Prediction of Virality Timing Using Cascades in Social Media

Ming Cheung, HKUST-NIE Social Media Lab
James She, HKUST-NIE Social Media Lab
Alvin Junus, HKUST-NIE Social Media Lab
Lei Cao, HKUST-NIE Social Media Lab

Predicting content going viral in social networks is attractive for viral marketing, advertisement, entertainment, and other applications, but is still a challenge in the big data era today. Previous works mainly focus on predicting the possible popularity of content, rather than the timing of reaching such popularity. This work proposed a novel yet practical iterative algorithm to predict the virality timing, in which the correlation between the timing and growth of content popularity is captured by using its own big data naturally generated from users' sharing. Such data is not just able to correlate the dynamics and associated timings in social cascades of viral content, but also can be useful to self-correct the predicted timing against the actual timing of the virality in each iterative prediction. The proposed prediction algorithm is verified by datasets from 2 popular social networks - Twitter and Digg, as well as 2 synthesized datasets with extreme network densities and infection rates. With about 50% of the required content virality data available (i.e., halfway before reaching its actual virality timing), the error of the predicted timing is proven to be bounded within a 40% deviation from the actual timing. To the best of our knowledge, this is the first work that predicts content virality timing iteratively by capturing social cascades dynamics.

Categories and Subject Descriptors: H.4 [**Information System**]: Information System Applications

General Terms: Design, Algorithms, Measurement

Additional Key Words and Phrases: virality timing, virality prediction, social cascade, social media and networks

ACM Reference Format:
ACM Trans. Multimedia Comput. Commun. Appl. 0, 0, Article 0 (YYYY), 23 pages.
DOI:http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Social media and networks are very influential in our society today; sharing text, video and other types of content is now a daily lifestyle for many individuals. Digg and Twitter are popular social networks to spread breaking news and interesting content [Bakshy et al. 2012] [Cha et al. 2009], where users share these pieces of content and cause some of them go viral. Fig. 1 (a) shows the number of votes for a popular piece of news shared in Digg, which was voted for at least 10000 times in the first 140 minutes. Fig. 1 (b) shows a popular mobile game, Angry Birds, which had over 500 million downloads in about 2 month; and Fig. 1 (c) is a globally popular song, Gangnam Style, which received over 1 billion views on YouTube in a year. However, such virality (i.e., the tendency of content to be viral), is not guaranteed as only a few pieces of content can

This work is supported by the HKUST-NIE Social Media Lab, HKUST.
Author's addresses: HKUST-NIE Social Media Lab., Rm 3117, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

DOI:http://dx.doi.org/10.1145/0000000.0000000

attract millions, and thus it is challenging to know when it will happen. Predicting virality is challenging due to the nature of the content and user sharing behaviors, as well as users' influences in social networks. So far, there is no universal method that can generically predict content virality and its timing in social networks. Content virality, or the popularity, can be measured as a targeted number of views of, shares of or votes on content.

With the ubiquity of smartphones and wearable devices today, content sharing is no longer limited by time and location. Users seamlessly generate and share content with others on social networks all the time, and a big amount of data (big data) about much social information such as friendships, user interests and sharing timings is naturally generated along with these sharings. This kind of data indeed provides useful information for precise virality prediction that was hardly achievable before the era of big data. However, how to utilize such data is still being investigated today due to its

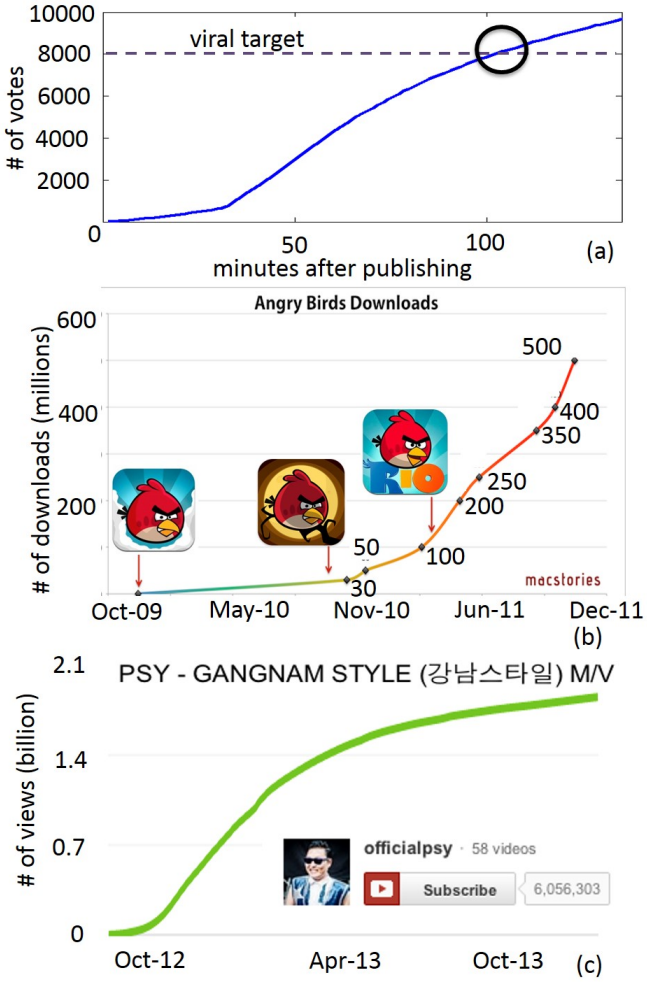

Fig. 1: Examples of viral content: (a) popular story on Digg, (b) Angry Birds (source:www.macstories.net), and (c) Gangnam Style (source:www.youtube.com).

volume, variety, velocity and other challenges. Most of the previous works focus on the final popularity of viral content, but there are only a few investigations into the timing to reach such virality [Cheung et al. 2013][Junus et al. 2015]. For applications in viral marketing, advertising, entertainment, etc., predicting the virality timing is very important to the business needs or even the media risks. For example, if the popularity of a product is predicted to reach the viral target in a social network after 2 months, it will be cost-effective to adjust the related marketing campaign not to finish until then. Hence, the proposed algorithm aimes at predicting the timing of content virality reaching a given viral target. This work is motivated to address the following issues:

— How to predict the timing for content virality?
— How will the big data be useful for such a prediction?

To let the prediction results be practical for real-world applications with an acceptable error range, the novelty of this work focuses on how such a prediction of virality timing is possible with early data on content virality by utilizing the data naturally generated in social cascades. The contributions of this paper can be summarized as follows:

(1) intensively measured social cascades of viral content from Twitter and Digg to describe and evaluate correlations between cascade dynamics and virality timing;
(2) proposed a practical self-correcting iterative algorithm to predict virality timing by using data that describes the related cascade dynamics;
(3) extensively verified the proposed algorithm with data from Twitter and Digg, and with simulated data with extreme conditions to prove the prediction's effectiveness.

This paper starts with the related work in Section II. The definition of social cascade, and a discussion of its relations to content virality are presented in Section III. Characteristics of social cascades for viral content are evaluated by massive measurements in Section IV. The proposed iterative prediction algorithm of virality timing is proposed in Section V, and is followed by experimental results in Section VI to prove its effectiveness. Section VII concludes the paper with possible long-term impacts.

2. RELATED WORK

Virality prediction has become one of the most trendy research topics recently, in which virality can be measured by the number of views of a video on Youtube, to the number of shares of content on social media. Research works focus on detecting the time of the burst, whether the virality, such as the number of views, will reach a target, or the time to reach a viral target [Cheung et al. 2013]. One of the common methods to predict the virality is to apply the properties of content: the length, title and the category of a video on Youtube [Cheng et al. 2007][Figueiredo et al. 2014][Jiang et al. 2014], the wordings and language of a tweet on Twitter[Jenders et al. 2013][Bandari et al. 2012], and the structure of the social graph [Hong et al. 2011][Jenders et al. 2013]. Based on these properties, researchers implemented classification models [Hong et al. 2011][Jenders et al. 2013][Bandari et al. 2012] to obtain viral content. Another common method is to use early popularity, such as number of views/share of a video, to predict the virality. It is proven that the early popularity has a strong correlation with final popularity [Cha et al. 2007][Szabo and Huberman 2010][Figueiredo et al. 2014][Zhao et al. 2015], and the growth of the popularity follows certain distributions [Pinto et al. 2013][Cheng et al. 2007][Yang and Leskovec 2011][Bandari et al. 2012][Bauckhage et al. 2015][Bauckhage et al. 2014][Bauckhage et al. 2013]. By fitting the distributions with early or currently available data, it is possible to predict the future virality. Although promising results have been obtained, this method has no insight into how social media helps the spreading of content.Additional information on spreading on social media could lead to improved predictions.

