

# Modeling and Predicting Retweeting Dynamics on Microblogging Platforms

Shuai Gao                      Jun Ma                      Zhumin Chen  
gao\_shuai@mail.sdu.edu.cn   majun@sdu.edu.cn   chenzhumin@sdu.edu.cn

School of Computer Science and Technology, Shandong University

## ABSTRACT

Popularity prediction on microblogging platforms aims to predict the future popularity of a message based on its retweeting dynamics in the early stages. Existing works mainly focus on exploring effective features for prediction, while ignoring the underlying arrival process of retweets. Also, the effect of user activity variation on the retweeting dynamics in the early stages has been neglected. In this paper, we propose an extended reinforced Poisson process model with time mapping process to model the retweeting dynamics and predict the future popularity. The proposed model explicitly characterizes the process through which a message gain its retweets, by capturing a power-law temporal relaxation function corresponding to the aging in the ability of the message to attract new retweets and an exponential reinforcement mechanism characterizing the “richer-get-richer” phenomenon. Further, we introduce the notation of weibo time and integrate a time mapping process into the proposed model to eliminate the effect of user activity variation. Extensive experiments on two Weibo datasets, with 10K and 18K messages respectively, well demonstrate the effectiveness of our proposed model in popularity prediction.

## Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Online Information Services—*Web-based services*

## General Terms

Algorithm, Measurement

## Keywords

Popularity Prediction; Reinforced Poisson Process; Retweeting Dynamics; Microblogging Platforms

## 1. INTRODUCTION

In recent years, online social networks (OSNs) such as Facebook, Twitter, YouTube etc., have become increasingly popular for their important roles in information sharing and Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

WSDM '15, February 2–6, 2015, Shanghai, China.

Copyright 2015 ACM 978-1-4503-3317-7/15/02 ...\$15.00.

<http://dx.doi.org/10.1145/2684822.2685303>.

interpersonal communication. The prevalence of OSNs has brought along an enormous and ever growing amount of user generated content (e.g., blogs, videos, photos and messages etc.). Recently, the study of predicting the popularity of user generated content has drawn much attention for its remarkable practical value in numerous applications, including trend forecasting [7, 10, 17, 32], understanding the collective human behavior [6, 19, 31], election prediction [28, 30], movie revenue estimation [2] and media advertising [1, 15] etc. Future popularity of user generated content indicates the intensity with which people would react and hence has the potential to influence the policy decisions.

On microblogging platforms, such as Twitter<sup>1</sup> and Weibo<sup>2</sup>, messages gain their popularity as being retweeted [13]. Generally, the popularity prediction problem on microblogging platforms is defined as follows: given a message, predicting its future popularity based on its retweeting dynamics in the early stages, where the popularity of a message is represented by the number of its retweets. To address this problem, considerable efforts have been invested in the past years. In general, current methods fall into the following two main paradigms. One treats the prediction problem as a regression or classification task, making predictions by exploring relevant factors and applying standard regression or classification methods [4, 8, 11, 18]. These methods have been successful in revealing many explicable and effective factors for prediction. However, there are still numerous factors to be considered and these methods are always based on some assumptions or simplification of major factors. The other one treats the retweeting dynamics as time series, making predictions by fitting these time series into certain class of functions [9, 19, 31]. The drawback of these methods is that they model the retweeting dynamics in a mean-field way by focusing on the average amount of retweets received in a fixed time window, ignoring the underlying arrival process of retweets.

Recently, Shen et al. [23] proposed a generative probabilistic model using a *reinforced Poisson process* (RPP) to model explicitly the process through which individual items gain their popularity. The model captures three key ingredients of popularity dynamics: (i) fitness of an item, characterizing its inherent competitiveness against other items; (ii) a general temporal relaxation function corresponding to the aging in the ability to attract new attention; (iii) a reinforcement mechanism characterizing the well-known “richer-get-richer” phenomenon. Despite

<sup>1</sup><http://twitter.com>

<sup>2</sup><http://weibo.com>

its remarkable success in modeling the citation dynamics as reported in [23], there remains some limitations when applied to model the retweeting dynamics. Firstly, unlike the temporal relaxation function which can be adjusted with respect to specific domain, the reinforcement mechanism in RPP model is fixed to be a linear function of received popularity. However, in this paper, we show that this may not hold true for retweeting dynamics. Moreover, the original RPP model does not consider the effect of user activity variation on the popularity dynamics. Generally, users are more active during the daytime than in the midnight. Hence, considering the number of retweets that a message receives in the first hour after been posted, a message posted during daytime will on average receive more retweets than messages posted during midnight. Neglecting the effect of user activity variation may cause the misinterpretation of the relative interestingness of a given message.

In this paper, we extended the general reinforced Poisson process model and addressed the above deficiencies by introducing a power-law temporal relaxation function, an exponential reinforcement function and a time mapping process. Specifically, through analyzing the real data, we found that the temporal relaxation in retweeting dynamics follows a power-law distribution instead of a log-normal distribution as in [23]. Further, observing that the reinforcement effect becomes weaker when the popularity of message grows larger, we proposed a reinforcement function with an exponential decay to capture this phenomenon. Moreover, to eliminate the effect of user variation on the retweeting dynamics, we introduced the notation of *weibo time* and applied a time mapping process to transform the time moments in the retweeting dynamics into weibo time. Putting all these together, we named the proposed model PETM (a reinforced Poisson process model with **P**ower-law relaxation, **E**xponential reinforcement and **T**ime **M**apping process). Finally, we provided a detailed description for the modeling, learning and prediction process of the proposed model.

To evaluate the performance of our proposed model, focusing on the popularity prediction task, we conducted extensive experiments on two Weibo datasets, with 10K and 18K messages respectively. The experimental results show that our proposed model outperforms two baseline models, i.e., SH model [25] and ML model [21], and achieves a remarkable improvement over the original RPP model with a log-normal relaxation function and a linear reinforcement function. Moreover, we show that the prediction performance of a series of RPP models can be significantly improved by integrating the time mapping process. Additionally, we also investigate the inherent characteristics of the parameters of our proposed model.

The novelties and main contributions of this paper are summarized as follows:

- By analyzing real datasets, we reveal that the temporal relaxation which characterizes the aging of a message in retweeting dynamics follows a power-law distribution, and the reinforcement effect declines as the popularity grows which can be captured by a reinforcement function with an exponential decay.
- We introduce the notation of weibo time and propose a time mapping process to eliminate the effect of user activity variation on the retweeting dynamics.

- We present an extended reinforced Poisson process model, namely PETM, by integrating a power-law temporal relaxation function, an exponential reinforcement function and a time mapping process with the original RPP model. Extensive experiments on two Weibo datasets validate the effectiveness of the proposed model in popularity prediction task.

The rest of this paper is organized as follows. We present a review of related work in Section 2. The preliminaries including the formal definition of popularity prediction problem and the description of general reinforced Poisson process model are provided in Section 3. We give a detailed description for the modeling, learning and prediction process of the proposed model in Section 4. In Section 5, we describe the datasets, comparison methods and evaluation metrics used in our experiments. The experimental results and discussion are presented in Section 6. We conclude our paper in Section 7.

