# Modelling Cascades Over Time in Microblogs

Wei Xie, Feida Zhu
*Living Analytics Research Centre*
*Singapore Management University*
{*wei.xie.2012, fdzhu*}*@smu.edu.sg*

Siyuan Liu
*Smeal College of Business*
*Penn State University*
*siyuan@psu.edu*

Ke Wang
*Simon Fraser University*
*Singapore Management University**
*wangk@cs.sfu.ca*

***Abstract*—One of the most important features of microblogging services such as Twitter is how easy it is to re-share a piece of information across the network through various user connections, forming what we call a "cascade". Business applications such as viral marketing have driven a tremendous amount of research effort predicting whether a certain cascade will go viral. Yet the rarity of viral cascades in real data poses a challenge to all existing prediction methods. One solution is to simulate cascades that well fit the real viral ones, which requires our ability to tell *how a certain cascade grows over time*. In this paper, we build a general time-aware cascade model for each particular cascade, in which the chance of one user's re-sharing behaviour over time is modelled as a hazard function of time. Based on two key observations on user retweeting behaviour, we design an appropriate hazard function specifically for Twitter network. We evaluate our model on a large real Twitter dataset with over two million retweeting cascades. Our experiment results show our proposed model outperforms other baseline models in terms of model fitting. Further, we make use of our model to simulate viral cascades, which are otherwise few and far in-between, to alleviate the imbalance issue in cascade data, offering a 20% boost in viral cascade discovery.**

## I. Introduction

Social network services nowadays have made it all too easy for everyone to pass a piece of information to someone else. In this paper, we call such a user a "re-sharer" and the entire diffusion process of a piece of message through users a *cascade*. For example, in the case of Twitter, to re-share is simply to retweet a tweet. The size of a cascade refers to the total number of users that the piece of information eventually reaches.

Not surprisingly, the fact that the size of a cascade often translates into the influence of a message (e.g. the business impact of a marketing campaign) has driven a great deal of research to answer questions like: "How many users will a given tweet eventually reach?" Prediction of the size of a cascade, especially at the early stage of its growth, is critical in identifying among billions the truly viral messages, so that we can monitor, trace and leverage their impact.

As a result, most of the research work devoted to this task, e.g., [3] [18] [20] [15] and [1], have focused on predicting the cascade size, especially the size of viral one. Despite the different approaches in their solutions, one thing in common is that all of them must have real viral cascade instances as input, the more the better. Unfortunately, however, in real life truly viral cascades are few and far in between as compared to the whole set of tweets (in our experiment only 1‰ re-tweeting cascades grow over 35), resulting in a challenge for all existing solutions to further enhance their prediction accuracy. One solution to alleviate the rarity of viral cascades is to be able to simulate cascades that well fit the real viral ones. This motivates us to answer *how a certain cascade grows over time*, and integrate the time dimension into cascade modelling for each particular cascade.

There has been several cascade models, including the well-known Threshold Model [14], Cascade Model [11] in which the time fact is not a concern, and recent works such as [12] [23] [13] [7], which focus on integrating the time dimension into the model. Particularly, in these recent works a hazard function of time is used to model how information diffuse over time in networks. In this paper, we call them time-aware cascade models. Following this line, we build a general time-aware cascade model for each particular cascade to depict its development over time in the social network. What's more, we make two key observations on user retweeting behaviour and developed four constraints based on which we designed an appropriate hazard function for the model. We present an illustrative example as well as extensive experiments to demonstrate how our model fits to the real data. We would like to point out that, although hazard function has been used to model information propagation in general, in this work we make effort to design customised hazard functions based on properties of information diffusion in microblogging services.

Finally, we propose a strategy to make use of the simulations of our model to remedy the imbalance issue in cascade data. We show with experiments that the prediction performance improves as a result of the strategy with more viral cascades successfully identified.

## II. Preliminaries

Since we are going to use hazard function in survival analysis [22] to model time in cascade, we first give a brief

*This work was done partially while the author was visiting Singapore Management University.

978-1-4799-9926-2/15/$31.00 ©2015 IEEE

introduction the basic concepts of survival analysis in this section.

Survival analysis focuses on time-to-event data, especially the survival time until an event of failure. While typical examples of events of interest are biological death and the failure in mechanical systems [16], survival analysis is in fact generic and can be applied to model any time-to-event data.

Consider as a random variable $T$ the time when an event of interest (e.g., when a Twitter user retweets one particular tweet) happens . The probability that the event happens before a certain time $t$ is $P(T \leq t) = F(t)$, which is the cumulative distribution function (CDF). Without loss of generality, suppose $F(0) = 0$. The probability density function (PDF) $f(t)$ is defined as the derivate of $F(t)$, i.e. $f(t) = \frac{\mathrm{d}F(t)}{\mathrm{d}t}$.

When processing time-to-event data, given that the event has not happened by time $t$, the probability that this event happens in the next time slice $(t, t + dt]$ is usually of great interest, i.e., $P(t < T \leq t + dt | T > t)$. For example, when we observe a Twitter user has not retweeted one particular tweet by time $t$, it is highly critical to estimate the probability this Twitter user retweets it in the next time slice $(t, t+dt)$. The **hazard function** (or **hazard rate**) [22] is defined as

$$h(t) = \lim_{dt \to 0} \frac{P(t < T \leq t + dt | T > t)}{dt} = \frac{f(t)}{1 - F(t)}.$$

The hazard function $h(t)$ reflects the chance that the event happens immediately after time $t$. By integrating $h(t)$, we have the cumulative hazard function $H(t) = \int_0^t h(u)\mathrm{d}u = \int_0^t \frac{F'(u)}{1-F(u)}\mathrm{d}u = -log(1 - F(u))|_0^t = -log(1 - F(t))$. It reveals the essential relationship between the cumulative distribution function $F(t)$ and the cumulative hazard function $H(t)$ in the following equation

$$F(t) = 1 - e^{-H(t)}.$$

In practice, it is usually impossible to know the exact formula of CDF $F(t)$. However, by studying the hazard rate, we can design the formula of hazard function $h(t)$ or cumulative hazard function $H(t)$, and further derive the formula of $F(t)$ to approximate the real distribution. One example is simply to set $H(t)$ as a linear term, i.e. $H(t) = \frac{t}{\lambda}$. Its corresponding CDF is $F(t) = 1 - e^{-\frac{t}{\lambda}}$, which is in fact the exponential distribution with mean $\lambda$. Another slightly complicated example is $H(t) = (\frac{t}{\alpha})^{\beta}$. Its corresponding CDF is $F(t) = 1 - e^{-(\frac{t}{\alpha})^{\beta}}$, which is simply the Weibull distribution [27] with scale parameter $\alpha$ and shape parameter $\beta$. It turns out that the different choices of hazard functions lead to different probability models. In this work, our job is to construct the proper hazard function to approximate the real cascade development in social network.

## III. Time-aware Cascade Model

In this section, we describe a general time-aware cascade model from the aspect of one particular cascade, which is based on previous time-aware cascade models such as [12] [23] [13]. Particularly here we focus on the hazard function.

