediction performance of Feature-deep model, which transforms and recombines the hand-crafted features in a non-linear way. Actually, feeding all hand-crafted features into deep neural networks can achieve quite excellent prediction results, with $MLSE = 2.047$ when observing for 1 hour in Sina Weibo and $MLSE = 1.479$ when observing for 5 years in APS. However, such Feature-deep model cannot help people understand the underlying popularity dynamics of information cascades since it lacks an analogy to interpretable factors like DeepHawkes, and this is where the deep learning framework is often criticized by researchers. In contrast, the proposed DeepHawkes model inherits the high interpretability of Hawkes process while also possesses the competing predictive power of deep learning methods, bridging the gap between prediction and understanding of information cascades. Indeed, we can combine the learned quantity (i.e.,the last hidden units) of DeepHawkes model into Feature-deep framework to further improve the prediction performance as Mishra et al. did in [12].

Comparing the two different scenarios for popularity prediction, we can see that prediction errors are much smaller in APS than that in Sina Weibo. This indicates that the future size of retweet cascades in Sina Weibo is more difficult to predict than the future citation of papers in APS.

Overall, the proposed DeepHawkes model performs fairly well on popularity prediction for both retweets of messages and citations of papers, not only outperforming both feature-based and generative approaches, but also outperforming the state-of-the-art deep learning methods for popularity prediction.

6.4 Analysis of User Embedding and Path Encoding

To demonstrate the effectiveness of the components of user embeddings and path encoding in the proposed DeepHawkes model, we give a detailed analysis in this part. For this purpose, we present three simplified versions of DeepHawkes, denoted as DH-origin,

Table 3: Prediction performance of variants of DeepHawkes

	Sina Weibo			APS		
T (hours/years)	1	2	3	5	7	9
DH-origin	5.261	4.934	4.717	2.060	1.920	1.874
DH-embedding	4.585	4.296	3.922	1.725	1.506	1.380
DH-path	2.485	2.448	2.299	1.564	1.381	1.260
DeepHawkes	2.448	2.279	2.223	1.510	1.337	1.211

DH-embedding and DH-path, where one or two components are removed from the complete DeepHawkes model.

DH-origin model directly uses the number of fans as the representation of a user like previous works [11, 12], and simply summarizes the influence of each retweet in Hawkes process, instead of encoding the influence of the entire retweet path through a GRU structure. This is the simplest version of DeepHawkes, abandoning the component of user embeddings and path encoding. We construct this version to offer a basic model for comparison.

DH-embedding model includes the component of user embeddings on the basis of DeepHawkes-origin. We construct this version to demonstrate the effectiveness of the component of user embeddings, i.e., learning user embedding as the representation of user's influence by the guide of future popularity, instead of using hand-crafted quantities, e.g., the number of fan.

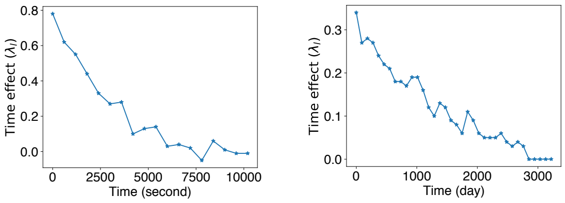
DH-path model includes the component of path encoding through GRU structures on the basis of DeepHawkes-origin. We construct this version to demonstrate the effectiveness of the component of path encoding, i.e., considering the influence of the entire retweet path instead of only the current retweet user in Hawkes process.

Table 3 shows the prediction performance of these three simplified versions of DeepHawkes. We can see that DH-embedding leads to a certain reduction of prediction error when compared with DH-origin, demonstrating that learning the user embedding matrix as proposed in this paper, instead of directly using the number of fans as the representation of a user, can improve the performance of popularity prediction. This result further confirms that it is a better way to learn the user embeddings from history, directly guided by future popularity, rather than designing hand-crafted features, like the number of fans. Similarly, DH-path brings a remarkable improvements of the prediction performance comparing with DH-origin, demonstrating the necessity and effectiveness of the component of path encoding. This indicates that the future popularity is not only influenced by current retweet user, but also influenced by the entire retweet path.