## 2. RELATED WORK

Popularity prediction of user generated content has been widely studied in the literature. Recent studies on micro-blogging platforms include popularity prediction for hashtag (or trend) and individual message. Typical solutions include the regression models [4, 12, 27], classification models [8, 10, 11, 18, 16] and time series models [19, 31, 32]. For regression or classification models, previous works mainly focus on exploring effective features for prediction. As an initial effort, Bongwon et al. [24] examined a number of features that might affect retweetability of tweets and found that content features such as URLs and hashtags have strong relationship with retweetability. Tsur et al. [27] employed a regression model to predict hashtag frequency on a weekly basis based on 25-week Twitter data. The features used in their experiments are mainly extracted from the hashtag itself. Hong et al. [11] treated the prediction problem as a classification task and investigated a wide spectrum of features including content features derived from the message, context features (structural features) extracted from the underlying user network and temporal features obtained from the observed popularity dynamics. Similar features have also been investigated in [18] and [8]. Generally speaking, the context features are more effective than content features, while the prediction performance can be further improved when combining the temporal features. For time series models, Matsubara et al. [19] proposed SpikeM, which used the self-excited Hawkes conditional Poisson process to model the power-law relaxation of popularity dynamics. Despite of its promising predictive power, the dependence on exogenous factors and the fixed power-law relaxation function limit its generalization ability.

Recently, there have been pioneering data-driven analysis of popularity prediction on different kinds of user generated content, e.g., news articles [3, 26], YouTube videos [1, 7, 15, 21, 25], diggs in Digg [1, 14, 25], posts in Facebook [5] and papers [23, 29] etc. Focusing on diggs in Digg and videos in YouTube, Szabo et al. [25] found a high log-linear correlation existed between the popularity on early days and later days, and presented three regression models for prediction. Following [25], Pinto et al. [21] proposed a multivariate linear regression model which split the training period into several equal intervals, and predicted the future

popularity of a video as a linear function of views count in every interval.

The most close research to ours was conducted by Shen et al. [23]. They proposed the general framework of reinforced Poisson process model and applied the model to predict the future citations of a given paper. Inspired by this study, in this paper, we proposed a extended reinforced Poisson process model to characterize the retweeting dynamics.

### 3. PRELIMINARIES

#### 3.1 Problem Statement

In this work, we aim to model the retweeting dynamics of a single message in the early stages and use the model to predict its popularity in the future. For a given message  $m$ , we use the number of “retweets” to measure its popularity. Then, we sort all its retweets according to their post time in ascending order, forming a chain of messages. The post time for the  $k$ th retweet is denoted by  $\mathcal{T}_k^m$ . Specially, we use  $\mathcal{T}_0^m$  to denote the post time of the original message. The post time  $\mathcal{T}$  is measured in Unix timestamp which is a number indicating the seconds which have elapsed since 1970-01-01 00:00:00, UTC. Further, we define the *age* of  $k$ th retweet  $t_k^m$  as the elapsed time from  $\mathcal{T}_0^m$ , which can be calculated by  $t_k^m = \mathcal{T}_k^m - \mathcal{T}_0^m$ . The number of retweets at age  $t$  is denoted by  $N(t)$ . During the modeling and prediction process, following the definition in [25], we call *reference time*  $T_r$  the time when we intend to predict the popularity of a message whose age with respect to the post time is  $T_r$ . Also, by *indicator time*  $T_i$  we refer to the age the message when we perform the prediction, namely that how long we can observe the retweeting history in order to extrapolate. Generally, we have  $T_i < T_r$ . Intuitively,  $T_i$  measures the length of the training period for the model. The retweeting dynamics of message  $m$  up to  $T_i$  is characterized by a set of time moments  $\{t_k^m\} (1 \leq k \leq n_m)$  when each retweet arrives, where  $n_m$  is the total number of retweets in this training period. Formally, the prediction problem can be described as: for a message  $m$ , given its retweeting dynamics  $\{t_k^m\}$  up to the indicator time  $T_i$ , predicting its popularity at the reference time  $T_r$ .

#### 3.2 Reinforced Poisson Process Model

The reinforced Poisson process (RPP) model was first proposed in [29, 23] to model the stochastic popularity dynamics for items in a complex evolving system. The model captures three key ingredients simultaneously: (i) fitness of an item, characterizing its inherent competitiveness against other items; (ii) a general temporal relaxation function corresponding to the aging in the ability to attract new attention; (iii) a reinforcement mechanism characterizing the well-known “richer-get-richer” phenomenon. In the following, with respect to the retweeting dynamics on microblogging platforms, we give a more general definition of RPP model.

Given a message  $m$ , its retweeting dynamics  $\{t_k^m\}$  up to  $T_i$  can be modeled as reinforced Poisson process which is characterized by the rate function  $\lambda_m(t, k)$  as

$$\lambda_m(t, k) = c_m f_m(t) r_m(k) \quad (1)$$

where  $c_m$  is intrinsic attractiveness of the message,  $f_m(t)$  is the general temporal relaxation function which characterizes

the aging effect, and  $r_m(k)$  is the general reinforcement function depicting the “richer-get-richer phenomenon”. Instead of defining  $r_m(k)$  to be the number of retweets as in [29, 23], we generalize the reinforcement function to be any piecewise constant function which is defined on number of retweets  $k$ , i.e., during the time interval between  $k$ th and  $(k+1)$ th retweet,  $r_m(k)$  stays unchanged.

Next, we describe the modeling process of general RPP model. Given the  $(k-1)$ th retweet arrives at  $t_{k-1}^m$ , the probability that the  $k$ th retweet arrives at  $t_k^m$  follows

$$p_1(t_k^m | t_{k-1}^m) = c_m f_m(t_k^m) r_m(k-1) e^{-\int_{t_{k-1}^m}^{t_k^m} c_m f_m(s) r_m(k-1) ds} \quad (2)$$

while the probability that no retweet arrives between  $t_{n_m}^m$  and  $T_i$  is

$$p_0(T_i | t_{n_m}^m) = e^{-\int_{t_{n_m}^m}^{T_i} c_m f_m(s) r_m(n_m) ds} \quad (3)$$

The RPP model assumes that retweets from different time intervals are statistically independent. Hence by incorporating Equation 2 and Equation 3, the likelihood of the observing retweeting dynamics  $\{t_k^m\}$  up to  $T_i$  follows

$$\mathcal{L} = p_0(T_i | t_{n_m}^m) \prod_{k=1}^{n_m} p_1(t_k^m | t_{k-1}^m) \quad (4)$$

By taking logarithm, we can get the log-likelihood function  $\ell$  as

$$\begin{aligned} \ell &= \ln \mathcal{L} = \ln \left( p_0(T_i | t_{n_m}^m) \prod_{k=1}^{n_m} p_1(t_k^m | t_{k-1}^m) \right) \\ &= n_m \ln c_m + \sum_{k=1}^{n_m} \left( \ln f_m(t_k^m) + \ln r_m(k-1) \right) - c_m X_m \end{aligned} \quad (5)$$

where

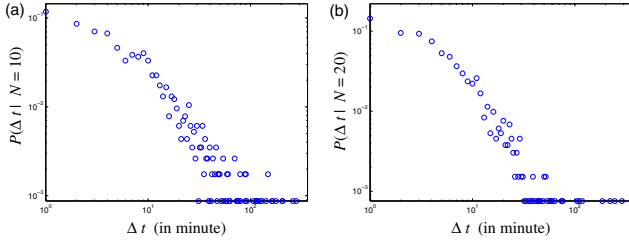
$$X_m = \sum_{k=1}^{n_m} \int_{t_{k-1}^m}^{t_k^m} f_m(s) r_m(k-1) ds + \int_{t_{n_m}^m}^{T_i} f_m(s) r_m(n_m) ds \quad (6)$$

### 4. MODEL

In this section, we first give the specific forms for temporal relaxation function and reinforcement function with respect to retweeting dynamics. Then we describe the hourly variation of user activity and the proposed time mapping process. Lastly, we give a detailed description for the proposed model, including the modeling, learning and prediction process.