We consider a general social network $G =< U, E >$, where $U$ represents the set of users, and $E$ is a set of directed links between users of $U$ (See Figure 1), each representing the channel through which information in a cascade could flow in the directions as indicated. For example in Twitter, if $u_i$ follows $u_j$, there is a directed link going from $u_j$ to $u_i$, denoted as $(u_j, u_i)$. To study cascades, given any user $u_i$, we are interested in the set of users whose information can potentially reach $u_i$ in cascades, which we call $u_i$'s *followee set* and denote as $Followee^{(i)} = \{u_j | (u_j, u_i) \in E\}$. Similarly, we also care about the set of users who can potentially receive information from $u_i$, which we call $u_i$'s *follower set*, and denote as $Follower^{(i)} = \{u_j | (u_i, u_j) \in E\}$. For a cascade of any message, we denote as $u_0$ the original source and use a random variable $T_i$ to denote the timestamp when user $u_i$ re-shares the message. Correspondingly, $t_i$ is the observation of random variable $T_i$. As a trivial case, $T_0 = t_0$ with probability 1 and $t_i = \infty$ if $u_i$ never re-shares the message.

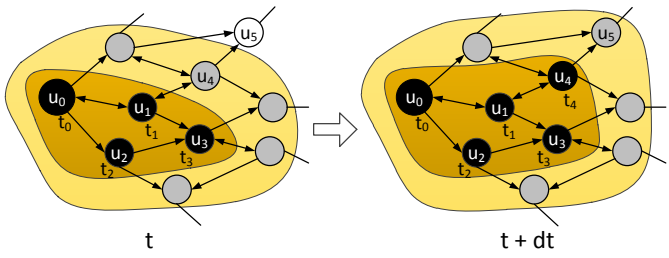



Figure 1. A snapshot in the development of a cascade (from timestamp $t$ to $t + dt$).

As illustrated in Figure 1, at any timestamp in the cascade development, we identify three types of nodes: (I) **Black Nodes**: the users who have already re-shared the information, with their re-share timestamps; (II) **Grey Nodes**: the users who are exposed to the information, yet have not re-shared it; and (III) **Blank Nodes**: the users who yet to be exposed to the information. As the cascade develops, Blank Nodes could become Grey Nodes (e.g., $u_5$), which in turn could become Black Nodes (e.g., $u_4$).

Intuitively, we model the development of a cascade as a stochastic process as follows. Suppose at timestamp $t_0$, the information source $u_0$ posts a piece of information. At each following timestamp $t$, we take a probabilistic point of view similar to Cascade Model toward the growth of the cascade. The key is to focus on the Grey Nodes as they are the only ones to potentially re-share the information in the next small time slice $(t, t + dt]$. We denote the Grey Nodes as $\mathbb{X}(t)$.

At time $t+dt$, $\mathbb{X}(t)$ can be divided into two parts: (I) those who have shared the information by $t+dt$ (i.e., they become Black Nodes), which is denoted as $\mathbb{X}^{(1)}(t)$, and (II) the rest, which is denoted as $\mathbb{X}^{(2)}(t)$, i.e., they remain Grey Nodes, $\mathbb{X}^{(2)}(t) = \mathbb{X}(t) \setminus \mathbb{X}^{(1)}(t)$.

At any timestamp $t$, a cascade can be represented by a set of pairs $\mathbb{C}(t) = \{< u_i, t_i > | t_i \leq t\}$, e.g., $\mathbb{C}(t) = \{< u_0, t_0 >, < u_1, t_1 >, < u_2, t_2 >, < u_3, t_3 >\}$ in Figure 1. Similarly, $\mathbb{C}(t, t+dt) = \{< u_i, t_i > | t < t_i \leq t+dt\}$, e.g., $\mathbb{C}(t, t+dt) = \{< u_4, t_4 >\}$ in Figure 1. Denote $P_i(t)$ as the probability that user $u_i$ re-shares the information in time slice $(t, t+dt]$ given she has not re-shared the information by time $t$. (It is worth noting that $P_i(t)$ is information specific, see Equation 2 below.) Then the probability that the cascade grows from $\mathbb{C}(t_0)$ into $\mathbb{C}(t)$ can be defined as the following recursive equation.

$$\begin{cases} P(\mathbb{C}(t+dt)) = P(\mathbb{C}(t+dt)|\mathbb{C}(t)) \cdot P(\mathbb{C}(t)) \\ P(\mathbb{C}(t_0)) = 1 \\ P(\mathbb{C}(t+dt)|\mathbb{C}(t)) = \prod\limits_{u_i \in \mathbb{X}^{(1)}(t)} P_i(t) \cdot \prod\limits_{u_{i'} \in \mathbb{X}^{(2)}(t)} (1 - P_{i'}(t)) \end{cases} \tag{1}$$

Equation 1 defines the entire stochastic process of a growing cascade, which only depends on the probability $P_i(t)$. From previous analysis, it is not hard to see that $P_i(t)$ depends on the users in her followee set who have already shared this piece of information by time $t$, which is denoted as $Followee^{(i)}(t) = \{u_j | t_j \leq t, u_j \in Followee^{(i)}\}$. It follows that $P_i(t) = P(t < T_i \leq t+dt | T_i > t, \{T_j = t_j\}_{u_j \in Followee^{(i)}(t)})$. We model this probability using the following hazard function.

$$P_i(t) = h_i(t, \{t_j\}_{u_j \in Followee^{(i)}(t)}; \mathbf{\Theta}) \cdot dt \tag{2}$$

where $\mathbf{\Theta}$ are the parameters which are related to the original information posted by $u_0$ and, reflect how tensely this piece of information gets the interest of public and how quickly users react to it. If at time $t$ for user $u_i$, $Followee^{(i)}(t)$ is an empty set (e.g. the blank node $u_5$ at time $t$ in Figure 1), then just let $h_i(t, \{t_j\}_{u_j \in Followee^{(i)}(t)}; \mathbf{\Theta}) = 0$.

To apply this general time-aware cascade model to any particular social network, one only has to identify the appropriate hazard function in Equation 2 and the parameters $\mathbf{\Theta}$, after which the stochastic process would be fully defined. So the crucial thing here is to design a proper formula for the hazard function in Equation 2, which really fits the real cascades.

## IV. Model Application: Twitter

In this section, we show how the general time-aware model in Section III can be applied to the concrete setting of Twitter to model cascades of tweets. As mentioned in Section III, the key issue is to design the appropriate hazard function in Equation 2. For this, we first observe how a tweet is retweeted in Twitter, then give criteria for the hazard function, at last propose the the appropriate hazard function in Twitter.

### A. Observations

In Twitter, we identify the following two observations which affect how a tweet would be retweeted in a cascade, and accordingly provide us the clues to design the appropriate hazard function in Equation 2.