The attention of users can lead to higher infection rates, such as, a higher number of searches in Google can lead to a higher number of views in Youtube a day later [Bauckhage et al. 2015]. Social media helps to draw the attention of users through sharing. In [Eysenbach 2011], a higher number of Tweets on Twitter about a paper can lead to a higher citation of the paper. A highly shared video on social media is likely to have a higher number of views [Broxton et al. 2013]. The content shared on social media spreads through links among users, and a cascade is formed. The mechanisms in social media, such as notifications, are proven to be effective in information diffusion [Cha et al. 2008][Yu and Fei 2009][Hodas and Lerman 2014][Broxton and Wattenhofer 2013]. The growth of a cascade follows a different structure and life time, depending on the virality of the cascade [Goel et al. 2013][Wu and Huberman 2007]. The number of shares, the size of a cascade, or the burst time, can be used as a measurement of virality, and it is proven that the structure of the social graph [Lerman and Ghosh 2010][Leskovec et al. 2007][Cheng et al. 2014][Lehmann et al. 2012], interactions among users [Galuba et al. 2010][Shamma et al. 2011] and content properties [Tang et al. 2009][Goyal et al. 2010][Guerini et al. 2013] are highly related to the growth rate of a cascade. Based on these observations, researchers apply classification [Cui et al. 2013][Wang et al. 2015][Yu et al. 2014], previous sharing [Saito et al. 2008] and model fitting [Sun et al. 2009][Wang et al. 2015][Yu et al. 2015]. However, it is not clear how the time to reach a viral target can be predicted in these works using cascade information from a piece of content. In [Yu et al. 2015], the authors predict the size of a cascade using the behaviors of nodes, but only focus on predicting the virality of a cascade. In [Cha et al. 2008], the authors applied basic reproduction number [May and Lloyd 2001], or the expected number of infections by a new infection, to capture these factors, and it has been proven that the basic reproduction number can be well modeled by the network structure, interactions among users and the content properties. It is interesting to investigate how basic reproduction number helps to predict the virality of content.

Our previous work [Cheung et al. 2013] incorporated cascade dynamics to predict the virality timing of a single cascade for a given viral target. The required information is the infection durations, the time taken for a node to be infected, and the cascade growth, which is commonly available on any social network. This paper extends [Cheung et al. 2013] by measuring and analyzing how the user behaviors affect the prediction, and predicts the virality timing of a piece of content from all of its associated social cascades, and does so through an iterative and self-correcting algorithm using big data. This paper is different from our previous works [Cheung et al. 2013][Junus et al. 2015] in the following: 1) conducted in practical situation to predict the virality of content which consists of multiple cascades, that is not possible in [Cheung et al. 2013]; 2) intensively measured the infection duration, that is not reported and studied in our previous works, and is not obvious but necessary for virality prediction of content, which consists of multiple cascades; 3) formulated mathematically the approach in [Cheung et al. 2013], and proposed a practical algorithm and system for virality prediction; 4) added one more dataset, Twitter, to verify the proposed algorithm, which is not studied in our previous works, and studied the infection duration.

3. SOCIAL CASCADE

In this section, the concept of social cascade is described, followed by the basic reproduction number and its relation to the growth of content popularity.

3.1. Definition

A social cascade is a process of information diffusion in a social network [Cha et al. 2008]. An example of a social cascade in Flickr is shown in Fig. 2, in which each node

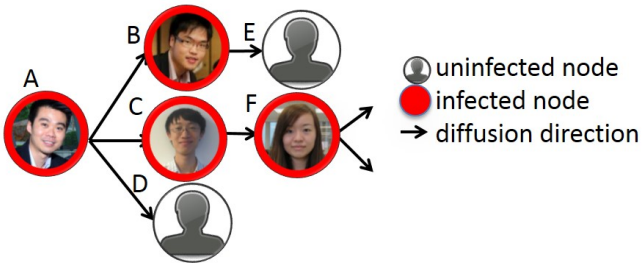

Fig. 2: Social cascade.

(i.e., a user) is sharing a common photo with connected nodes (i.e., his/her friends) in a social graph. Node A, who first shares a photo P, is the initial node (or the seed) and has generation index 1 in a social cascade. To form a social cascade, two users must have a social connection (e.g., friends or followers) first, either bi-directionally or uni-directionally, in a social graph. Nodes B and C, who have social connections with node A, reshare photo P after node A does. Both nodes B and C have generation index 2 in this cascade, and node F is infected with generation index 3 through its existing social connection with node C. The infection takes varying time durations to occur. For example, node B may need more time to read the content than node C before sharing it with others. Such infection duration variations depend on cascade dynamics. [Cha et al. 2008] investigated the social cascade of images in Flickr, and concluded that viral spreading can be more effective than physical infectious diseases like measles. In online social networks like Digg and Twitter, friends of a voter/poster can see the story/tweet from someone, and can vote for the story or retweet the same content to others through their existing social connections. A lot of data about user sharing will thus be available for virality investigation in social cascades.

3.2. Basic Reproduction Number

Basic reproduction number, R_0, is defined as the expected number of secondary infections due to an infected node in a cascade. In epidemiological models, if $R_0 > 1$, one infected node will infect more than one nodes as shown in Fig. 3 (a) and the solid line in Fig. 3 (d), and the cascade size will grow fast. R_0 = 1 is the critical case where the cascade grows linearly with the generation, as shown in Fig. 3 (c) and the dotted line in Fig. 3 (d). If $R_0 < 1$, the number of infected nodes will decrease for each subsequent generation, as shown in Fig. 3 (b), and the cascade will fizzle out before it can infect many nodes, as per the broken line in Fig. 3 (d).

R_0 can be approximated if the number of uninfected nodes is much larger than the number of infected nodes and if the network is highly infectious in nature [Cha et al. 2008]. Since the proposed algorithm focuses on the initial fast growing stage, the first assumption holds. [Broxton and Wattenhofer 2013] reports that online content generally gains traction and fades quickly, but as reported in [Szabo and Huberman 2010], their popularity can still grow even after a long time, e.g., years, after their cascades start. All nodes are assumed to be infected eventually, thus fulfilling the second assumption. The theory of epidemiological models from [May and Lloyd 2001] shows that the basic reproduction number in a network is given by:

$$R_0 = \rho_0(\overline{k^2})/(\overline{k})^2, \tag{1}$$

where $\rho_0 = \beta\gamma\overline{k}$. β and γ are the transmission rate and infection duration, respectively, while k is the node degree, and $\overline{k}$ represents the mean value of the node degree. Eq. 1

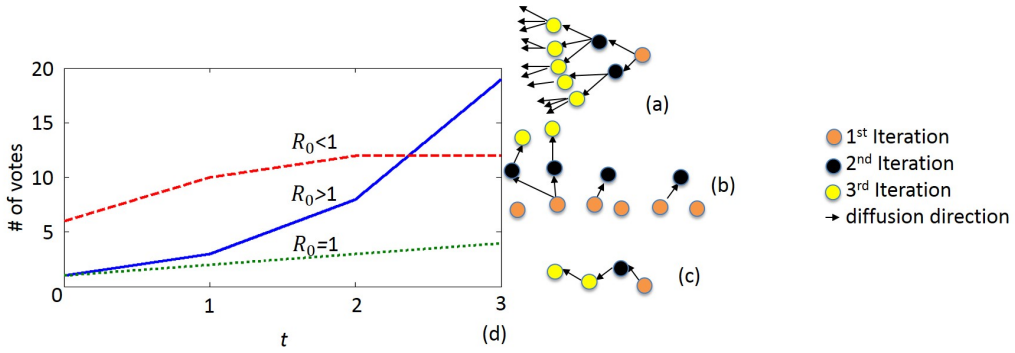

Fig. 3: Social cascades when (a) $R_0 > 1$, (b) $R_0 < 1$, (c) $R_0 = 1$, and (d) the prediction curves.