In summary, the components of user embeddings and path encoding through GRU structure in the proposed DeepHawkes model improve the performance of popularity prediction respectively, which are learned under deep learning framework and guided by future popularity. Experimental results further demonstrate the effectiveness and necessity of these components in DeepHawkes.

6.5 Analysis of Time Effect

We now investigate whether the non-parametric way of learning time decay effect can automatically capture the specific temporal decay of influence in each scenario.



(a) Sina Weibo

(b) APS

Figure 7: Time decay effect learned by DeepHawkes

Recall that we now use L discrete parameters $\lambda_l, 1 \leq l \leq L$ to approximate the unknown time decay function $\phi(t), 0 \leq t < T$. The learned parameters λ_l are shown in Figure 7(a) for Sina Weibo and Figure 7(b) for APS, where the length of observation time window T is 3 hours and 9 years respectively. It can be seen that the time effect decreases with time passing, indicating that the fresher a retweet is, the more influence it causes. This is consistent with the time decay factor assumed in Hawkes process. However, there are several differences about the time decay effect between Sina Weibo and APS. First, the time decay effect is more apparent in Sina Weibo than in APS, since that the effect of a retweet message is 0.78 when it is just published in Sina Weibo while the effect of a reference paper is only 0.34 when it is just emerged in APS. This phenomena indicates that the popularity of messages is dominated by early retweeters, e.g., source users, while citations are accrued gradually. Second, the time decay effect decays much faster in Sina Weibo than in APS, since the time decay effect decreases 50% of its original effect after 30 minutes in Sina Weibo while the time decay effect decreases 50% of its original effect after 3 years. This is due to that the messages in Sina Weibo are updated much faster than papers in APS, resulting that the attention of old retweet messages are replaced by new messages much quickly in Sina Weibo. These experimental results demonstrate that our non-parametric way is flexible enough to capture these different time decay effect without prior domain knowledge.

7 CONCLUSION

In this paper, we proposed the DeepHawkes model to bridge the gap between prediction and understanding of information cascades, leveraging end-to-end deep learning to learn the interpretable factors of Hawkes process—a widely-used generative process to model information cascade. Consequently, the proposed DeepHawkes model not only inherits the high interpretability of Hawkes process but also possesses the high predictive power of deep learning methods. Experiments conducted on two scenarios, i.e., predicting the size of retweet cascades in Sina Weibo and predicting the citation of papers in APS, demonstrate that DeepHawkes model not only outperforms both feature-based and generative approaches, but also outperforms state-of-the-art deep learning methods for popularity prediction.

The DeepHawkes model captures and extends the three interpretable factors of Hawkes process under deep learning framework, i.e., influence of users, self-exciting mechanism of each retweet and the time decay effect in information diffusion. It's instructive to find that learning user embeddings as the influence representation of users by the guide of future popularity is useful in popularity prediction. Besides, considering the entire retweet path through GRU structure instead of only considering the current retweet user in Hawkes process can significantly improve the prediction performance. In addition, the DeepHawkes model is flexible to learn the time decay effect using the proposed non-parametric way without prior domain knowledge. As future work, we will devote to directly learning the rate function of the Hawkes process using deep learning methods, beyond making an analogy between the learned features and the factors of Hawkes process.

8 ACKNOWLEDGEMENT

This work was funded by the National Basic Research Program of China (973 Program) under grant numbers 2014CB340401 and 2013CB329602, and the National Natural Science Foundation of China under grant numbers 61472400, 61425016, 61232010, and 61602439. Huawei Shen is also funded by K. C. Wong Education Foundation, Youth Innovation Promotion Association of CAS and the CCF-Tencent RAGR (No. 20160107).

REFERENCES

[1] A. Tatar, M. D. D. Amorim, S. Fdida, and P. Antoniadis. 2014. A survey on predicting the popularity of web content. *Journal of Internet Services and Applications* 5, 1, 8.