**Observation 1.** *Only the first re-sharer matters.* For any user, only the tweet from the re-sharer who first re-shares it would appear in the user's home timeline. For example, in Figure 1, suppose $u_1$ re-shares the tweet from $u_0$ earlier than $u_2$, i.e., $t_1 < t_2$. Then although both $u_1$ and $u_2$ are in the followee set of $u_3$, only the tweet re-shared by $u_1$ appears in $u_3$'s home timeline at time $t_1$. Based on this observation, we conclude that only the first re-sharer in a user's followee set would affect her subsequent retweeting behaviour. Consequently, we have the following simpler formula (Equation 3) instead of the general formula in Equation 2.

$$P_i(t) = h_{i,j^\star}(t, t_{j^\star}; \mathbf{\Theta}) \cdot dt \tag{3}$$

where $j^\star = argmin_j \{t_j | u_j \in Followee^{(i)}(t)\}$.

**Observation 2.** *The chance of a tweet to be retweeted decreases as time goes by.* As more recent tweets appear higher in a user's home timeline and is more likely to attract user's attention, the chance of a tweet to be retweeted decreases as it sinks down along the timeline. Based on this observation, we further refine the formula as follows in Equation 4.

$$P_i(t) = h_{i,j^\star}(t - t_{j^\star}; \mathbf{\Theta}) \cdot dt \tag{4}$$

where $h_{i,j^\star}(\tau, \mathbf{\Theta})$ is a decreasing function of $\tau$ ($\tau = t - t_{j^\star} > 0$). It is worth noting that not all the hazard functions are decreasing. Take the risk of death as an example, as people become older and older, the risk of death in fact becomes higher and higher.

Besides, as a trivial case, when $t \leq t_{j^\star}$, $u_i$ is not exposed to the tweet from the original user $u_0$, so there is no chance for $u_i$ to retweet this tweet. And when $t > t_{j^\star}$, $u_i$ have a chance to read the tweet from her own home timeline, and it is always possible for $u_i$ to retweet it. Naturally we have the following constrains in Equation 5.

$$\begin{aligned} h_{i,j^\star}(\tau; \mathbf{\Theta}) &= 0, (\tau \leq 0) \\ h_{i,j^\star}(\tau; \mathbf{\Theta}) &> 0, (\tau > 0) \end{aligned} \tag{5}$$

### B. Hazard Function Design

The analysis in Subsection IV-A leads to the intuition that the appropriate hazard function in Twitter setting should be a decreasing function $h_{i,j^\star}(\tau; \mathbf{\Theta})$. In this subsection, we develop the concrete formula of $h_{i,j^\star}(\tau; \mathbf{\Theta})$. First, we make

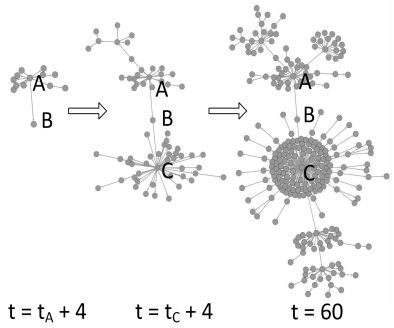



Figure 2. A Real Cascade Example.

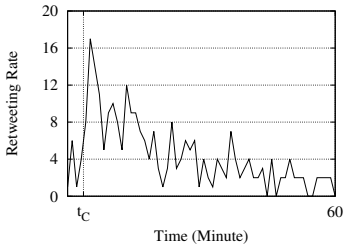



Figure 3. Retweeting Rate Per Minute.

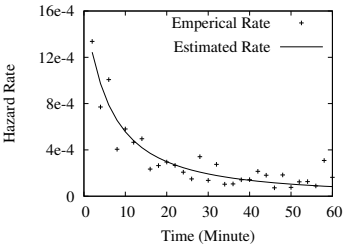



Figure 4. Hazard Rate Comparison.

a simplification to replace $h_{i,j^\star}(\tau;\mathbf{\Theta})$ by $h(\tau;\mathbf{\Theta})$, which means no matter who $u_i$ is and who is the first re-sharer $u_{j^\star}$, their hazard functions share the same formula $h(\tau;\mathbf{\Theta})$. Without misunderstanding, we just omit the parameter set $\mathbf{\Theta}$, and the hazard function is simply denoted as $h(\tau)$.

As $H(\tau)$ is the integration of $h(\tau)$, according to Equation 5, we have $H(0) = 0$ and $H(\tau)$ should be a increasing function of $\tau$. Besides, another fact is that if user $u_i$ hasn't retweeted the tweet from $u_0$, $u_i$'s home timeline will be full of other new incoming tweets, and $u_i$ will never re-share it. It means $F(\infty) < 1$, and consequently $H(\infty) = -log(1 - F(\infty)) < \infty$. Based on all the analysis in Subsection IV-A and IV-B, we list the constrains for $H(\tau)$ in Table I.

| | |
|---|---|
| I) | $H(0) = 0$. |
| II) | $H(\infty) = -log(1 - F(\infty)) < \infty$. |
| III) | $H(\tau)$ is an increasing function of $\tau$. |
| IV) | $h(\tau) = \frac{\mathrm{d}H(\tau)}{\mathrm{d}\tau}$ is a decreasing function of $\tau$. |

Table I
HAZARD FUNCTION CONSTRAINTS FOR TWITTER.

Denote $\lambda = H(\infty)$ as one of the parameters. According to these constrains, our observations from the data (see Figure 4 below) and the insight of human behaviour from other work [2], we eventually propose the following heavy-tail hazard functions $H(\tau)$ and $h(\tau)$ for Twitter setting.

**Hazard Function For Twitter Cascades:**

$$H(\tau) = \lambda \cdot (1 - (\frac{\tau}{\alpha} + 1)^{-\beta}) \tag{6}$$

$$h(\tau) = \frac{\mathrm{d}H(\tau)}{\mathrm{d}\tau} = \lambda \cdot \frac{\beta}{\alpha} \cdot (\frac{\tau}{\alpha} + 1)^{-(\beta+1)} \tag{7}$$

where $\lambda > 0$, $\alpha > 0$, $\beta > 0$. Here the parameter set $\mathbf{\Theta} = \{\lambda, \alpha, \beta\}$. In fact, our experiment shows that the probability that one user who is exposed to one tweet actually retweets it at last is quite small (around 0.01), which means $F(\infty)$ is near to 0 and $F(\infty) \approx H(\infty) = \lambda$. So the parameter $\lambda$ describes the eventual re-tweeting probability. The larger $\lambda$ is, the larger is the proportion of users who re-tweet one particular tweet in the users who are exposed to it. According to Equation 7, the parameter $\alpha$ describes the scale of hazard function $h(\tau)$, and the parameter $\beta$ describes the shape of $h(\tau)$, which are similar to the scale parameter and shape parameter of Weibull distribution.

### C. Hazard Rate Illustration

We use a real cascade example here to illustrate how the proposed hazard function fits the hazard rate in real data. Section VI presents more comprehensive evaluation including how different choices of hazard function affect the fitting performance of the model.