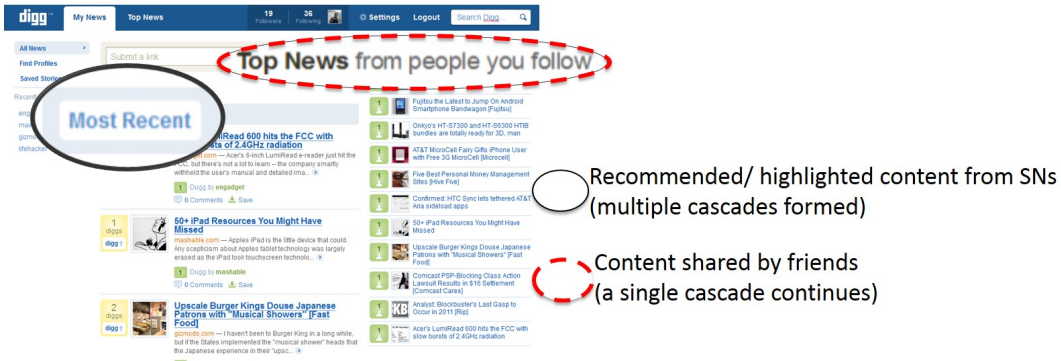

Fig. 4: Content spreading caused in multiple cascades by highlight/recommendation mechanisms in a social network (as circled in the solid line), rather than shared from friends (as circled in the broken line).

is proven to accurately model more than 1000 shared pictures in Flickr over different social cascades. The growth of a cascade is affected by different factors: content, seed, resharer, cascade structure and temporal features [Cheng et al. 2014], which can be captured by R_0. For example, more viral content has a higher β. The properties of the seed and resharers, as well as the cascade structure, can be captured by k in Eq. 1. [Cha et al. 2008] states that the basic reproduction number R_0 can be obtained by counting the number of infected nodes directly from the seed. For example, $R_0(1)$ and $R_0(2)$ in Fig. 3 (a) are 2 and 2.5, respectively, while $R_0(1)$ and $R_0(2)$ in Fig. 3 (b) are 0.667 and 0.5, respectively. A viral piece of content will infect more nodes in one generation, resulting in a higher R_0. The initial R_0 cannot capture the network structure and the cascade dynamics as they keep changing as the cascade grows. R_0 should be re-calculated when more data is available, and thus will be updated from collected data at each prediction computation in our proposed algorithm.

3.3. Multiple Cascades of Viral Content

Top news from followed people on Digg in Fig. 4 allows an ongoing cascade to grow. However, a piece of content may be spread among users through multiple cascades through recommendations, as in Fig. 4, or even beyond social network mechanisms.

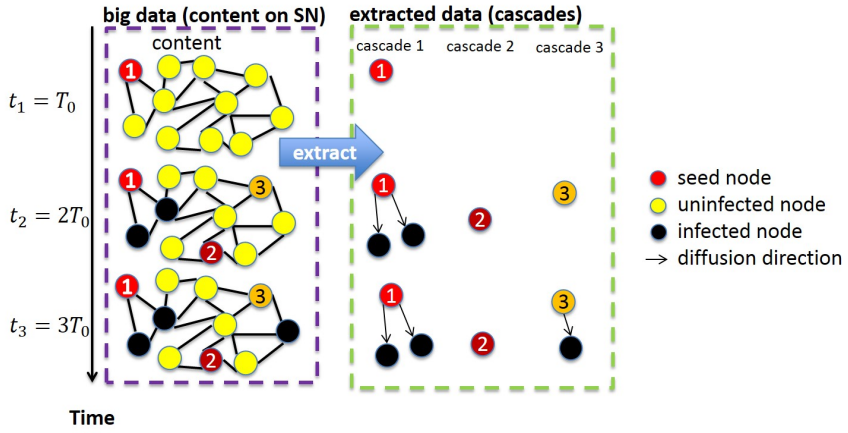

Fig. 5: Extracting social cascades from big data.

Those users are not infected through a single cascade, so they are considered as the seeds of new cascades. All cascades of the same content are changing individually, and should be aggregately evaluated for the growth of the content virality. Fig. 5 is an example of a piece of content shared on a social network. At t_2, the same piece of content has generated three cascades. In predicting the piece of content's size, cascades 1, 2, and 3 need to be considered. However, cascades may have different dynamics, as captured by R_0. Calculating R_0 in each cascade may not properly capture the cascade growth, so R_0 is calculated with all the cascades of the content, and is used to represent the content. Thus, predictions are calculated for all the cascades separately, and the content virality is predicted from the collective behavior.

4. DYNAMICS OF SOCIAL CASCADES FOR VIRALITY

In this section, a detailed analysis of cascade dynamics with respect to content virality will be introduced. The properties of the datasets will be measured first, followed by the properties of cascades.

4.1. Datasets

Two datasets from social networks, Digg, and Twitter, are evaluated in this work. The first social network, Digg, is a website where users can submit stories/news, and their friends can track the submissions and votes. A newly submitted story can be voted on, and becomes visible to the voter's friends so that they can also vote. The dataset was scraped and evaluated by [Lerman and Ghosh 2010] in 2006 and 2009 with Digg API, and includes submitter IDs, submission time, list of voter IDs and their friends' IDs, and the time of each vote. There are 3,018,197 records involving 3,553 stories and 139,409 users, who were anonymized before sharing with this works [1]. Similar to the definition of a social cascade in [Cha et al. 2008], a user must be a follower of a voter before the user can vote for the same story. The second dataset was scraped from Twitter in 2012 by [Baos et al. 2013] with Twitter API, and includes tweeter ID, tweet time, retweeter ID, and retweet time. There are a total of 1,431,573 collected tweets on the Spanish protest, the 15-M movement, with 398,790 users, gained by col-

[1] data available at: http://www.isi.edu/ lerman/downloads/digg2009.html

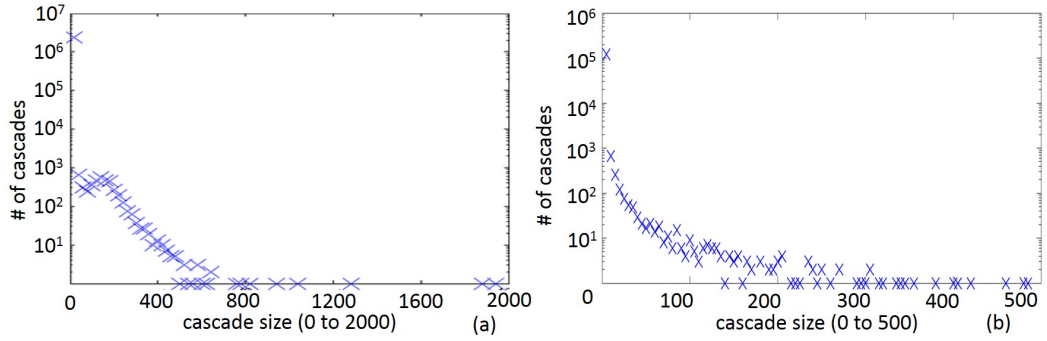

Fig. 6: Cascade size in: a) Digg, b) Twitter.