#### 4.1 Temporal Relaxation Function

As shown in the rate function (Equation 1), the retweeting dynamics of a message is controlled by three independent forces, which are difficult to separate from each other. Hence to determine the specific form of temporal relaxation function, we need to control the other factors, isolating the temporal decay. Specifically, we group messages with same fitness  $c$  and same reinforcement  $r(k)$ , and look at the time when they were retweeted again. Since we do not know  $c$  beforehand, we roughly consider messages which are posted in a fixed time period and receive the same number of retweets in one hour after being posted as having the same fitness. Note that the reinforcement  $r(k)$  is a function of the number of retweets  $k$ , by selecting messages with



**Figure 1: Empirical validation of power-law decay**

same retweet number, the reinforcement is also controlled. Therefore we selected messages posted between 10am and 11am with fixed cumulative retweet number  $N$  in the first hour after being posted, and tracked the moment when their retweets number changed from  $N$  to  $N + 1$ . Then we measured  $\Delta t$  in minutes, i.e., minutes passed when  $N \rightarrow N + 1$  took place. Note that all values of  $\Delta t$  are greater than 60 since the messages have been posted for more than 1 hour before the  $(N + 1)$ th retweet happens. Hence, to illustrate the distribution of  $\Delta t$  more clearly, we subtract 60 from  $\Delta t$  given that subtracting a constant from a random variable will not change its distribution. Then we show  $P(\Delta t|N)$  when  $N$  is 10 and 20 in Figure 1. We can easily find that the temporal relaxation function is best approximated by a power-law function

$$f_m(t_k) = t_k^{-\gamma_m} \quad (\gamma > 0) \quad (7)$$

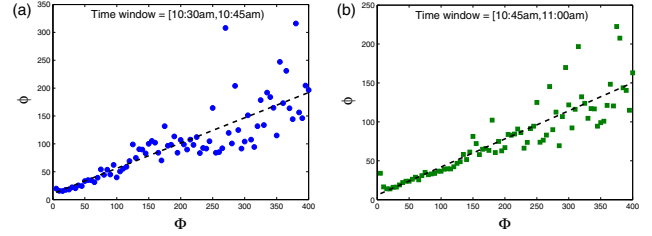
The power-law decay of influence has also been reported in blogs [20] and videos [6].

## 4.2 Reinforcement Function

In this section, we study the reinforcement mechanism in retweeting dynamics and address the following issues: (1) whether the reinforcement mechanism exists in retweeting dynamics, i.e., whether the previously retweets can trigger more subsequent retweets? (2) if the reinforcement mechanism does exist, how to model it? To answer the first question, we follow the similar method as proposed in [22], which is to measure the preferential attachment mechanism from the empirical data, one keeps a constant time window and looks at the growth of degrees as a function of existing degree. Similarly, here in the analysis of retweeting dynamics, we first fix a constant time window, and then measure the number of retweets that a message received before and within this time window. Specifically, given a fixed time window, we use  $\Phi$  to denote the number of retweets that each message received before this time window, and use  $\phi$  to measure the average number of retweets for each message with  $\Phi$  retweets received in this time window. In Figure 2, we show how  $\phi$  varies with  $\Phi$  when the time window is fixed to be 10:30am–10:45am (Figure 2(a)) and 10:45am–11:00am (Figure 2(b)). Generally speaking, for both examined time window,  $\phi$  is linearly proportional to  $\Phi$ , i.e.,  $\phi \propto \Phi$ . The dashed lines correspond to the best linear fitting. This is reasonable since this linearity can lead to a power-law message popularity distribution as shown in Figure 4. In previous literature [23, 29], a simple linear function is used to capture this linear reinforcement

$$r_m(k) = k + 1 + \epsilon \quad (8)$$

where  $k \geq 0$  and  $\epsilon$  denotes the effective number of retweets which plays the role of prior belief. The linearly



**Figure 2: Empirical validation of reinforcement mechanism**

reinforcement function is defined based on the assumption that all retweets (including the original message) are created equally. Therefore, we can rewrite Equation 8 as  $r_m(k) = \sum_{j=0}^k 1 + \epsilon$ , where in the summation  $\sum_{j=0}^k 1$  each retweet contributes 1 to the reinforcement function as assumed. However, by further examining Figure 2, we can find that when  $\Phi$  grows larger, a large amount of  $\phi$  fall below the dashed line, indicating that  $\phi$  grows slower than linearly with  $\Phi$ . One possible reason for this phenomenon is that the retweets are not created equally and with the number of retweets grows larger, the contribution of the subsequent retweets to the reinforcement function become smaller. Hence, to capture the decay of contribution for each retweet, we assume that the contribution decay follows an exponential distribution, i.e., for the  $j$ th retweet, the contribution to the reinforcement function  $r_m(k)$  is  $e^{-\alpha_m j}$ , where  $\alpha_m > 0$ . By adding the contribution of all retweets together, we get the modified reinforcement function as follows

$$r_m(k) = \sum_{j=0}^k e^{-\alpha_m j} + \epsilon = \frac{\epsilon - e^{-\alpha_m(k+1)}}{1 - e^{-\alpha_m}} \quad (\alpha_m > 0) \quad (9)$$

where  $\epsilon = 1 + \epsilon - \epsilon e^{-\alpha_m}$ . Henceforth, we call Equation 9 the exponential reinforcement function.

## 4.3 User Activity and Time Mapping

In this section, we examine another important factor, user activity, which is not considered by the original RPP model. To investigate the hourly variation of user activity, we randomly selected about 10 thousand users from Weibo and crawled all their messages posted between 2013-07-01 00:00:00 and 2013-07-31 23:59:59, which results in 1.6 million messages. We first show the average number of messages posted during a given hour as a function of time in Figure 3(a). Clearly, we can see that user activity varies over time, namely that users are highly active during daytime and inactive in midnight. Hence, depending on the time of day when messages are posted, the number of initial retweets that messages get will differ greatly, as illustrated in Figure 3(b), where we show the average number of retweets that a message receives in the first hour after being posted (blue curve). As can be expected, messages posted at higher active periods of a day will on average receive more retweets in the first hour than messages posted at less active periods. For example, a message posted at 11am can on average get 25 retweets in the first hour, while it will only get 5 retweets if it is posted at 4am. Therefore, if we do not eliminate the effect of user activity variation, we may misinterpret the relative interestingness of a message only based on the observation made in a few hours after it has been posted.