Figure 2 shows the growing process of a real cascade of a local news tweet — The tweet was initiated from node $A$ the original source, and passed through node $B$ before reaching node $C$ which has the largest degree and has played the most important role in triggering the dramatic growth of the cascade. This is also demonstrated by Figure 3 which shows the retweeting rate of this cascade, i.e., the number of users who have retweeted per minute. The spike corresponding to the greatest drive to the retweeting rate happened at timestamp $t_C$, after user $C$ retweeted the message.

We calculate the empirical hazard rate of this cascade as follows. Two sets of users are involved: (I) the set of users who have retweeted the tweet by time $T$, i.e., $\mathbb{C}(T)$ and (II) the set of users who have been exposed to the tweet but haven't retweeted it yet, i.e., $\mathbb{X}(T)$, where $T$ is long enough for estimating the empirical hazard rate. For user $u_i$ in $\mathbb{C}(T)$, we calculate $\tau_i = t_i - t_{j^\star}$, where $j^\star = argmin_j\{t_j | u_j \in Followee^{(i)}(t)\}$. $\tau_i$ measures how long it takes for $u_i$ to retweet the tweet after being exposed to it. For user $u_i$ in $\mathbb{X}(T)$, we calculate $\tau_i = T - t_{j^\star}$, which measures how long $u_i$ has been exposed to the tweet. According to the definition of hazard rate in Section II, the empirical hazard rate of this cascade is calculated by using the following Equation 8.

$$\begin{aligned}\hat{h}(\tau) &= \frac{\hat{P}(\tau < T \leq \tau + dt | T > \tau)}{dt} \\ &= \frac{|\{u_i | u_i \in \mathbb{C}(T), \tau < \tau_i \leq \tau + dt\}| + 1}{|\{u_i | u_i \in \mathbb{C}(T) \cup \mathbb{X}(T), \tau < \tau_i\}| + 2} \cdot \frac{1}{dt}\end{aligned} \tag{8}$$

Figure 4 shows the empirical hazard rate and the estimated hazard rate based on our proposed hazard function in Equation 7. It is clear that the estimated hazard rate fits the empirical hazard rate quite well, which means that our proposed hazard function is an appropriate one giving a good approximation to the real hazard rate. It can also be observed that the real hazard rate does decrease as the longer users are exposed to the tweet.

## V. Model Implementation

In this section, we give detailed algorithms for model implementation. In particular, we show the parameter estimation algorithm and the cascade simulation algorithm.

**Input :** social network $G$, cascade $\mathbb{C}(T)$, time interval $dt$, hazard function $h(t;\mathbf{\Theta})$

**Output :** estimated parameters $\hat{\mathbf{\Theta}}$

1: set log likelihood function $ll(\mathbf{\Theta}) = 0$
2: set type II users $\mathbb{X} = \{\}$
3: set $Current\_Sharers = \{u_0\}$
4: **for** $t = 0$ **to** $T$ **step** $dt$ :
5: **for** $u_i \in Current\_Sharers$
6: add $Follower^{(i)} \backslash \mathbb{C}(t)$ into set $\mathbb{X}$
7: **endfor**
8: **for** $u_i \in \mathbb{X}$
9: **if** $u_i \in \mathbb{C}(t, t+dt)$
10: $ll(\mathbf{\Theta}) = ll(\mathbf{\Theta}) + log(P_i(t;\mathbf{\Theta}))$
11: **else**
12: $ll(\mathbf{\Theta}) = ll(\mathbf{\Theta}) + log(1 - P_i(t;\mathbf{\Theta}))$
13: **endif**
14: **endfor**
15: **for** $u_i \in \mathbb{C}(t, t+dt)$
16: remove $u_i$ from set $\mathbb{X}$
17: **endfor**
18: set $Current\_Sharers = \mathbb{C}(t, t+dt)$
19: **endfor**
20: $\hat{\mathbf{\Theta}} = argmax_{\mathbf{\Theta}} ll(\mathbf{\Theta})$
21: **return** $\hat{\mathbf{\Theta}}$

Table II
The Parameter Estimation Algorithm.

### A. Parameter Estimation

Given a cascade $\mathbb{C}(T)$, based on Equation 1, maximum likelihood estimation is used to estimate the unknown parameters,

$$\hat{\mathbf{\Theta}} = argmax_{\mathbf{\Theta}} log(P(\mathbb{C}(T);\mathbf{\Theta})).$$

The detailed algorithm is shown in Table II.

The key part is to calculate the log-likelihood function $ll(\mathbf{\Theta}) = log(P(\mathbb{C}(T);\mathbf{\Theta}))$, then optimisation techniques such as gradient ascent can be used to estimate the best parameter $\hat{\mathbf{\Theta}}$. At each time step $t$ (line 4), users who are just exposed to the information by time $t$ are added into the set of type II Grey Node users $\mathbb{X}$ (line 5-7). For all the users in $\mathbb{X}$, the log-likelihood of their re-sharing behaviour at $t$ is calculated (line 8-14). Finally, we have the log-likelihood function $ll(\mathbf{\Theta})$, and obtain the estimated parameters $\hat{\mathbf{\Theta}}$ by optimising it (line 20).

**Input :** social network $G$, cascade $\mathbb{C}(T_0)$ by time $T_0$, simulation duration $\Delta T$,time interval $dt$,parameters $\mathbf{\Theta}$, hazard function $h(t;\mathbf{\Theta})$

**Output :** simulated cascade $\widetilde{\mathbb{C}}(T_0 + \Delta T)$

1: set type II users $\mathbb{X} = \{\}$
2: set $Current\_Sharers = \mathbb{C}(T_0)$
3: **for** $t = T_0$ **to** $T_0 + \Delta T$ **step** $dt$ :
4: **for** $u_i \in Current\_Sharers$
5: add $Follower^{(i)} \backslash \mathbb{C}(t)$ into set $\mathbb{X}$
6: **endfor**
7: **for** $u_i \in \mathbb{X}$
8: draw a random number $r$ from $U(0,1)$
9: **if** $r < P_i(t;\mathbf{\Theta})$
10: add pair $< u_i, t+dt >$ into $\widetilde{\mathbb{C}}(t, t+dt)$
11: **endif**
12: **endfor**
13: **for** $u_i \in \widetilde{\mathbb{C}}(t, t+dt)$
14: remove $u_i$ from set $\mathbb{X}$
15: **endfor**
16: set $Current\_Sharers = \widetilde{\mathbb{C}}(t, t+dt)$
17: **endfor**
18: **return** $\widetilde{\mathbb{C}}(T_0 + \Delta T)$

Table III
The Cascade Simulation Algorithm.

### B. Cascade Simulation

Once the parameters $\mathbf{\Theta}$ are known, based on Equation 2, Monte Carlo simulation is used to simulate the cascade from time $T_0$ to time $T_0 + \Delta T$. The detailed algorithm is presented in Table III.

In this algorithm, the key part is to simulate the re-sharing behaviour of each user at each time step $t$. Similar to the estimation algorithm, at each time step $t$ (line 3), users who are just exposed to the information by time $t$ are added into the set of type II Grey Node users $\mathbb{X}$ (line 4-6). For all the users in $\mathbb{X}$, their re-sharing behaviour at $t$ are simulated (line 7-12). Finally at time $T_0 + \Delta T$, we have the simulated cascade $\widetilde{\mathbb{C}}(T_0 + \Delta T)$ (line 18).