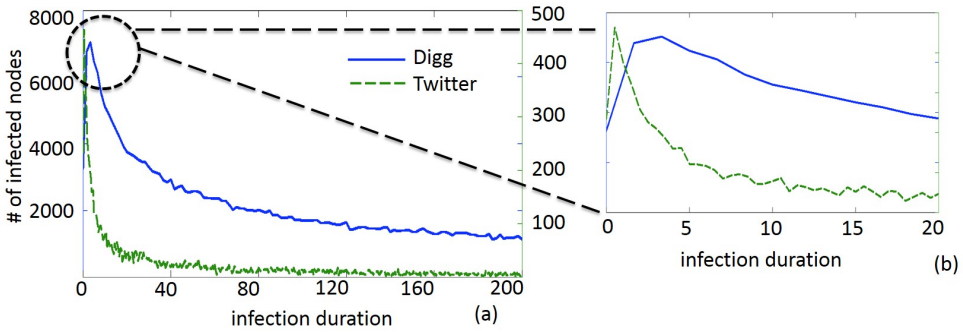

Fig. 7: Infection duration (minutes) vs no. of infected nodes in the first: (a) 200 min; (b) 20 min.

lecting tweets with hashtags related to the movement. User relationships are captured through retweets and mentions to make sure that they are infected by a social cascade. The Twitter data is collected using the streaming API, which only allows no more than 1% of all tweets in the Twitter public timeline. The actual number of tweets should be much higher, but the lack of control on how APIs return tweets may lead to underestimating the size of cascades. However, it is still a good source for the experiment as the cascade properties can be identified as in Digg. The properties of cascades on Digg and Twitter are discussed in the next subsection.

4.2. Dynamics of Cascades

In this subsection, the dynamics of the cascades are studied. Fig. 6 shows the histogram for the cascade sizes for both datasets, which exhibit a common behavior in both datasets: most of the cascades are small in size, and only a few of them are large. This confirms the previous discussion that content virality can be caused by multiple cascades. If that is the common situation for most content in all social networks, then at least one of the required steps in the proposed prediction algorithm is to identify the large cascades and process their aggregate impact on the virality timing due to their significant growths.

It is interesting to observe the infection duration of the cascades. This is the elapsed duration from a node sharing a piece of content to the moment its neighboring node

shares the same content, i.e., the time required for a user to read and share the news on the social network. A measurement is carried out to obtain these dynamics in a social cascade. Fig. 7 (a) shows a distribution of the duration required for a cascade to infect potential nodes. It is observed that the peak of the infections occurs at the early stage, that is, a short period of time after a node is exposed to a new piece of content. Based on the results for these datasets, it is hard to conclude that the distribution follows a distribution that most infections occur at the beginning of the cascade, and it is zoomed in, as shown in Fig. 7 (b). There will be fast-growing stages in the popularity growth of viral content, and infection durations in these stages are small. Since this paper focuses on the fast-growing stage, using a deterministic approach in obtaining the infection durations can reasonably estimate those for cascades in this stage.

5. THE PROPOSED PREDICTION ALGORITHM

The focus of this work is to predict the time needed for content to reach a viral target, i.e., the targeted population size. This is achieved by a 5-stage (stage A to stage E) algorithm, as shown in Fig. 8, an iterative algorithm using real-time big data about multiple cascade dynamics with changing $R_0(t)$ evaluated in each prediction at time t. The first stage, stage A, is the collection and processing of the data, in which cascades of a piece of content are extracted from the social network without interruption and related cascade information is gathered. For each cascade, the growth is predicted iteratively to capture the dynamics of the cascade and network, as in stages B and C. Those predictions on cascades of the content are summed up and the predicted time for the content to reach the viral target is obtained by a root-finding method, as in stage D. Finally, the algorithm loops back to stage B, with updated extracted data after a predefined cycle, T_o, or is stopped when stopping conditions are fulfilled as in stage E. It is summarized in Alg. 1. Each stage is described in detail below.

Algorithm 1 The proposed prediction algorithm

Input: viral target N and targeted content
Output: predicted timing of virality $t(N)$ of the targeted content
while true **do**
  if $N(t) \geq N$ or $t \geq T_{out}$ **then** break;
  end if
  obtain $L(t)$ cascades of the targeted content at time t
  initiate future time t' = 24 hrs
  while true **do**
    initiate $N_0(t, t')$=0
    for each cascade a in targeted content **do**
      obtain the infection durations of infected nodes at t
      obtain the optimal duration from those infection durations
      predict the number of infections at t', $n'_a(t, t')$
      $N_0(t, t') = N_0(t, t') + n'_a(t, t')$
    end for
    if $N_0(t, t')$ reaches targeted resolution **then** break;
    else t' = new future time as Alg. 2
    end if
  end while
end while

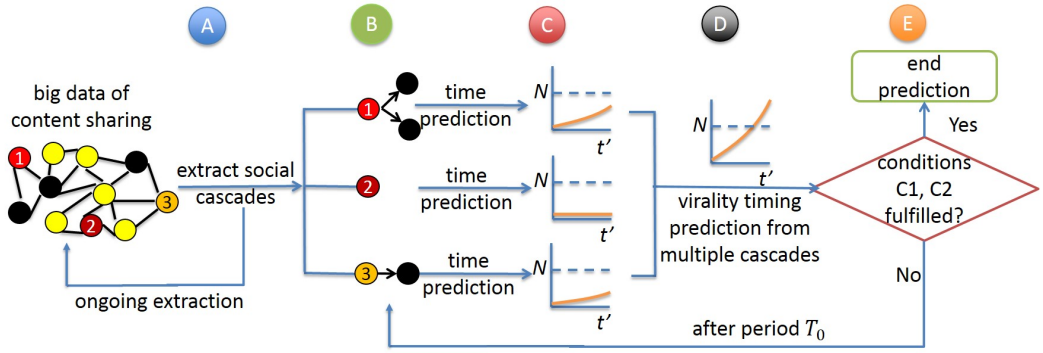

Fig. 8: Stages of the proposed algorithm.

5.1. Extracting Multiple Social Cascades of Viral Content - Stage A

Parameters of social cascades that are necessary for the virality timing prediction in later stages are extracted in this stage. Fig. 5 is an example of multiple cascades extracted from the data in a social network. The left-hand side shows how cascades of content spread on the social network with a group of users, and the right-hand side shows the extracted cascades. The first nodes in cascades 2 and 3 are not infected through a social cascade, but through other mechanisms in the social network, e.g., a highlighted/recommended post on the front page of a social network, as in Fig. 4. Once a node is infected, its neighbors will read through the news feed and decide if they will share the content. The y-axis represents the timing of the sharing behaviors in the social network at time t_1, t_2 and t_3. In each prediction cycle, $N(t)$, the number of infected nodes of content at time t, and the total number of cascades for the content at time t, $L(t)$, are extracted. For example, in Fig. 5, $N(t_3)$ and $L(t_3)$ are 6 and 3 respectively. $g_a(j)$ is the generation index of node j in cascade a. The number of newly infected nodes at time t for cascade a, $\Delta n_a(t)$, and $R_0(t)$ are also counted from the extracted data. The infection duration of each node in the cascade a is recorded, and the set of infection durations, $D_a(t)$, is updated. This process is carried out continuously as in Fig. 8. Table I summarizes the extracted parameters for the prediction.