To eliminate the effect of user activity variation, we introduce the notation of *weibo time*, where we measure time not by wall time (seconds), but by the number of messages that users post on Weibo. To map the wall time into weibo time, we first use  $\tau$  and  $\hat{\tau}$  to denote the second in a day, which are measured by wall time and weibo time respectively. Both  $\tau$  and  $\hat{\tau}$  take values in  $[0, 86400)$  where 86400 is the number of seconds in a day. The difference between them is that  $\tau$  can only be integer while  $\hat{\tau}$  can be decimal. Then through analyzing the dataset, we get the average number of messages posted per second  $M^*$  and in each second  $M_\tau$ .  $M^*$  is a constant number while  $M_\tau$  varies with  $\tau$ . To be clear, we show  $M_\tau$  as a function of  $\tau$  in Figure 3(c). Obviously,  $M_\tau$  takes larger value in daytime than in midnight, which can further illustrate the hourly variation of user activity. Now, we can define the mapping function  $g$  as

$$\hat{\tau} = g(\tau) = \sum_{j=0}^{\tau} M_j / M^* \quad (10)$$

The image of mapping function  $g(\tau)$  is shown in Figure 3(d) (blue curve). Next, for each original timestamp  $\mathcal{T}$  in dataset, we denote the corresponding mapped timestamp as  $\hat{\mathcal{T}}$ . To map from  $\mathcal{T}$  to  $\hat{\mathcal{T}}$ , we first need to transform  $\mathcal{T}$  into the corresponding second of day  $\tau_*$  by

$$\tau_* = \text{mod}(\mathcal{T} + 8 * 3600, 86400) \quad (11)$$

where  $\text{mod}(a, b)$  returns the modulus after division of  $a$  by  $b$  and 3600 is the number of seconds in an hour. Since Weibo locates in China, UTC+8 time zone, we need to add  $8 * 3600$  to each  $\mathcal{T}$  before the modulo operation. Lastly, by incorporating Equation 11 and Equation 10, we can get the mapped timestamp  $\hat{\mathcal{T}}$  by

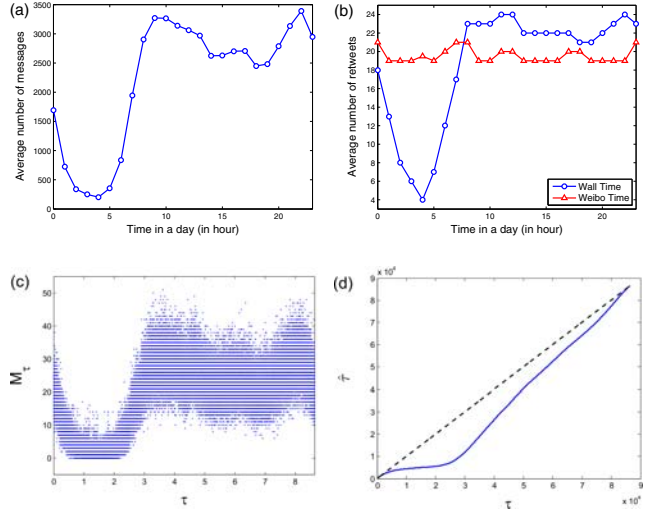
$$\hat{\mathcal{T}} = \mathcal{T} - \tau_* + \hat{\tau}_* = \mathcal{T} - \tau_* + g(\tau_*) \quad (12)$$

After timestamp mapping, we show the average number of retweets that a message receives in the first weibo hour after being post in Figure 3(b) (red curve). As can be seen, the average retweets number slightly changes over the post time, indicating that the mapping from wall time to weibo time can largely mitigate the impact of message's post time on its early-stage popularity and eliminate the effect of user activity variation.

#### 4.4 Modeling Learning and Prediction

In this section, we describe in detail how to model the retweeting dynamics for a given message under the reinforced Poisson process framework with power-law temporal relaxation function (Section 4.1) and exponential reinforcement function (Section 4.2). Also, we show how to integrate the time mapping process (Section 4.3) into the reinforced Poisson process framework. In the following, we describe the model learning and prediction process step by step, and we start with the input of the model.

**Input time moments.** As defined in Section 3.1, for a given message  $m$ , its retweeting dynamics can be characterized by a set of non-decreasing time moments  $\{t_k^m\}$ , where  $t_k^m$  denotes the age of the  $k$ th retweet and  $t_0^m$  is the age of the original message which is 0. The drawback of this definition is that since the temporal relaxation function  $t^{-\gamma m}$  takes a power-law form, it will be infinity when  $t = t_0^m = 0$ . To avoid this, we manually add a constant value  $t^*$  to all  $t_k^m$ .



**Figure 3: User activity variation and Time mapping**

Formally, we redefine  $t_k^m$  as

$$t_k^m = \mathcal{T}_k^m - \mathcal{T}_0^m + t^* \quad (13)$$

Also, henceforth in the modeling and prediction process, when we refer to indicator time  $T_i$  and reference time  $T_r$ , their values are  $T_i + t^*$  and  $T_r + t^*$  respectively.

Further, to integrate the time mapping function into the RPP model, we first need to map the original timestamps  $\{\mathcal{T}_k^m\}$  into  $\{\hat{\mathcal{T}}_k^m\}$  using Equation 12, and then get  $t_k^m$  as

$$t_k^m = \hat{\mathcal{T}}_k^m - \hat{\mathcal{T}}_0^m + t^* \quad (14)$$

Here, we use time series  $\{t_i^m\}$  up to  $T_i$  as the input of the modeling process. For ease of calculation of the power-law temporal relaxation function, we measure  $t_k^m$  in hours and set  $t^* = 1$  for all the messages. For simplicity and clarity, henceforth we will omit the superscript  $m$  which is used to represent message  $m$ .

**Model Formulation.** By substituting the power-law temporal relaxation function Equation 7 and the exponential reinforcement function Equation 9 into the general rate function Equation 1, we can get the specific form of rate function for retweeting dynamics as

$$\lambda(t_k, k) = \frac{ct_k^{-\gamma}(\varepsilon - e^{-\alpha(k+1)})}{1 - e^{-\alpha}} \quad (15)$$

Then, by substituting Equations 7 and 9 into the general log-likelihood function Equation 5, we can get the log-likelihood for the retweeting dynamics  $\{t_k\}$  up to  $T_i$  as

$$\begin{aligned} \ell &= n \ln c + \sum_{k=1}^n \left( \ln f(t_k) + \ln r(k-1) \right) - cX \\ &= n \ln \frac{c}{1 - e^{-\alpha}} + \sum_{k=1}^n \left( \ln(\varepsilon - e^{-\alpha k}) - \gamma \ln t_k \right) - cX \end{aligned} \quad (16)$$

where

$$X = \sum_{k=1}^{n+1} \frac{(t_k^{1-\gamma} - t_{k-1}^{1-\gamma})(\varepsilon - e^{-\alpha k})}{(1-\gamma)(1 - e^{-\alpha})} \quad (17)$$

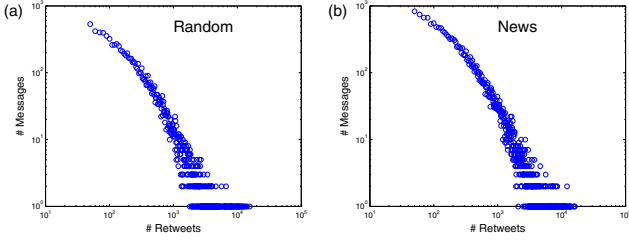


Figure 4: Popularity distribution

Note that we manually set  $t_{n+1} = T_i$  for the sake of simplification. It does not mean that the  $(n+1)$ th retweets arrives at  $T_i$ .