## VI. Experiment

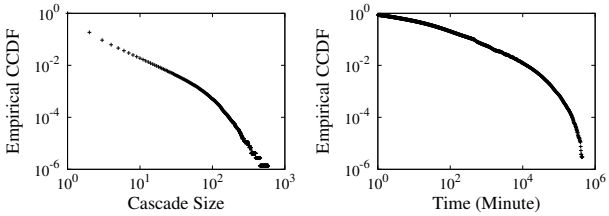
In this section, we conduct extensive experiments to evaluate our proposed model, in the following aspects: (I) probabilistic model fitting, (II) prediction of cascade growth, and (III) virality prediction. In all the experiments, the length of time slice $dt$ is set to 0.1 minute.

### A. Dataset

We use a Singapore based Twitter data set which contains more than 3 million users [28]. We crawl these users from a seed set of Singapore local celebrities and active users in a snowball-style way. The follow links between them and their tweets are periodically crawled. In this work, we use the subset of tweets from January 1st, 2010 to December 31th, 2012. From these tweets, we get all the retweets to construct retweeting cascades (see Figure 1). In all these cascades, we only consider the cascades in which the original tweet

posters are the Singapore users we crawled, so that we have the information about the root users of the cascades.

In all, we get 2,425,348 cascades which have at least one retweeter. Figure 5 (a) presents the cumulative distribution of the sizes of these cascades. It shows the large cascades are rare, which implies the difficulty in predicting the cascade growth, as most cascades do not grow anymore when they are in small sizes. Figure 5 (b) presents the cumulative distribution of the retweeting time delays of users after they are exposed to the tweets. From this figure, we can see even after long time, it is still possible for one tweet to be retweeted. This observation confirms our choice of hazard function in Equation 7, which is heavy tailed.




(a) (b)

Figure 5. (a) Cumulative distribution of the cascade sizes. (b) Cumulative distribution of the time delays.

### B. Probabilistic Model Fitting

First, we evaluate the validity of our model. As the common way to check a probabilistic model, perplexity, which is a measurement of how well a probability model predicts a held-out sample, is used to measure the model validity. Given a set of cascades with size $n$, $\{\mathbb{C}_i(t)\}_{i=1}^n$, for each cascade $\mathbb{C}_i(t)$, we first observe it by time $T_0$, then $\Delta T$ later, based on a model $\mathcal{M}$, the probability $P_{\mathcal{M}}(\mathbb{C}_i(T_0+\Delta T)|\mathbb{C}_i(T_0))$ is calculated. The formula of perplexity is defined in the Equation 9. The smaller the perplexity is, the better fitting the model is.

$$Perplexity(\mathcal{M}) = e^{-\sum_{i=1}^{n} \frac{1}{n} \cdot log(P_{\mathcal{M}}(\mathbb{C}_i(T_0+\Delta T)|\mathbb{C}_i(T_0)))} \quad (9)$$

In this experiment, we examine: first whether the time-aware cascade model (TCM) is better than the traditional ones such as Threshold Model (TM) in terms of modelling information cascade over time; secondly whether our proposed hazard function in Equation 7 is more suitable in Twitter setting than others. As in TM, time is not a factor of concern, we adapt the models by projecting the growing cascade on the time dimension according to its original idea. In order to study how the different choices of hazard functions affect the time-aware model, we examine the following different choices of hazard functions – our proposed long tail hazard function (TCM-LH) in Equation 7, constant hazard function (TCM-CH) in Equation 12 and exponential hazard function (TCM-EH) in Equation 14. All these models are expressed in terms of hazard function, $h_i(t)$, which represents user $u_i$'s re-sharing chance at time $t$. And all the parameters in these models are estimated by using maximum likelihood estimation algorithm listed in Table II.

- **TM$_t$**: Threshold Model proposed the key concept "threshold", which in the setting of Twitter network is such a value: if the number of a user's followees who have retweeted one tweet exceeds this value, then this user retweets this tweet. In TM, the key point is threshold rather than time. In order to integrate time into this baseline TM$_t$, we use sigmoid function as the continuous thresholding function [9] and the hazard function is given in the following Equation 10.

$$h_i(t) = \lambda \cdot s(|Followee^{(i)}(t)|) \quad (10)$$

  where $s(x) = \frac{1}{1+e^{-a(x-b)}}$, and $\lambda$, $a$ and $b$ are parameters to be estimated.
- **TCM-CH**: Constant hazard in Equation 12 is the easiest way to define a hazard function, which is also considered in [13]. However, in this case $\lim_{\tau \to \infty} H(\tau) = \infty$, which does not satisfy the constrains listed in Table I.

$$H(\tau) = \lambda \cdot t \quad (11)$$

$$h(\tau) = \frac{H(\tau)}{d\tau} = \lambda \quad (12)$$

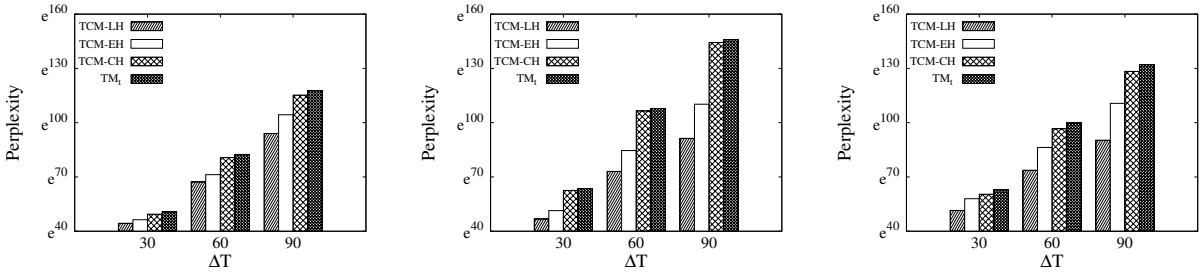
  where parameter set $\mathbf{\Theta} = \{\lambda\}$.
- **TCM-EH**: As some works such as [20] reported, the exponential function may be a proper function to model the time delay of retweeting. In this baseline, we use the exponential hazard function in the following Equation 13 and 14, which also satisfy the constrains in Table I.

$$H(\tau) = \lambda \cdot (1 - e^{-k \cdot \tau}) \quad (13)$$

$$h(\tau) = \frac{H(\tau)}{d\tau} = \lambda \cdot k \cdot e^{-k \cdot \tau} \quad (14)$$

  where parameter set $\mathbf{\Theta} = \{\lambda, k\}$.