5.2. Estimating Key Dynamics of Cascades - Stage B

In each iteration, the number of infected nodes of individual cascades increases as more data becomes available. A key parameter for the prediction is the cascade duration, i.e., the duration of time for infections in one generation to occur. The cascade duration $d_a(t)$ should be recalculated such that most of the nodes' generations are correct for cascade a at current time t, as illustrated in Fig. 9. The first cascade and the extraction start at t_0, and the first evaluation on the key cascade dynamic is from t_1 to t_3, and so on. In Fig. 9 (a), d_1 at t_3 is calculated such that most of the infections are in the right generation except the unfilled node, which takes much longer to be infected. Generation indexes of many infected nodes are incorrectly identified if $d_a(t)$ cannot capture the infection duration, which in turn reduces the accuracy of the prediction, as shown in Fig. 9 (b), with duration d_2. There are many unfilled nodes, which means that many generation indexes are incorrect. Eq. 2 illustrates this concept. Case 1 of Eq. 2 represents a node's generation index correctly captured by some duration. However, the duration may not estimate all the generation index correctly, as illustrated by the

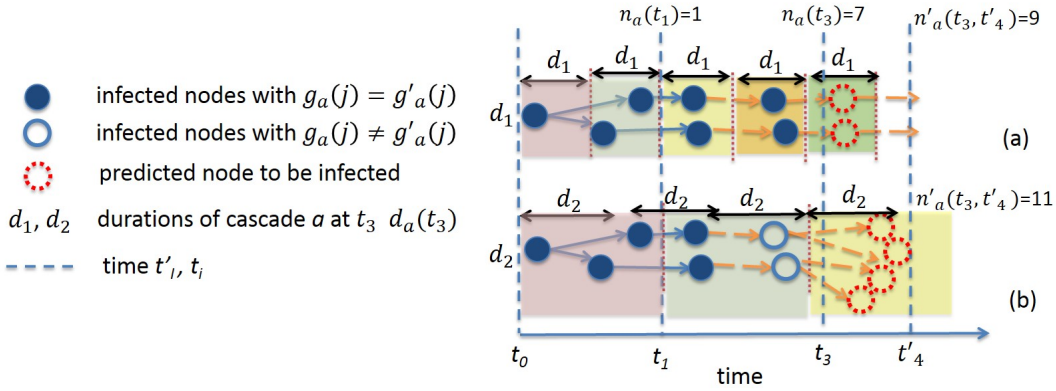

Fig. 9: Cascade a with different values of $d_a(t_3)$ and the predicted number of infected nodes for cascades a: (a) d_1, (b) d_2.

unfilled nodes in Fig. 9 (b). This is expressed in case 2 of Eq. 2.

$$\begin{cases} g_a(j,d) = g'_a(j,d) & case\ 1 \\ g_a(j,d) \neq g'_a(j,d) & case\ 2 \end{cases} \tag{2}$$

An indicator function $I(j,d)$ is defined below to evaluate whether an infected node is captured with the correct generation index when the prediction algorithm processes the extracted data available in real-time.

$$I(j,d) = \begin{cases} 1 & g_a(j) = g'_a(j,d), \quad case\ 1 \text{ correct generation index} \\ 0 & g_a(j) \neq g'_a(j,d), \quad case\ 2 \text{ wrong generation index} \end{cases} \tag{3}$$

Therefore, a suitable cascade duration $g_a(t)$ that is used to compute the virality timing should be a duration that gives the highest count of nodes with the correct generation indexes at time t. Such a cascade duration can be identified from the collection of infection durations in a cascade learned from the data extracted so far at time t:

$$d_a(t) = \arg \max_{d \subset D_a(t)} \sum_{j=1}^{n_a(t)} I(j,d), \tag{4}$$

where $D_a(t)$ is the set containing all possible choices of cascade durations learned from the collected data so far, $n_a(t)$ is the number of infected nodes at time t of cascade a, and d is a cascade duration to be evaluated for whether it is suitable for prediction.

5.3. Prediction of Virality Timing from an Individual Cascade - Stage C

As discussed earlier, a better basic reproduction number $R_0(t)$ could be calculated from the data scraped at time t as t increases. The predicted total number of infected nodes, $n'_a(t,t')$, in cascade a at time t for a future time t', can be modeled by the sum of a geometric series [Cheung et al. 2013]:

$$n'_a(t,t') = n_a(t) + \sum_{j=1}^{k_a(t,t')} \Delta n_a(t) \cdot (R_0(t))^j, \tag{5}$$

where $n_a(t)$ is the current number of infected nodes, and $\Delta n_a(t)$ is the number of newly infected nodes at time t in cascade a. The maximum value of generation index, $k_a(t,t')$,

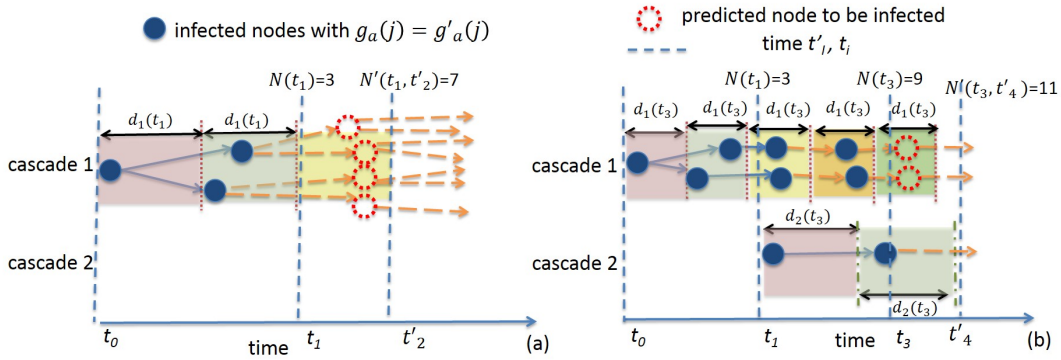

Fig. 10: Predicted cascades growth with suitable durations at: (a) t'_2, (b) t'_4.

in cascade a at time t for a future time t', is:

$$k_a(t, t') = \lfloor \frac{t' - t}{d_a(t)} \rfloor, \tag{6}$$

where $d_a(t)$ is the duration from Eq. 4.

5.4. Prediction of Virality Timing from Multiple Cascades - Stage D

This section presents how $N'(t, t')$ can be calculated based on the result of Stage C, and applies a bisection approach to obtain the future time t' that $N'(t, t')$ reaches N. As discussed in Section 3.3, multiple cascades can spread for the same piece of content simultaneously, and the growth of a cascade is predicted separately in stage C. To predict the virality timing of a given target N at time t, each cascade of the same content is extracted and the growth is predicted at future time t' as shown in stage D of Fig. 8. The total number of cascades of the content at time t, $L(t)$, is updated from the collected data in stage A of Fig. 8. At current time t, the total number of infected nodes $N'(t, t')$ for the content at a future time t' could therefore be estimated as follows:

$$N'(t, t') = \sum_{a=1}^{L(t)} n'_a(t, t'). \tag{7}$$

Fig. 10 illustrates this idea. The growth of some content at t_1 is captured in Fig. 10 (a). There is only one cascade and the growth is predicted accordingly. When more data is extracted at t_3, the growths of the 2 cascades at t', $n'_a(t_3, t'_4)$ are predicted and summed up to calculate $N'(t, t')$. Hence, a predicted timing of virality, $t(N)$, will be the soonest time at which $N'(t, t')$ of the content will reach or go beyond a given viral target N, which could be computed by solving the following:

$$t(N) = \arg\min_{t'} N'(t, t') \geq N \tag{8}$$

Computing $t(N)$ in Eq. 8 can be solved iteratively by Eq. 7 with different values of t', which is a root-finding problem for which a start time is set for the first iteration. The start time depends on the application, such as a month if the viral target is expected to be reached in years. Based on our dataset, the algorithm starts with an initial value of t' as 24 hours (i.e., $t' = 24$ hours). If the initial $N(t, t') < N$, i.e., the viral target can be reached at the future time t' ($t' = 24$ initially), the prediction problem falls into case 1 of Fig. 11 (a), and into case 2 of Fig. 11 (b) otherwise, and t' will be doubled

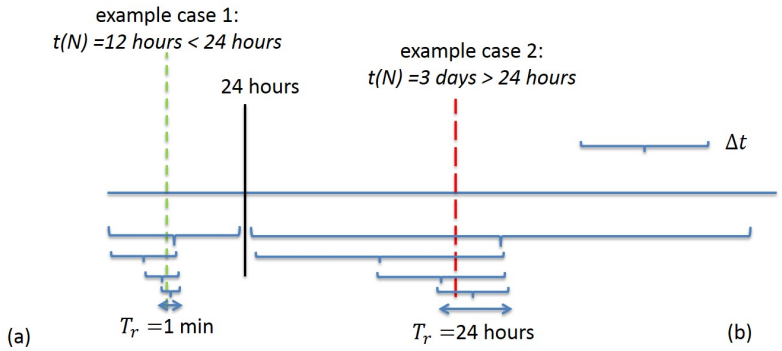

Fig. 11: Finding $t(N)$ through Bisection Algorithm for: (a) resolution of 1 minute when $t(N) < 24$ hours, (b) with resolution of 24 hours when $t(N) > 24$ hours.