**Parameter Estimation.** Here, we denote the optimal values for parameters  $\{c, \gamma, \alpha\}$  as  $\{c^*, \gamma^*, \alpha^*\}$ . For the fitness parameter  $c$ , we can get its optimal value  $c^*$  in closed form by maximizing the log-likelihood function in Equation 16 as

$$c^* = \arg \max_c \ell = n/X \quad (18)$$

For parameters  $\gamma$  and  $\alpha$ , the optimal values can be found by maximizing the log-likelihood  $\ell$  using the gradient ascent method. The gradients for each parameter are

$$\frac{\partial \ell}{\partial \gamma} = - \sum_{k=1}^n \ln t_k - c \frac{\partial X}{\partial \gamma} \quad (19)$$

$$\frac{\partial \ell}{\partial \alpha} = - \frac{ne^{-\alpha}}{1 - e^{-\alpha}} + \sum_{k=1}^n \frac{e^{-\alpha} + ke^{-\alpha k}}{\varepsilon - e^{-\alpha k}} - c \frac{\partial X}{\partial \alpha} \quad (20)$$

where

$$\frac{\partial X}{\partial \gamma} = \sum_{k=1}^{n+1} \frac{\varepsilon - e^{-\alpha k}}{1 - e^{-\alpha}} \left( \frac{t_{k-1}^{1-\gamma} \ln t_{k-1} - t_k^{1-\gamma} \ln t_k}{1 - \gamma} + \frac{t_k^{1-\gamma} - t_{k-1}^{1-\gamma}}{(1 - \gamma)^2} \right) \quad (21)$$

$$\frac{\partial X}{\partial \alpha} = \sum_{k=1}^{n+1} \frac{(t_k^{1-\gamma} - t_{k-1}^{1-\gamma})(ke^{-\alpha k} - (k-1)e^{-\alpha(k+1)} - e^{-\alpha})}{(1 - \gamma)(1 - e^{-\alpha})^2} \quad (22)$$

Following the standard gradient ascent method, update rules at the  $j$ th iteration are shown in the following equations.

$$\gamma^{(j+1)} = \gamma^{(j)} + \beta_1 \cdot \frac{\partial \ell}{\partial \gamma} \quad (23)$$

$$\alpha^{(j+1)} = \alpha^{(j)} + \beta_2 \cdot \frac{\partial \ell}{\partial \alpha} \quad (24)$$

where  $\beta_1$  and  $\beta_2$  are the learning rate at each iteration. The algorithm stops when the change in an iteration is small enough.

**Prediction.** With the obtained optimal parameters, i.e.,  $c^*$ ,  $\gamma^*$  and  $\alpha^*$ , we can use the model to predict the expected number of retweets  $N(t)$  at any given time moment  $t$ . Incorporating with the rate function of retweeting dynamics in Equation 15, we treat the prediction task as the following differential equation

$$\frac{dN(t)}{dt} = c^* t^{-\gamma^*} \frac{\varepsilon - e^{-\alpha^*(N(t)+1)}}{1 - e^{-\alpha^*}} \quad (25)$$

with the boundary condition  $N(T_i) = n$ . By solving this differential equation, we get the prediction function

$$N(t) = (\ln(1 + e^Y) - Y - \ln \varepsilon - \alpha^*) / \alpha^* \quad (26)$$

where

$$Y = \frac{\varepsilon c^* \alpha^* (T_i^{1-\gamma^*} - t^{1-\gamma^*})}{(1 - \gamma^*)(1 - e^{-\alpha^*})} - (n+1)\alpha^* - \ln(\varepsilon - e^{-\alpha^*(n+1)}) \quad (27)$$

## 5. EXPERIMENTAL SETUP

### 5.1 Datasets

Since this paper focuses on modeling the retweeting dynamics and predicting the future popularity of a given message on microblogging platforms, we conducted extensive experiments on two datasets of Weibo messages. Each dataset contains a set of original messages posted during July 1–31, 2013. For each message, we recorded the post timestamps for it and all its retweets in one month after been posted. We adopted two different crawling strategies and obtained the following two datasets.

- *Random*: To construct the random dataset, we first randomly selected 10,000 seed users and crawled all their original messages posted during July 1–31 2013. The crawling process produced in total 0.8 million original messages. Then we manually removed messages with more than 20,000 retweets since the crawling process for their retweets is too time consuming. Also, messages with less than 50 retweets are removed since they usually stop receiving more retweets in the first few hours. Thus we randomly selected 10K messages from the rest of the crawled messages to construct the random dataset. In this dataset, the mean and median values for message popularity are 450 and 210 respectively.
- *News*: In this dataset, we concentrated on a specific type of message—news. To begin with, we manually selected 25 prestigious news accounts and crawled all their original posts during July 1–31, 2013. Similarly with Random dataset, messages with retweets number larger than 20,000 or less than 50 are neglected. The resulting dataset contains in total 18K original messages and its statistics is shown in Table 1. As can be seen, since the number of followers for all these news account are larger than 1 million, their posted messages on average gain more retweets than these in Random dataset. For News dataset, the mean and median values for message popularity are 529 and 238 respectively.

Since few messages can receive more retweets after been posted for one month, we use the number of retweets that a message gets in the first month since posted to measure its final popularity. We show the distribution of the message final popularity for both datasets in Figure 4. As can be seen, both distributions follow power-law.

### 5.2 Comparison Methods

To examine the efficiency of our modified RPP models, two variations of RPP model and two widely used regression models were introduced for comparison. Specifically, the comparison methods in our experiments are listed as follows:

- **SH** [25]: the linear regression method for logarithmic popularity, which is proposed by Szabo and Huber-



**Table 1: Statistics of News Dataset**

Account Name	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13
# Follower <sup>1</sup>	3.8	9.2	10.8	3.3	11.1	5.1	4.2	9.2	7.2	9.4	5.9	6.0	6.3
# Message	1062	232	1164	77	1373	239	301	1246	1519	672	318	922	577
Avg. # Retweet	113.3	653.8	741.1	170.0	331.4	453.8	339.4	797.6	315.8	234.7	323.3	380.4	751.7
Account Name	A14	A15	A16	A17	A18	A19	A20	A21	A22	A23	A24	A25	
# Follower	3.7	9.4	3.8	9.6	0.7	4.8	5.9	16.0	1.4	20.8	22.2	5.6	
# Message	1843	1922	158	178	693	1685	1103	1143	383	1381	972	830	
Avg. # Retweet	118.1	143.1	772.3	1479.5	42.0	410.1	362.1	378.9	499.8	1543.3	1887.4	168.7	

<sup>1</sup> # Follower is measured in million.

man. The SH model is represented by

$$\ln N(T_r) = a_1 \ln N(T_i) + a_0 \quad (28)$$

- **ML** [21]: the multivariate linear regression model. The training period  $[0, T_i]$  is split equally into  $p$  intervals and the number of retweets received in each interval is seen as popularity *deltas*. The ML model predicts the popularity of a message at  $T_r$  as a linear function of  $p$  popularity deltas. Here, we set  $p = 6$ .
- **LL** [23]: RPP model with a log-normal relaxation function as in Equation 29 and a linear reinforcement function as in Equation 8. This model has been used to modeling the citation dynamics.

$$f_m(t; \mu_m, \sigma_m) = \frac{1}{\sqrt{2\pi}\sigma_m t} e^{-\frac{(\ln t - \mu_m)^2}{2\sigma_m^2}} \quad (29)$$

- **PL**: RPP model with a power-law relaxation function as in Equation 7 and a linear reinforcement function as in Equation 8.
- **PE**: RPP model with a power-law relaxation function as in Equation 7 and an exponential reinforcement function as in Equation 9.