In this experiment, $T_0$ is set to 30 minutes, and $\Delta T$ is set to 30, 60 and 90 minutes respectively. We evaluate these models on all the cascades which has a size larger than 10 at $T_0$ in year 2010, 2011 and 2012. Figure 6 shows the perplexities of different models. We can see for different year and $\Delta T$ the time-aware cascade models outperforms threshold model, and our model TCM-LH performs best among these time-aware cascade models , which means TCM-LH is a proper model for growing retweeting cascades in Twitter. One interesting observation is that performances of different time-aware cascade models are very different, which implies that the choice of hazard function is critical for fitting model to real cascade data. The other observation is that the perplexities of all the models in year 2010 are




(a) Year 2010 (b) Year 2011 (c) Year 2012

Figure 6. Perplexities of different models.

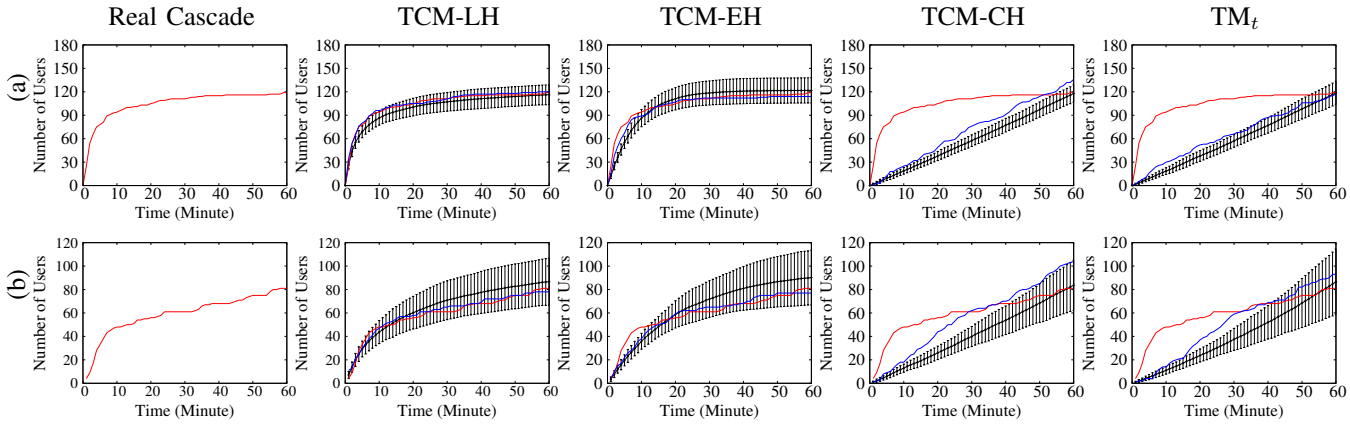



Figure 7. Fitting performances of our proposed model (TCM-LH) and other baseline models for two Twitter cascades (a) and (b). In each plot, the red curve shows how the number of users of the real cascade increases over time. Based on the real cascades, each model learns the parameters and then simulates 100 cascades. The black curve shows the average number of users of these 100 simulated cascades over time, and the error bars around it are the corresponding standard deviations. The blue curve represents the simulated cascade that is closest to the real one.

much smaller than the perplexities in year 2011 and 2012, which is due to the smaller sizes of cascades in year 2010.

We also study the following two representative retweeting cascades: cascade (a) triggered by a tweet which promotes a music festival; cascade (b) triggered by a tweet about a local breaking news. For each cascade, each model learns the parameters and then simulates 100 cascades using the estimating algorithm and simulating algorithm in Section V. Figure 7 shows the fitting performances of our proposed model (TCM-LH) and other baseline models for these two retweeting cascades (a) and (b). In each plot of Figure 7, the red curve shows how the number of users of the real cascade increases over time. The black curve shows the average number of users of the 100 simulated cascades over time, and the error bars around it are the standard deviations. The blue curve shows the simulated cascade which is nearest to the real cascade. We can see that: cascade (a) has a high initial retweeting rate, but its rate dramatically decreases; different from cascade (a), cascade (b) has a relatively slow initial retweeting rate, however, the tweet continually gets the interest of users and the cascade keeps growing over time. It can be observed that for both cascades our proposed model TCM-LH performs the best — the black and blue curves are close to the red curves. TCM-EH is worse than TCM-LH. Especially for cascade (b), TCM-EH can not generate a similar cascade to it. For other baseline models, it seems that they can not fit the real cascade well — the simulated cascades are far from the real cascades. Due to missing the effect of time in TCM-CH (constant hazard rate) and TM$_t$ (threshold based hazard rate), the simulated cascades keep growing with roughly the same rates all the time. Another interesting observation is that the results of TCM-CH and TM$_t$ are similar. We examined these two cascades and found that for most retweeter, only one of their followees is in the cascade. It makes the learned threshold parameter $b$ in TM$_t$ close to 1, so that there is no significant difference between TCM-CH and TM$_t$.

Besides, using our proposed model TCM-LH, the learned parameters for these two cascades are as follows: for cascade (a) $\lambda^{(a)} = 0.0082$, $\alpha^{(a)} = 1.35$, $\beta^{(a)} = 0.50$; for cascade (b) $\lambda^{(b)} = 0.0140$, $\alpha^{(b)} = 4.78$, $\beta^{(b)} = 0.45$. Although cascade (a) has more retweeters at the early stage, based on the learned parameters above we make the following interpretations: $\lambda^{(b)} > \lambda^{(a)}$ means cascade (b) is more attractive in terms of the eventual retweeting probability; $\alpha^{(b)} > \alpha^{(a)}$ means the "life time" of cascade (b) is longer.

### C. Predicting Cascade Growth

The other way to verify our proposed model is to evaluate its prediction performance. As mentioned in several existed works [3] [20], the initial information of a growing cascade in social networks can make the prediction much more accurate. It would be more practical to predict the cascade size after observing how the cascade grows initially, rather than to predict the cascade size from the very beginning. Here we conduct a prediction task, in which a cascade is tracked over time, and a sequence of predictions are made as the cascade grows. Rather than predict the final cascade size, each time we predict the cascade growth after a fixed time period (e.g. one hour), which is more practical.

Given $n$ cascades $\{\mathbb{C}_i(t)\}_{i=1}^n$, the prediction time $T_0$, and fixed time period $\Delta T$, denoting the growth of cascade $\mathbb{C}_i(t)$ as $\Delta_i(T_0, T_0+\Delta T) = |\mathbb{C}_i(T_0+\Delta T)| - |\mathbb{C}_i(T_0)|$, the following evaluation measures are considered:

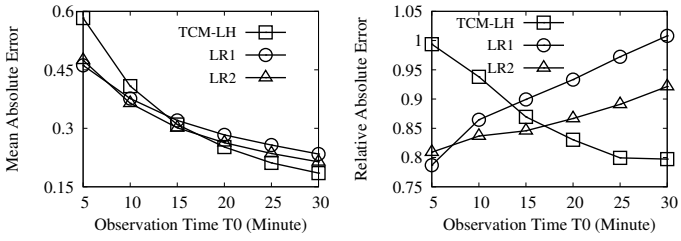
- **Mean Absolute Error (MAE)**
$$MAE = \frac{\sum_{i=1}^{n} |\Delta_i(T_0,T_0+\Delta T) - \hat{\Delta}_i(T_0,T_0+\Delta T)|}{n}$$
- **Relative Absolute Error (RAE)**
$$RAE = \frac{\sum_{i=1}^{n} |\Delta_i(T_0,T_0+\Delta T) - \hat{\Delta}_i(T_0,T_0+\Delta T)|}{\sum_{i=1}^{n} \Delta_i(T_0,T_0+\Delta T)}$$

where $\hat{\Delta}_i(T_0, T_0+\Delta T)$ is the estimation of $\Delta_i(T_0, T_0+\Delta T)$.