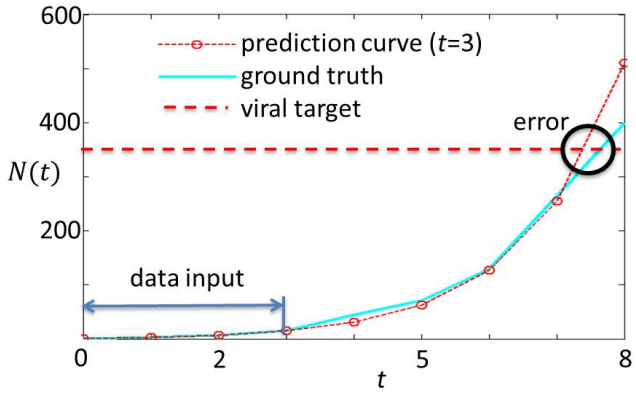

Fig. 12: Example of prediction, a better algorithm takes less data with a smaller error.

until $N(t, t') < N$. This becomes a problem of finding the first converging t' that fulfills the condition, i,e., the change of t', Δt, is smaller than the timing resolution, T_r. The proper value of Δt and ways to converge to a solution of t' are solvable by common root-finding methods like the Bisection Algorithm and Secant method [Wolfe 1959]. If the target is predicted to be reached after a few days or even weeks (case 2), a lower resolution is preferable: a high resolution will consume a lot of computing resources when such resolution is not required. If the viral target can be reached within a day (case 1), the algorithm needs to be more precise in determining the virality timing. T_r can be another number depending on the application, but still fall to case 1 and case 2, or even more cases. In the proposed system, T_r, is set to be 1 minute in case 1 and 1 day in case 2. The bisection method is terminated when Δt is smaller than T_r. The detailed steps are shown in Algorithm 2.

Fig. 12 shows an example of prediction. Cascade information from iterations 1 to 3 are input to the algorithm and the prediction curve at iteration 3 is calculated. The prediction error is marked by the circle. A smaller prediction error implies higher accuracy. However, there is a trade-off between the error (deviation of $t(N)$ from the actual virality timing) and the amount of data needed. A good algorithm should bound the error within a low value at a relatively early stage of the virality.

Algorithm 2 Bisection Algorithm to Compute $t(N)$

Input: viral target N
Output: predicted timing of virality $t(N)$
Initialize $t' = 24\,hrs,\ \ t'' = 0;$
$find\ N(t');$
if $N(t') == N$ **then** ▷ viral target is reached at exactly t'
  $t(N) = t';\ break;$
end if
if $N(t') < N$ **then** ▷ viral target is reached before t'
  $T_r = 1\,day;$
else
  $T_r = 1\,minute;$
end if
while $N(t') < N$ **do** ▷ viral target is reached after t'
  $t'' = t';\ \ t' = 2{*}t';\ \ find\ N(t');$
end while
while true **do**
  $\Delta t = |t' - t''|;$
  $t(N) = t' + \Delta t/2;$
  if $N(t') == N$ **then** ▷ viral target is reached at exactly t'
    $t(N) = t';\ break;$
  else if $\Delta t < T_r$ **then** ▷ timing resolution is reached
    $break;$
  else if $N(t'' + \Delta t/2) < N$ **then** ▷ bisection resolution increased
    $t'' = t'' + \Delta t/2;\ \ continue;$
  else
    $t' = t'';\ \ t'' = t'' - \Delta t/2;\ \ continue;$
  end if
end while
return $t(N);$

5.5. System Stopping Conditions - Stage E

Sometimes, the viral target cannot be reached in a reasonable time, for example, in weeks after a marketing campaign starts. The system can be stopped to save computational resources. As shown in Fig. 8, the prediction algorithm in this stage would be terminated when one of the conditions (C1 or C2) is realized.

C1. the number of currently infected nodes, $N(t)$, is higher than the viral target, N, and no more prediction is needed:

$$N(t) \geq N \tag{9}$$

C2. the current time is longer than a reasonable time T_{out} defined by the user:

$$t \geq T_{out} \tag{10}$$

In applications like viral marketing, the time to reach a viral target is important. For example, if the marketing campaign has run for a reasonably long time T_{out} and the target still cannot be reached, the operation can be terminated to save costs. T_{out} is defined by the user and depends on the application, such as days for a marketing campaign of a movie when it is on show in cinemas. Parameters used in the algorithm are defined in Table I.

Table I: Parameters used in the proposed prediction algorithm

(a): Values extracted from social media	
Parameters	Definition
$R_0(t)$	basic reproduction # at time t
$N(t)$	# of infected nodes at time t for content
$L(t)$	# of cascades in content at time t
$g_a(j)$	generation index of node j in cascade a
$n_a(t)$	# of infected nodes at time t of cascade a
$\Delta n_a(t)$	# of newly infected nodes at time t for cascade a
$D_a(t)$	the set of infection durations of nodes in cascade a at time t
(b): System parameters	
Parameters	Definition
N	user-defined viral target (targeted population of some content)
T_{out}	user-defined timeout period
T_o	user-defined prediction interval
(c): Values from computations	
Parameters	Definition
$g'_a(j)$	estimated $g_a(j)$ for a given duration
t'	a future time for the prediction
$n'_a(t,t')$	predicted $n_a(t)$ at time t for future time t'
$N'(t,t')$	predicted $N(t)$ for future time t' for content
$d_a(t)$	calculated cascade duration of cascade a used in time t
$t(N)$	predicted minimum time needed to reach N
$k_a(t)$	computed the maximum generation index at time t for cascade a
(d): Values for evaluation	
Parameters	Definition
t_{GT}	actual time the content reaches N
Er	computed percentage error in prediction

6. EXPERIMENTAL RESULTS

This section evaluates the trade-off between available data and prediction error. A greater amount of data available can reduce the prediction error, but will lose the prediction value as it is closer to the actual virality timing. A good algorithm should give a prediction with a reasonable bounded range of error at the earlier stage of its virality, such that the prediction result could be useful for practical applications. To benchmark the error range, Er is defined as the percentage of deviation from the ground truth (i.e., the actual time to reach the viral target) [Zhang and Rice 2003]:

$$Er = min(\frac{|t(N) - t_{GT}|}{t_{GT}}, 1) \tag{11}$$

where $t(N)$ is the predicted time to reach N and t_{GT} is the ground truth. A smaller Er implies a more accurate virality timing prediction. Note that Er is bounded by 1, as $E > 1$ means the prediction is totally different from the ground truth (e.g., $t(N) = 0$). As a result, the maximum of Er is set to 1 in the experiment. In order to compare the effectiveness of the proposed system, a regression model using Frechet distribution was built based on [Bauckhage et al. 2013]. This distribution fits the characteristic of a social cascade: a social cascade starts with a slow growth, followed by a fast growth, then a long burn-out period. The trend of the curves is fitted with time as the input and the number of infections as the output to estimate the time to reach the viral target. As the range of the distribution is from 0 to 1, the distribution is scaled by the final

number of infections, N_{final}, and fitted as following:

$$N(t) = (e^{-(t/\beta)^{-\alpha}})N_{final}, \tag{12}$$

where β and α are parameters to be fitted. The comparison is conducted with the same procedure, in which the prediction is made when there is more available data, and then is compared with the ground truth. As N_{final} is not available during the prediction, N_{final} is set with different values to test the efficiency of a regression approach.

6.1. Datasets from Online social networks

The proposed algorithm is evaluated with data from 2 popular online social networks, Digg and Twitter. More than 3 million and 1 million sharings are involved, respectively. In particular, the most popular story from each dataset is selected to demonstrate how the algorithm performs. The virality of the stories, e.g., a Digg story reached 1300 votes in less than four hours, makes them good samples for the experiments. The ground truth is based on the actual time that content reaches the viral target, N. The prediction results for the two popular stories in Fig. 13 are for a target of 15000. The results of the two stories show a similar tendency: when more data is available, the prediction is more accurate. In Fig. 13 (a), Er of the most viral content in the Digg dataset with 30%, 60% and 90% available data is 57.2%, 37.4% and 8.2%, respectively. In Fig. 13 (b), Er of the most viral content in the Twitter dataset with 30%, 60% and 90% available data is 36.6%, 10.2% and 1.2%, respectively.