[2] G. Szabo and B. A. Huberman. 2010. Predicting the Popularity of Online Content. *Commun. ACM* 53, 8, 80–88.

[3] H. Pinto, J. M. Almeida, and M. A. Gonçalves. 2013. Using Early View Patterns to Predict the Popularity of Youtube Videos. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining (WSDM '13)*. ACM, 365–374.

[4] J. Cheng, L. Adamic, P. A. Dow, J. M. Kleinberg, and J. Leskovec. 2014. Can Cascades Be Predicted?. In *Proceedings of the 23rd International Conference on World Wide Web (WWW '14)*. ACM, 925–936.

[5] B. Shulman, A. Sharma, and D. Cosley. 2016. Predictability of Popularity: Gaps between Prediction and Understanding. In *Tenth International AAAI Conference on Web and Social Media (ICWSM'16)*. AAAI press, 348–357.

[6] W. Ding, Y. Shang, L. Guo, X. Hu, R. Yan, and T. He. 2015. Video Popularity Prediction by Sentiment Propagation via Implicit Network. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM '15)*. ACM, 1621–1630.

[7] B. Chang, H. Zhu, Y. Ge, E. Chen, H. Xiong, and C. Tan. 2014. Predicting the Popularity of Online Serials with Autoregressive Models. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM '14)*. ACM, 1339–1348.

[8] C. Li, J. Ma, X. Guo, and Q. Mei. 2017. DeepCas: An End-to-end Predictor of Information Cascades. In *Proceedings of the 26th International Conference on World Wide Web Companion (WWW '17)*. ACM, 577–586.

[9] H. W. Shen, D. Wang, C. Song, and A. L. Barabási. 2014. Modeling and Predicting Popularity Dynamics via Reinforced Poisson Processes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI'14)*. AAAI Press, 291–297.

[10] P. Bao, H. W. Shen, X. Jin, and X. Cheng. 2015. Modeling and Predicting Popularity Dynamics of Microblogs Using Self-Excited Hawkes Processes. In *Proceedings of the 24th International Conference on World Wide Web (WWW '15)*. ACM, 9–10.

[11] Q. Zhao, M. A. Erdogdu, H. Y. He, A. Rajaraman, and J. Leskovec. 2015. SEISMIC: A Self-Exciting Point Process Model for Predicting Tweet Popularity. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15)*. ACM, 1513–1522.

[12] S. Mishra, M. A. Rizoiu, and L. Xie. 2016. Feature Driven and Point Process Approaches for Popularity Prediction. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM '16)*. ACM, 1069–1078.

[13] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts. 2011. Everyone's an Influencer: Quantifying Influence on Twitter. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining (WSDM '11)*. ACM, 65–74.

[14] T. Martin, J. M. Hofman, A. Sharma, A. Anderson, and D. J. Watts. 2016. Exploring Limits to Prediction in Complex Social Systems. In *Proceedings of the 25th International Conference on World Wide Web (WWW '16)*. ACM, 683–694.

[15] O. Tsur and A. Rappoport. 2012. What's in a Hashtag?: Content Based Prediction of the Spread of Ideas in Microblogging Communities. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining (WSDM '12)*. ACM, 643–652.

[16] D. M. Romero, C. Tan, and J. Ugander. 2013. On the Interplay between Social and Topical Structure. In *International Conference on Weblogs and Social Media (ICWSM '13)*.AAAI press, 516–525.

[17] M. Ahmed, S. Spagna, F. Huici, and S. Niccolini. 2013. A Peek into the Future: Predicting the Evolution of Popularity in User Generated Content. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining (WSDM '13)*. ACM, 607–616.

[18] P. Bao, H. W. Shen, J. Huang, and X. Cheng. 2013. Popularity Prediction in Microblogging Network: A Case Study on Sina Weibo. In *Proceedings of the 22Nd International Conference on World Wide Web (WWW '13)*. ACM, 177–178.