Note that we do not employ the LE model, i.e. RPP model with a log-normal relaxation function and an exponential reinforcement function, for comparison since the modeling and prediction process for it is far too complicated to be implemented. Moreover, the input for the RPP models can be with or without the time mapping as in Equation 12. In order to distinguish them, we add **TM** to each RPP models' name. The time mapping version of RPP models applied for comparison consist of LLTM, PLTM and PETM.

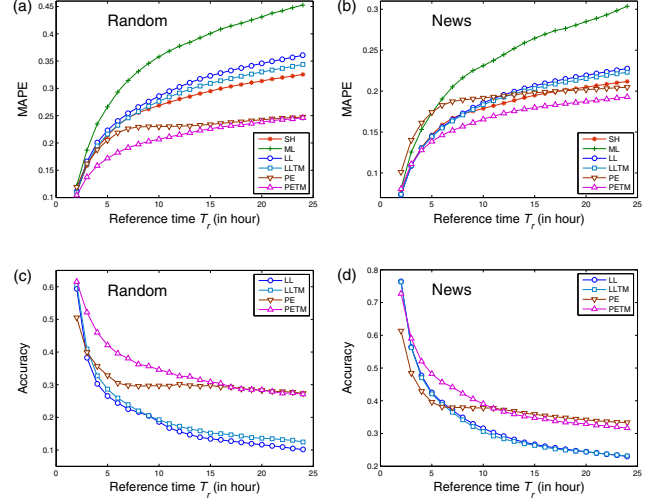
### 5.3 Evaluation Metrics

The same as in [23], we use Mean Absolute Percentage Error (*MAPE*) and *Accuracy* to evaluate the performance of all the prediction models. Let  $N_m(t)$  be the real number of retweets for message  $m$  up to time  $t$ , and  $\hat{N}_m(t)$  be the predicted number of retweets. The MAPE measures the average deviation between predicted and real popularity over an aggregation of messages. For a dataset of  $M$  messages, the MAPE is defined as

$$MAPE = \frac{1}{M} \sum_{m=1}^M \left| \frac{\hat{N}_m(t) - N_m(t)}{N_m(t)} \right| \quad (30)$$

The Accuracy measures the fraction of messages correctly predicted for a given error tolerance  $\zeta$ . It is defined as

$$Accuracy = \frac{1}{M} \sum_{m=1}^M \mathbb{I} \left[ \left| \frac{\hat{N}_m(t) - N_m(t)}{N_m(t)} \right| \leq \zeta \right] \quad (31)$$



**Figure 5: Prediction results for different models**

where  $\mathbb{I}[X]$  is an indicator function which returns 1 if the statement  $X$  is true and 0 otherwise. Here, we set the threshold  $\zeta = 0.1$  in this paper.

## 6. EXPERIMENTAL RESULTS

In this section, we first report extensive experimental results on both Random and News datasets to validate the effectiveness of our proposed PE and PETM models. Then we give detailed analysis of the model parameters.

### 6.1 Prediction Results

We begin our analysis by comparing the prediction performance of different models on Random and News datasets. Specifically, by fixing the indicator time  $T_i$  to be 1, we evaluate the prediction performance of our proposed PE and PETM models as well as four comparison models including SH, ML, LL and LLTM, with reference time  $T_r$  varying from 2 to 24. The parameter  $\epsilon$  is set to be 10 for all RPP models. Figure 5 shows the comparison results. Generally speaking, the proposed PETM model can always achieve the best prediction performance for each  $T_r$  on both datasets. When removing the time mapping process, the PE model can also show better performance than the other four models on Random dataset as shown in Figure 5(a) and 5(c). However, on News dataset in Figure 5(b) and 5(d), the PE model shows worse performance than the other models except ML model in terms of MAPE when  $T_r$  is smaller than 10, and it begins to show better performance when  $T_r$  grows larger. That indicates the PE model can well predict the long-term popularity of a given message but it

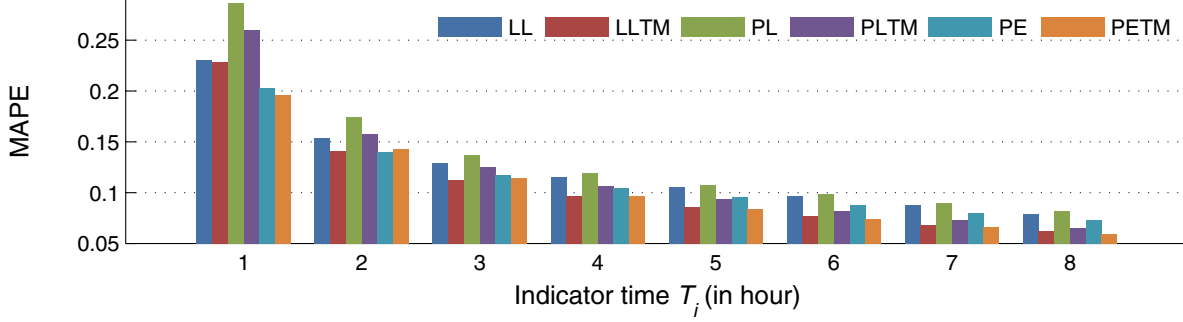


Figure 6: The performance comparison for different RPP models on News dataset with the variation of  $T_i$

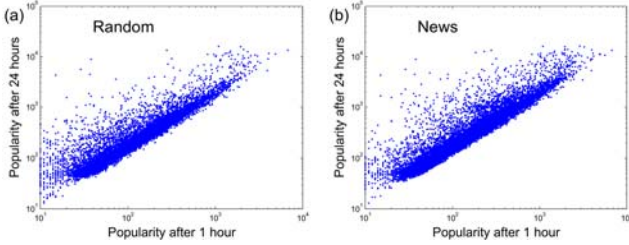


Figure 7: Popularity Correlation

cannot well capture the early evolution of the retweeting dynamics. Also, by comparing LL and LLTM models with PE and PETM models, we find that the performance of PE models are better than LL models in terms of both measurements on both datasets and the superiority increases with  $T_r$ , which means the proposed PE models are more appropriate to model the retweeting dynamics on microblogging platforms. For the other two baseline methods, SH model performs better than LL models in terms of MAPE and ML model shows the worst performance on both datasets. Both of them are regression method and try to capture the correlation between the early popularity and future popularity. The ML model works linearly and the SH model works in a logarithmic manner. As can be seen from Figure 7, the early popularity and future popularity is logarithmically correlated. That explains why the SH model can yield a moderate MAPE value while the ML model cannot achieve a better prediction performance. However, both of them show poor performance in terms of Accuracy. We do not show the accuracy for these two method since they are all equal to zero when  $T_i \geq 2$ .