In order to obtain a general performance of the algorithm, the predictions of the most popular pieces of content which reach the viral target 2000 in the two datasets are also evaluated by Er. Any reasonable viral target can give similar results, and of 2,000 is used as an example. There are a total of 293 and 21 pieces of content that reach the viral target on Digg and Twitter, respectively. Fig. 14 (a) shows the result for the Digg dataset, while Fig. 14 (b) shows the result for the Twitter dataset. It is observed that Er decreases with increasing amount of data. In the beginning, there is insufficient data to capture the cascade dynamics. When more data is captured in later iterations, Er is reduced drastically. Fig. 14 shows that with about 50% of the required content virality data available (i.e., halfway before reaching its actual virality timing), the error of predicted timing is bounded within 40% deviation from the actual timing. Although Er drops slower than the beginning when more data is available, one important conclusion about the algorithm is that the algorithm can make the prediction with a relatively small amount of data. As observed, although Er reduces with a

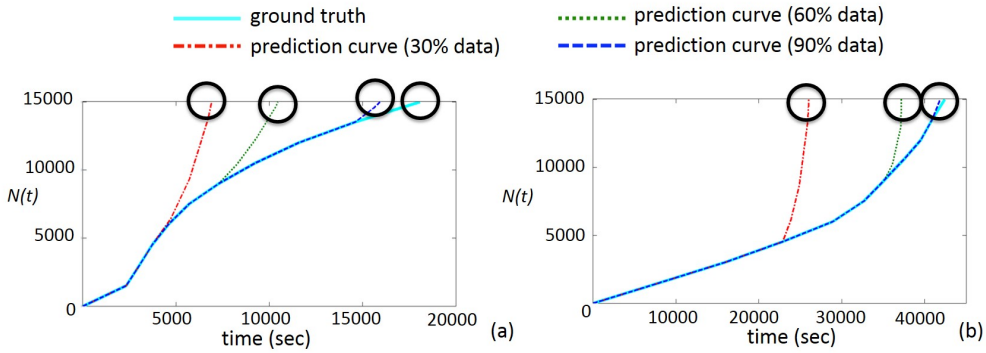

Fig. 13: Time and the number of infected nodes $N(t)$ of the most viral content in (a) Digg, (b) Twitter.

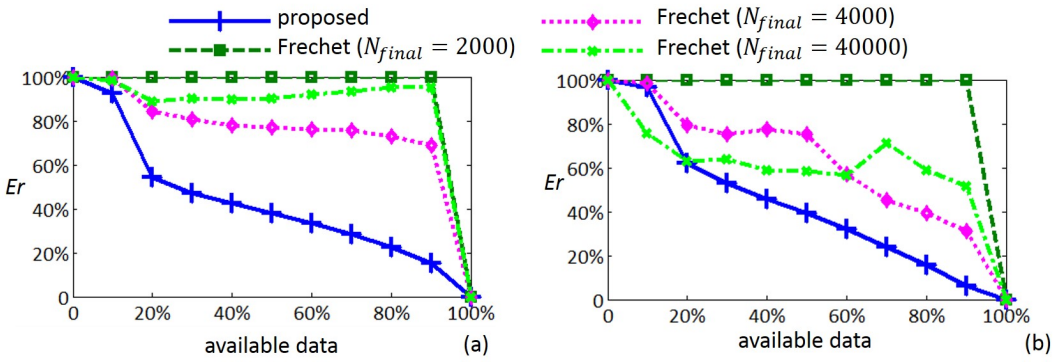

Fig. 14: Er for viral content ($N_{final} \geq 2000$) in (a) Digg, (b) Twitter.

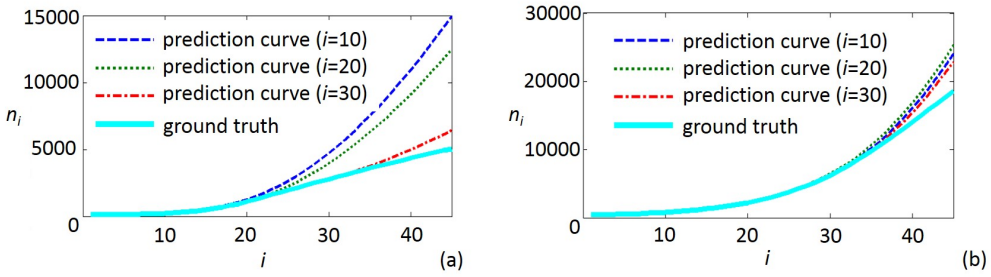

Fig. 15: i and $N(t)$ in (a) the forest fire model, (b) the Kronecker graphs.

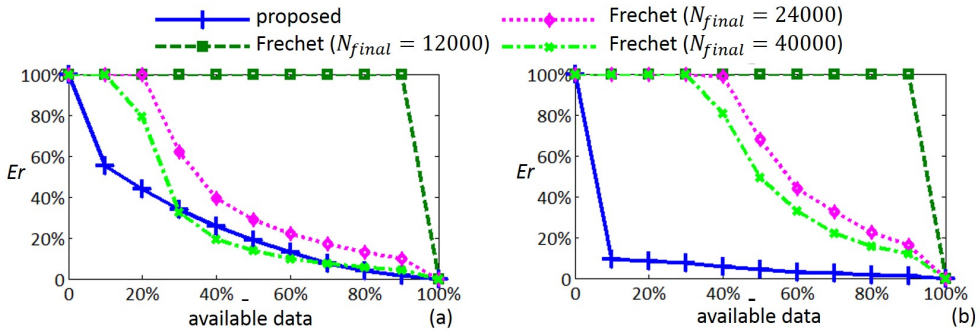

Fig. 16: Er in (a) the forest fire model; (b) the Kronecker graphs.

higher N_{final}, the fitting by Frechet distribution does not give a better performance. Although it has a smaller Er when more data is available, it generally has a higher Er than the proposed algorithm. The experiment is implemented as a Matlab program running on a machine with 8 Gb of memory and an i5-4570 CPU. The runtimes are 3,573.5 s and 87.3 s, or on average, 12.2 seconds and 4.16 seconds per content on Digg and Twitter, respectively.

Synthesized data will be used in the next section to simulate social cascades under extreme conditions with different infection rates and network densities.

6.2. Synthesized Datasets

In order to understand the generic prediction performance for future real-world applications, two synthesized social graphs with different network densities are generated. The first social graph is generated with the forest fire model [Leskovec et al. 2005] while the Kronecker graph [Leskovec et al. 2010] is used for the second social graph, which fulfills properties such as densification laws, shrinking diameters and other properties of social networks. A social graph with 50000 nodes with 268657 edges out of 2.5 billion edges is generated by the forest fire model, and another one with 59049 nodes with 9765625 edges out of 6 billion edges is generated for the Kronecker graph. The network densities are 0.011% (i.e., low density) and 0.163% (i.e., high density) for the forest fire model and Kronecker graph, respectively. Cascades are generated in the two social graphs under high and low infection rates using the Independent Cascade Model (ICM) [Granovetter 1978]. High infection probability is used such that most of the nodes will be infected eventually. Another low infection probability is used such that it is just enough for the content to reach the viral target. Fig. 15 shows the prediction curve and the ground truth of the generated cascades. In these datasets, the actual generation indexes of nodes are known. Predictions are on iterations 10, 20 and 30. A conclusion similar to that for the real datasets can be drawn from the two models: the accuracy of the prediction improves with more iterations, i.e., with more data available.