Based on the algorithms in Section V, TCM-LH predicts the cascade growth as follows. (I) Estimate the parameters based on $\mathbb{C}(T_0)$; (II) Simulate the cascade from time $T_0$ to $T_0+\Delta T$ for a large number of times, then take the median size of simulated cascades at time $T_0+\Delta T$.

We first compare TCM-LH with other baseline models in Section VI-B. We found that with big prediction errors, these baseline models are not suitable for this prediction task. We then compare TCM-LH to linear regression, which is widely applied in many prediction tasks such as popularity prediction in social media. We summarize most proposed factors [15] [1], which may drive the cascades grow or be relevant to the sizes of cascades in Twitter, including original tweeter features, tweet content features, social graph topological features and temporal features. And in the experiment, we found the temporal features are the most important predictors. These features of one tweet are denoted as a feature vector $\boldsymbol{f}$. We conduct the following two baselines.

- **LR1** In this baseline, the growth of a cascade is directly estimated. The regression formulation is given in the Equation 15. For some cascades, the increased cascade size is zero, so we use $log(\hat{\Delta}_i(T_0, T_0+\Delta T)+1)$ instead of $log(\hat{\Delta}_i(T_0, T_0+\Delta T))$.
$$log(\hat{\Delta}_i(T_0, T_0+\Delta T)+1) = k_0 + \boldsymbol{k}^T \cdot \boldsymbol{f} \qquad (15)$$
- **LR2** Some existing works such as [26] reported that there is a strong linear relationship found between the log-transformed popularities at different times. And $log(S(T_0))$ is indeed one feature included in the feature vector $\boldsymbol{f}$. So in this baseline, the cascade size at time $T_0+\Delta T$, i.e. $S(T_0+\Delta T)$, is first estimated. The regression formulation is given in the Equation 16.
$$\begin{aligned} log(\hat{S}(T_0+\Delta T)) &= k_0 + \boldsymbol{k}^T \cdot \boldsymbol{f} \\ \hat{\Delta}_i(T_0, T_0+\Delta T) &= \hat{S}(T_0+\Delta T) - S(T_0) \end{aligned} \qquad (16)$$




(a) (b)

Figure 8. (a) Mean Absolute Error (MAE). (b)Relative Absolute Error (RAE).

We randomly choose 10,000 cascades from our data set. These cascades are tracked over time (when $T_0$ = 5, 10, 15, 20, 25, 30 minutes), and at each time we predict the cascade growth one hour later ($\Delta T$ = 60 minutes). 10-fold cross validation is used. The prediction of TCM-LH is based on the median of 100 simulations. Figure 8 (a) and (b) show the Mean Absolute Error (MAE) and Relative Absolute Error (RAE) of TCM-LH, LR1 and LR2 respectively. We observe that: (I) In Figure 8 (a), the MAE decreases for all these three methods as longer we observe the cascades. One possible reason is that as time goes by, most cascades do not grow or only grow a little, so that MAE decreases over time. (II) However, in Figure 8 (b), we can see a very different trend: RAE does increase for LR1 and LR2 over time. One possible reason is that as time goes by, the correlation between the features and the cascade growth becomes smaller and smaller. (III) In Figure 8 (b), RAE does decrease for our model TCM-LH. Different from the feature based methods, TCM-LH models the retweeting process in Twitter network and predict the cascade growth based on the simulations of this process. So the longer we observe the cascades, the more accurate the estimations of the parameters in our model are, and the better the performance is.

### D. Improving Virality Prediction Using TCM Simulation

Virality prediction at early stage is very useful for many applications such as viral marketing and breaking news detection. A straightforward method is learning the parameters of a cascade at early stage, and then predicting based on the simulations of TCM-LH (the same as Section VI-C). We found in practice this method doesn't work. Because TCM-LH only works when enough data is observed (as shown in Figure 8), but at the early stage, the sizes of most

| Threshold | Measure | Random Guessing | Without Simulation | With Simulation |
|---|---|---|---|---|
| 20 | Recall | 0.4817 | 0.4535 | **0.6254** |
| | Precision | 0.0034 | **0.7285** | 0.5678 |
| | F1 | 0.0068 | 0.5590 | **0.5952** |
| 25 | Recall | 0.5764 | 0.4716 | **0.5808** |
| | Precision | 0.0026 | **0.7500** | 0.6215 |
| | F1 | 0.0053 | 0.5791 | **0.6005** |
| 30 | Recall | 0.4600 | 0.4333 | **0.5667** |
| | Precision | 0.0014 | **0.6915** | 0.6071 |
| | F1 | 0.0027 | 0.5328 | **0.5862** |
| 35 | Recall | 0.4653 | 0.3762 | **0.5446** |
| | Precision | 0.0009 | **0.6909** | 0.5612 |
| | F1 | 0.0019 | 0.4872 | **0.5528** |
| 40 | Recall | 0.4545 | 0.2424 | **0.4697** |
| | Precision | 0.0006 | **0.6667** | 0.4247 |
| | F1 | 0.0012 | 0.3556 | **0.4460** |

Table IV
THE RESULTS OF VIRALITY PREDICTION FOR DIFFERENT SOLUTIONS ON DIFFERENT THRESHOLDS.

cascades are very small, which makes it hard to estimate the proper parameters of cascades at early stage. However, we can make use of the simulations of our model TCM-LH to improve vitality prediction by remedying the imbalance issue in cascade data.

In particular, we conduct the following virality prediction task: at time $T_0$ = 5 minutes, predict whether one cascade goes viral in the future, which means its cascade growth exceeds a prefixed threshold. The skewness of distribution of the cascade sizes (See Figure 5 (a)) is a challenge of this prediction task. In our data set, only 1‰ cascades grow over 35. Our TCM-LH model can be used to make this data set less skew by adding several simulated viral cascades into it. In this experiment, we randomly choose 100,000 cascades from our data set. 10-fold cross validation are used. There are two types of training sets : (I) original training set without simulated viral cascades, (II) training set with simulated viral cascades: we choose the top 100 cascades from training set, then the parameters of these cascades are learned and each of them are simulated 100 times, and at last in all 10,000 simulated cascades are added into this training set. We use the features in Section VI-C to learn the logistic regression classifier. Table IV presents the prediction results of different solutions, from which we can observe that, after adding the simulated viral cascades into the training set, although the precision is not as good as before, the higher recall and F1 score are achieved. It shows that benefiting from the simulated viral cascades, the classifier can identify around 20% more viral cascades.