Further, we carried out extensive experiments on News dataset to examine the following issues: (i) the prediction performance of different RPP models when  $T_i$  varies; (ii) the effect of  $T_i$  on the prediction performance of RPP models; (iii) the effect of time mapping process on the prediction performance of RPP models. To be specific, we apply a series of RPP models, including LL, PL and PE models, with and without the time mapping process, on News dataset with the indicator time  $T_i$  varying from 1 to 8. Since most messages in dataset stop receiving more retweets after being posted for 24 hours, we fix the reference time  $T_r$  to be 24 to check the ability for different models in predicting the popularity of a message. We use MAPE to measure the prediction performance and set the parameter  $\epsilon$  again to be 10. The experimental results are shown in Figure 6.

First, we examine the prediction performance of different RPP models when  $T_i$  varies. For fair comparison, we compare the performance of RPP models with and without the time mapping process separately. As can be seen from Figure 6, for models without time mapping process, the PE model performs the best on the entire range of  $T_i$ , and the PL model shows the worst performance. The same observation can be made for models with time mapping process, except that when  $T_i = 2$  and 3, the performance of PETM is slightly worse than LLTM. Generally speaking, the superior predictive power of PE models indicates that the retweeting dynamics can be better characterized by integrating the power-law temporal decay function with the exponential reinforcement function. Also, we can find that the exponential reinforcement function plays a very important role in PE models since when the reinforcement function is replaced by a linear function, as in PL models, the prediction performance drops significantly.

Then, we examine the effect of  $T_i$  on the prediction performance of RPP models. Intuitively,  $T_i$  is the length of training period. As shown in Figure 6, for all RPP models, the MAPE decreases as  $T_i$  increases, indicating that increasing the training period can improve the prediction performance for all the models. However, we can also see that the rate at which MAPE declines slows down quickly, e.g., the rate of decline for PETM model drops from 26.9% at  $T_i = 1$  to 10.7% at  $T_i = 8$ . This means the marginal gain for performance improvement diminishes with the increasing of the training period. This is caused by that the amount of training data added to the model decreases as  $T_i$  increases, namely that, when  $T_i$  becomes larger, the number of retweets that a message receives in the  $T_i$ th hour is generally smaller than it receives in the  $(T_i - 1)$ th hour. Though  $T_i$  is increased by one hour at each time, the amount of training data does not grow proportionally.

Finally, we examine the effect of time mapping process on the prediction performance of all models. As we can see from Figure 6, for a fixed  $T_i$ , comparing with models without the time mapping, the MAPE drops significantly when adding the time mapping process to each model. This result confirms that the use activity variation can affect the retweeting dynamics and harm the performance of prediction models. By integrating the time mapping process to each prediction model, we can eliminate the effect of use activity variation and improve the prediction performance. Moreover, we observe that when  $T_i$  grows larger, the decrease in MAPE becomes more evident for



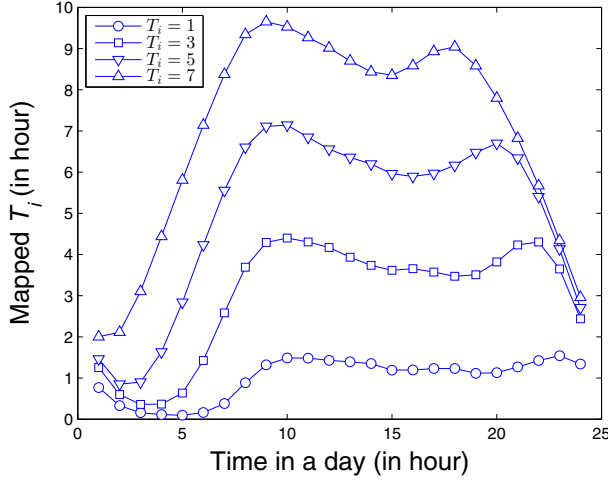


Figure 8: Variation of mapped indicator time  $T_i$

all models. The reason for this result is that the effect of user activity variation becomes more obvious with the increasing of  $T_i$ . To be more clear, in Figure 8 we show the variation of mapped  $T_i$  with the posted time of message varying from 1h to 24h. As we can see, the magnitude of change for the mapped  $T_i$  becomes larger when  $T_i$  increases. For example, the minimum value and maximum value of the mapped  $T_i$  are 0.09 and 1.54 when  $T_i = 1$ , while they are 2.00 and 9.65 when  $T_i = 7$ . That means, with the length of training period grows larger, the effect of user activity variation becomes more evident. Hence, the improvement in prediction performance by integrating the time mapping process becomes more obvious. To sum up, the experimental results certify the effectiveness of the proposed time mapping process.

## 6.2 Analysis of Model Parameters

In our proposed PE and PETM models, there are in total four parameters  $\{\epsilon, c, \gamma, \alpha\}$ , where  $\epsilon$  is set manually and the others are derived from the model learning process. In this section, we first examine the effect of parameter  $\epsilon$  on the prediction performance of the proposed models, and then investigate the inherent characteristics of the learned parameters.

As defined in Section 4.2, the parameter  $\epsilon$  is the effective number of retweets which plays the role of prior belief. Here, to examine the effect of  $\epsilon$ , by fixing  $T_i = 1$  and  $T_r = 24$ , we conducted experiments on News dataset to check the prediction performance of LL, LLTM, PE and PETM models with  $\epsilon = 5, 10, 15, 20$ . The results are shown in Table 2. As can be seen, we can draw the similar conclusion as in [23] that decreasing  $\epsilon$  can reduce MAPE for all four models. Since a larger  $\epsilon$  implies a weaker role of the reinforcement mechanism in RPP models, this result indicates the disparity in retweet numbers is appropriately captured by the reinforcement mechanism. Moreover, we find that if  $\epsilon$  is too small, e.g.  $\epsilon = 5$ , the MAPE becomes larger again. This result implies that  $\epsilon$  should not be set too small since it balances the strength in the reinforcement mechanism. Overall speaking, the experimental results certify the importance of reinforcement mechanism in modeling the retweeting dynamics.

Further, taking PETM model on News dataset as an example, we show the distribution of the learned parameters.

Table 2: Effect of prior number of retweets  $\epsilon$

Parameter	Methods <sup>1</sup>			
	LL	LLTM	PE	PETM
$\epsilon = 5$	0.2324	0.2286	0.2109	0.1944
$\epsilon = 10$	0.2274	0.2230	0.2046	0.1928
$\epsilon = 15$	0.2848	0.2557	0.2285	0.2168
$\epsilon = 20$	0.2928	0.2860	0.2510	0.2399

<sup>1</sup> The performance is measured in MAPE.

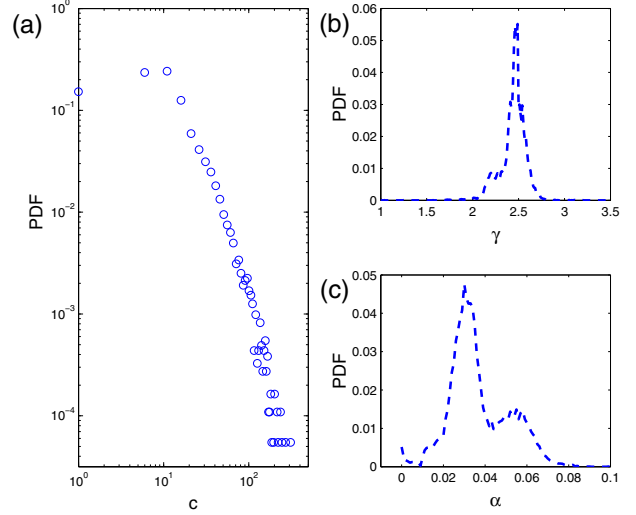


Figure 9: Distributions of model parameters

Figure 9(a) illustrates the distribution of fitness parameter  $c$ . Clearly we can see that  $c$  follows roughly a power-law distribution, indicating that most of messages share the similar small fitness value, while only a small portion of messages are created to be stronger competitive than others. For the exponent of power-law temporal relaxation function, parameter  $\gamma$ , as shown in Figure 9(b), most values of  $\gamma$  lie between 2 and 3. Moreover, we show the distribution of decay factor  $\alpha$  in exponential reinforcement function in Figure 9(c). Note that, a smaller  $\alpha$  indicates a weaker decay in the reinforcement function. When  $\alpha = 0$ , the exponential reinforcement function degenerates into the linear reinforcement function. From Figure 9(c) we can find that the value of  $\alpha$  ranges from 0 to 0.1, indicating that the decay parameter  $\alpha$  takes effect in the reinforcement function.