[19] L. Weng, F. Menczer, and Y. Y. Ahn. 2013. Virality Prediction and Community Structure in Social Networks. *Scientific Reports* 3, 8, 2522.

[20] S. Petrovic, M. Osborne, and V. Lavrenko. 2011. RT to Win! Predicting Message Propagation in Twitter. In *International Conference on Weblogs and Social Media (ICWSM '13)*.AAAI press, 586–589.

[21] Z. Ma, A. Sun, and G. Cong. 2013. On predicting the popularity of newly emerging hashtags in Twitter. *Journal of the Association for Information Science and Technology* 64, 7, 1399–1410.

[22] L. Hong, O. Dan, and B. D. Davison. 2011. Predicting Popular Messages in Twitter. In *Proceedings of the 20th International Conference Companion on World Wide Web (WWW '11)*. ACM, 57–58.

[23] K. Lerman and A. Galstyan. 2008. Analysis of Social Voting Patterns on Digg. In *Proceedings of the First Workshop on Online Social Networks (WOSN '08)*. ACM, 7–12.

[24] P. Cui, S. Jin, L. Yu, F. Wang, W. Zhu, and S. Yang. 2013. Cascading Outbreak Prediction in Networks: A Data-driven Approach. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '13)*. ACM, 901–909.

[25] R. Crane and D. Sornette. 2008. Robust dynamic classes revealed by measuring the response function of a social system. *Proceedings of the National Academy of Sciences of the United States of America* 105, 41, 15649.

[26] J. Gao, H. W. Shen, S. Liu, and X. Cheng. 2016. Modeling and Predicting Retweeting Dynamics via a Mixture Process. In *Proceedings of the 25th International Conference Companion on World Wide Web (WWW '16)*. 33–34.

[27] M. Rizoiu, L. Xie, S. Sanner, M. Cebrian, H. Yu, and P. V. Hentenryck. 2017. Expecting to Be HIP: Hawkes Intensity Processes for Social Media Popularity. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. ACM, 735–744.

[28] S. Xiao, J. Yan, X. Yang, H. Zha, and S. M. Chu. 2017. Modeling the Intensity Function of Point Process Via Recurrent Neural Networks. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI'17)*. AAAI Press, 1597–1603.

[29] S. Gao, J. Ma, and Z. Chen. 2015. Modeling and Predicting Retweeting Dynamics on Microblogging Platforms. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining (WSDM '15)*. ACM, 107–116.

[30] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. G. Rodriguez, and L. Song. 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 (KDD '16)*. ACM, 1555–1564.

[31] How Jing and Alexander J. Smola. 2017. Neural Survival Recommender. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM '17)*. ACM, 515–524.

[32] S. Ouyang, C. Li, and X. Li. 2016. A Peek Into the Future: Predicting the Popularity of Online Videos. *IEEE Access* 4, 3026–3033.

[33] Q. Cao, H. W. Shen, H. Gao, J. Gao, and X. Cheng. 2017. Predicting the Popularity of Online Content with Group-specific Models. In *Proceedings of the 26th International Conference on World Wide Web Companion (WWW '17)*. ACM, 765–766.

[34] M. X. Hoang, X. H. D, X. Wu, Z. Yan, and A. K. Singh. 2017. GPOP: Scalable Group-level Popularity Prediction for Online Content in Social Networks. In *Proceedings of the 26th International Conference on World Wide Web Companion (WWW '17)*. ACM, 725–733.

[35] S. Hochreiter and J. Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (Nov.), 1735–1780.

[36] K. Cho, B. V. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *Empirical methods in natural language processing*, 1724—1734.

[37] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. 1988. Neurocomputing: Foundations of Research. MIT Press, Chapter Learning Representations by Back-propagating Errors, 696–699.

[38] P. Bao, H. W. Shen, W. Chen, and X. Cheng. 2013. Cumulative effect in information diffusion: empirical study on a microblogging network. *Plos One* 8, 10, e76027.