## VII. RELATED WORK

Information diffusion or information cascade in networks has been studied for a long time. Especially after the boom of online social networks, it gets a lot of attention from the computer science researchers. We summarise the related works as follows.

**Cascade Modelling** The most influential cascade models are Threshold Model[14], Cascade Model [11] and tons of their extensions (e.g. [19], [17], [25], [21]). In the original Threshold Model and Cascade Model, time is not a factor of concern. In recent years, many works have integrated the time dimension into the cascade model for different purposes. We call them time-aware cascade models. [23] models how information diffuse in networks when external out-of-network sources exist. [12] and [13] infer the unobserved networks. [7] uncovers topic-sensitive information diffusion networks. [6] studies scalable methods for influence estimation in diffusion networks. What is common in these works is that a hazard function of time is used to model how information diffuse over time in networks. We follow this line – using a general time-aware cascade model to describe how each particular cascade grows over time. However, different from these existing works, we focus on the possible choices of hazard functions in the setting of Twitter and make effort to fit model to real cascade data. Particularly based on our observations on user retweeting behaviour, we design a specific hazard function for Twitter network.

**Cascade Prediction** There is also a branch of works on cascade prediction, including cascade size prediction (e.g. [15], [20], [29], [3]), and viral (or outbreaking) cascade prediction (e.g. [18], [4]). In most of these works, the prediction is based on the features learned from the training cascades. Due to lack of data at the early stage, our model is not suitable to do the cascade prediction task directly. However, we improve the prediction performance in an "orthogonal" direction – making use of the simulations of our model to remedy the imbalanced cascade data, which can potentially benefit these feature based prediction solutions.

**Others** Works such as [24] presents the differences in the mechanics of information diffusion across topics, [10] gives an empirical study on the structure of online diffusion networks, [5] studies characteristics of large Facebook cascades, [8] empirically studies rumour cascades on Facebook. These works provide us all kinds of insights about how cascades develop in networks. In this paper, we focus on the effect of time.

## VIII. CONCLUSIONS

In this paper, we used a general time-aware cascade model to describe the dynamic process of growing cascades in social networks over time. Based on this general model, a concrete model TCM-LH was designed for the retweeting cascades in Twitter. We conducted extensive evaluations based on a large real Twitter data set with over two million retweeting cascades. Our experiment results show our proposed TCM-LH fits the real cascade data better than other baselines in terms of model fitting. We also empirically showed that our proposed TCM-LH could benefit applications such as virality prediction.

## ACKNOWLEDGMENT

This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office, Media Development Authority (MDA).

## REFERENCES

[1] P. Bao, H. Shen, J. Huang, and X. Cheng. Popularity prediction in microblogging network: a case study on sina weibo. In *WWW (Companion Volume)*, pages 177–178, 2013.

[2] A.-L. Barabasi. The origin of bursts and heavy tails in human dynamics. *Nature*, 435(7039):207–211, 2005.

[3] J. Cheng, L. A. Adamic, P. A. Dow, J. M. Kleinberg, and J. Leskovec. Can cascades be predicted? In *WWW*, pages 925–936, 2014.

[4] P. Cui, S. Jin, L. Yu, F. Wang, W. Zhu, and S. Yang. Cascading outbreak prediction in networks: a data-driven approach. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, pages 901–909, 2013.

[5] P. A. Dow, L. A. Adamic, and A. Friggeri. The anatomy of large facebook cascades. In *Proceedings of the Seventh International Conference on Weblogs and Social Media, ICWSM 2013, Cambridge, Massachusetts, USA, July 8-11, 2013.*, 2013.

[6] N. Du, L. Song, M. Gomez-Rodriguez, and H. Zha. Scalable influence estimation in continuous-time diffusion networks. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3147–3155, 2013.

[7] N. Du, L. Song, H. Woo, and H. Zha. Uncover topic-sensitive information diffusion networks. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2013, Scottsdale, AZ, USA, April 29 - May 1, 2013*, pages 229–237, 2013.

[8] A. Friggeri, L. A. Adamic, D. Eckles, and J. Cheng. Rumor cascades. In *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1-4, 2014.*, 2014.

[9] W. Galuba, K. Aberer, D. Chakraborty, Z. Despotovic, and W. Kellerer. Outtweeting the twitterers - predicting information cascades in microblogs. In *WOSN*, 2010.

[10] S. Goel, D. J. Watts, and D. G. Goldstein. The structure of online diffusion networks. In *ACM Conference on Electronic Commerce, EC '12, Valencia, Spain, June 4-8, 2012*, pages 623–638, 2012.

[11] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters*, 12(3):211–223, 2001.

[12] M. Gomez-Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the temporal dynamics of diffusion networks. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 561–568, 2011.

[13] M. Gomez-Rodriguez, J. Leskovec, and B. Schölkopf. Modeling information propagation with survival theory. In *ICML (3)*, pages 666–674, 2013.

[14] M. Granovetter. Threshold models of collective behavior. *American journal of sociology*, 83(6):1420, 1978.

[15] L. Hong, O. Dan, and B. D. Davison. Predicting popular messages in twitter. In *WWW (Companion Volume)*, pages 57–58, 2011.

[16] D. W. Hosmer Jr, S. Lemeshow, and S. May. *Applied survival analysis: regression modeling of time to event data*, volume 618. Wiley. com, 2011.

[17] M. O. Jackson and L. Yariv. Diffusion on social networks. *Economie Publique*, 16:3–16, 2005.

[18] M. Jenders, G. Kasneci, and F. Naumann. Analyzing and predicting viral tweets. In *WWW (Companion Volume)*, pages 657–664, 2013.

[19] D. Kempe, J. M. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD*, pages 137–146, 2003.

[20] 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.

[21] W. Lee, J. Kim, and H. Yu. CT-IC: Continuously activated and time-restricted independent cascade model for viral marketing. In *ICDM*, pages 960–965, 2012.

[22] R. G. Miller Jr. *Survival analysis*, volume 66. John Wiley & Sons, 2011.

[23] S. A. Myers, C. Zhu, and J. Leskovec. Information diffusion and external influence in networks. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, pages 33–41, 2012.

[24] D. M. Romero, B. Meeder, and J. M. Kleinberg. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In *WWW*, pages 695–704, 2011.

[25] K. Saito, M. Kimura, K. Ohara, and H. Motoda. Learning continuous-time information diffusion model for social behavioral data analysis. In *ACML*, pages 322–337, 2009.

[26] G. Szabó and B. A. Huberman. Predicting the popularity of online content. *CoRR*, abs/0811.0405, 2008.

[27] W. Weibull et al. A statistical distribution function of wide applicability. *Journal of applied mechanics*, 18(3):293–297, 1951.

[28] W. Xie, F. Zhu, J. Jiang, E.-P. Lim, and K. Wang. Topicsketch: Real-time bursty topic detection from twitter. In *ICDM*, pages 837–846, 2013.

[29] T. Zaman, E. B. Fox, and E. T. Bradlow. A bayesian approach for predicting the popularity of tweets. *CoRR*, abs/1304.6777, 2013.