## 7. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed an extended reinforced Poisson process model with time mapping process to model the retweeting dynamics and predict the future popularity for messages on microblogging platforms. The proposed model captures a power-law temporal relaxation function which corresponds to the aging in the ability of a message to attract new retweets and an exponential reinforcement function which characterizes the “richer-get-richer” phenomenon. Both the temporal relaxation function and the reinforcement function are derived from real data. Further, observing that the hourly variation of user activity can affect the retweeting dynamics in the early stages, we introduced the notation of weibo time and applied a time mapping process to transform the time moments in the retweeting dynamics into weibo

time. To evaluate the effectiveness of our proposed model, we conducted extensive experiments on two Weibo datasets, with 10K and 18K messages respectively. The experimental results show that our proposed model outperforms existing prediction models in popularity prediction problem.

In the future, we will focus on enriching the proposed model by incorporating more relevant factors, e.g., structural factors derived from the underlying user network and the content factors extracted from the message itself.

## Acknowledgements

This work is supported by the National Natural Science Foundation of China (61272240, 61103151), the Doctoral Fund of Ministry of Education of China (20110131110028), the Excellent Middle-Aged and Youth Scientists of Shandong Province (BS2012DX017) and the Natural Science Foundation of Shandong Province (ZR2012FM037).

## 8. REFERENCES

- [1] M. Ahmed, S. Spagna, F. Huici, and S. Niccolini. A peek into the future: predicting the evolution of popularity in user generated content. In *WSDM*, pages 607–616, 2013.
- [2] S. Asur and B. A. Huberman. Predicting the future with social media. In *WI-IAT*, volume 1, pages 492–499, 2010.
- [3] R. Bandari, S. Asur, and B. A. Huberman. The pulse of news in social media: Forecasting popularity. In *ICWSM*, 2012.
- [4] P. Bao, H.-W. Shen, J. Huang, and X.-Q. Cheng. Popularity prediction in microblogging network: a case study on sina weibo. In *WWW*, pages 177–178, 2013.
- [5] J. Cheng, L. Adamic, P. A. Dow, J. M. Kleinberg, and J. Leskovec. Can cascades be predicted? In *WWW*, pages 925–936, 2014.
- [6] R. Crane and D. Sornette. Robust dynamic classes revealed by measuring the response function of a social system. *PNAS*, 105(41):15649–15653, 2008.
- [7] F. Figueiredo. On the prediction of popularity of trends and hits for user generated videos. In *WSDM*, pages 741–746, 2013.
- [8] S. Gao, J. Ma, and Z. Chen. Effective and effortless features for popularity prediction in microblogging network. In *WWW*, pages 269–270, 2014.
- [9] M. Gomez-Rodriguez, J. Leskovec, et al. Modeling information propagation with survival theory. In *ICML*, pages 666–674, 2013.
- [10] M. Gupta, J. Gao, C. Zhai, and J. Han. Predicting future popularity trend of events in microblogging platforms. *ASIST*, 49(1):1–10, 2012.
- [11] L. Hong, O. Dan, and B. D. Davison. Predicting popular messages in twitter. In *WWW*, pages 57–58, 2011.
- [12] A. Kupavskii, L. Ostroumova, A. Umnov, S. Usachev, P. Serdyukov, G. Gusev, and A. Kustarev. Prediction of retweet cascade size over time. In *CIKM*, pages 2335–2338, 2012.
- [13] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *WWW*, pages 591–600, 2010.
- [14] K. Lerman and T. Hogg. Using a model of social dynamics to predict popularity of news. In *WWW*, pages 621–630, 2010.
- [15] H. Li, X. Ma, F. Wang, J. Liu, and K. Xu. On popularity prediction of videos shared in online social networks. In *CIKM*, pages 169–178, 2013.
- [16] F. M. Lilian Weng and Y.-Y. Ahn. Virality prediction and community structure in social networks. *Scientific reports*, 3, 2013.
- [17] S. Lin, X. Kong, and P. S. Yu. Predicting trends in social networks via dynamic activeness model. In *CIKM*, pages 1661–1666, 2013.
- [18] Z. Ma, A. Sun, and G. Cong. On predicting the popularity of newly emerging hashtags in twitter. *JASIST*, 2013.
- [19] Y. Matsubara, Y. Sakurai, B. A. Prakash, L. Li, and C. Faloutsos. Rise and fall patterns of information diffusion: Model and implications. In *KDD*, pages 6–14, 2012.
- [20] M. McGlohon, J. Leskovec, C. Faloutsos, M. Hurst, and N. Glance. Finding patterns in blog shapes and blog evolution. In *ICWSM*, 2007.
- [21] H. Pinto, J. M. Almeida, and M. A. Gonçalves. Using early view patterns to predict the popularity of youtube videos. In *WSDM*, pages 365–374, 2013.
- [22] S. Redner. Citation statistics from 110 years of physical review. *PHYSICS TODAY*, 58(6):49–54, 2005.
- [23] H.-W. Shen, D. Wang, C. Song, and A.-L. Barabási. Modeling and predicting popularity dynamics via reinforced poisson processes. In *AAAI*, 2014.
- [24] B. Suh, L. Hong, P. Pirolli, and E. H. Chi. Want to be retweeted? large scale analytics on factors impacting retweet in twitter network. In *Socialcom*, pages 177–184, 2010.
- [25] G. Szabo and B. A. Huberman. Predicting the popularity of online content. *Commun. ACM*, 53:80–88, 2010.
- [26] M. Tsagkias, W. Weerkamp, and M. De Rijke. Predicting the volume of comments on online news stories. In *CIKM*, pages 1765–1768, 2009.
- [27] O. Tsur and A. Rappoport. What’s in a hashtag?: content based prediction of the spread of ideas in microblogging communities. In *WSDM*, pages 643–652, 2012.
- [28] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Weppe. Predicting elections with twitter: What 140 characters reveal about political sentiment. *ICWSM*, 10:178–185, 2010.
- [29] D. Wang, C. Song, and A.-L. Barabási. Quantifying long-term scientific impact. *Science*, 342(6154):127–132, 2013.
- [30] C. Williams and G. Gulati. What is a social network worth? facebook and vote share in the 2008 presidential primaries. American Political Science Association, 2008.
- [31] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *WSDM*, pages 177–186, 2011.
- [32] P. Zhang, X. Wang, and B. Li. On predicting twitter trend: Factors and models. In *ASONAM*, 2013.