Similarly, the prediction is evaluated by Er with 20 trials conducted with high and low growth rates, with 20 random nodes in each trial as the seeds to model the effect of multiple cascades on the same piece of content with high and low infection rates. $t(N)$ is calculated and compared with the ground truth in each iteration, and the average Er is obtained. The results are summarized in Fig. 16. There is a similar trend for Er at both high and low infection rates: Er drops dramatically at the beginning. Later iterations have a smaller Er as more data is available. Er as low as 20% can be achieved with only 20% of the extracted data. The two synthesized datasets prove that the proposed algorithm works well on networks with different densities and cascade dynamics. As observed, the fitting by Frechet distribution does not give a better performance. Although it has a smaller Er when more data is available, it generally has a higher Er than the proposed algorithm. Fig. 17 shows the mean and standard deviation of the results of different simulations that start with randomly selected nodes on the synthesis datasets. The curve is Er with a different amount of available data, with the upper and the lower bound of the vertical line one standard deviation from the mean. It is observed that that standard deviation reduced with more data. The experiment is also implemented as a Matlab program running on the same machine. The runtimes are 17.1 ms and 37.1 ms on the content of the fire forest model and Kronecker graph respectively.

In summary, results from 2 popular social networks - Twitter and Digg, as well as results from the 2 synthesized networks with extreme network densities and infection rates, prove that the proposed algorithm only requires 50% of the data (i.e., halfway before reaching its actual virality timing), to achieve an error bounded within a 40% deviation from the actual timing.

6.3. Discussion

In this section, two pieces of content were selected to illustrate how the proposed algorithm could fail, as shown in Fig. 18, and directions are highlighted to improve the proposed algorithm based on the illustration. Fig. 18 (a) shows $N_{(t)}$ with t of the content with the highest Er in Digg. It is observed that $N_{(t)}$ grows fast at the beginning, but becomes slow when it is 20% from the viral target, and a large Er can be observed.

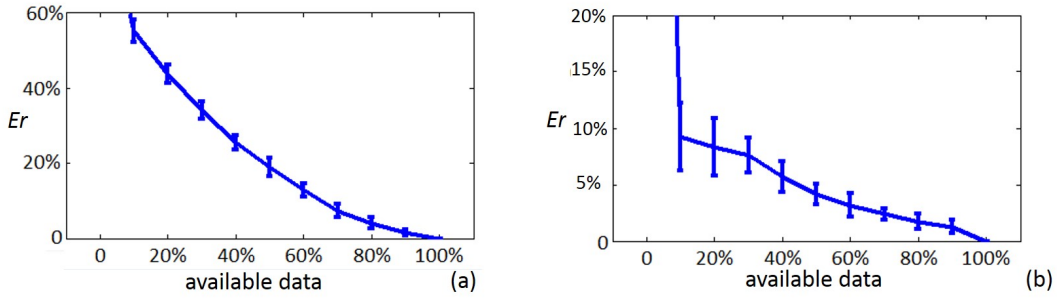

Fig. 17: Er with mean and one standard deviation: (a) the forest fire model, (b) the Kronecker graphs.

One of the reasons is the members of communities that are interested in the content are mostly infected, so the growth rate becomes very small [Weng et al. 2013]. The proposed algorithm can be improved by considering communities. However, if the viral target is set to be 1000, the performance of the proposed algorithm will be good, as the growth at the beginning suits the proposed method well. When the cascade is first started, there are many users that are available for infection, which fits the assumption well. Fig. 18 (b) shows content with a high Er at the beginning, where the growth rate at the beginning is small but it grows very fast later. The growth may be triggered by the growth of social cascades, such as infecting users with a high number of followers. The proposed algorithm can be improved by using outbreak time prediction with social cascades.

It is also interesting to understand when the proposed algorithm gives the best performance. Fig. 19 shows the prediction with the best performance on Digg. There is a viral content, in which the viral target is met within 1 day. It is observed that the number of infections of this content grows exponentially, and the error rate of the prediction is less than 20% in most of the cases. As discussed in the previous section, the proposed algorithm gives the best performance at the fast growing stage, i.e., the viral target is met when the growth of $N_{(t)}$ is still fast growing. A similar analysis is conducted on the content with the best performance on Twitter. A similar observation can be found: the viral target is reached at the fast growing stage. More investigation is needed, such as identifying similar content [Weng et al. 2013] for prediction.

Table II shows the average value of major parameters in Table I. Similar to the previous section, the values are measured from 10% data to 90% data with a bin size of

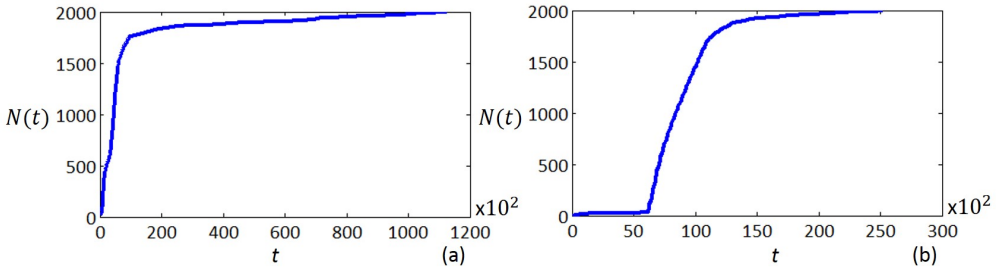

Fig. 18: Examples of content with a high Er, $N_{(t)}$ growth (a) stops by conditions, and (b) the growth is triggered by external events.

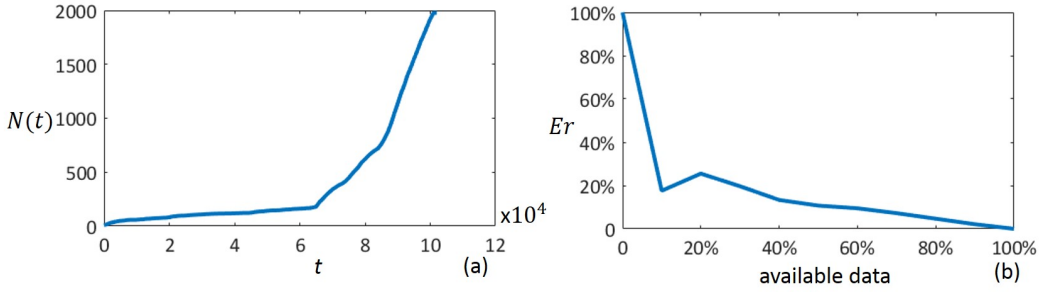

Fig. 19: Content with the best performance on Digg: (a) growth of the content; (b) Er.

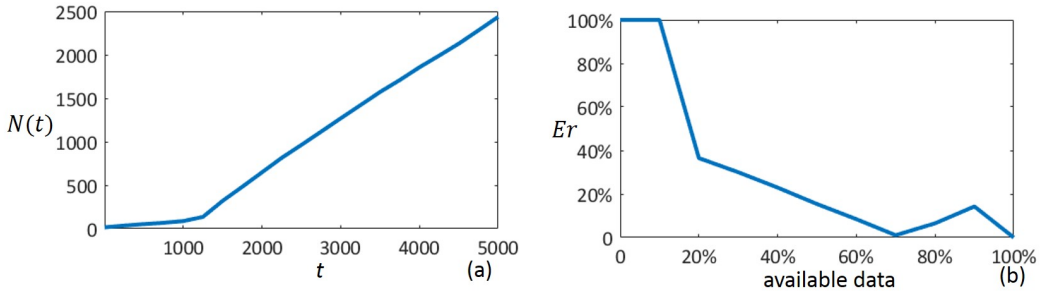

Fig. 20: Content with the best performance on Twitter: (a) grows of the content; (b) Er.

10%. For example, the target in Digg and Twitter is 2,000, the values are measured when the number of infected nodes is 200, 400, 600 and so on. For the time to reach the viral target, t_{GT}, content on Digg takes a longer time than content on Twitter. t_{GT} on the Kronecker graph is reached faster than the forest fire, for both high and low infection rates. One of the reasons behind is the network density, as the networks generated by Kronecker graph has a much higher density (10 times higher). It is also observed that $R_0(t)$ and $\Delta n_a(t)$ are small, and $L(t)$, the number of cascades per content is big on Digg and Twitter. As shown in Fig 7, most of the cascade sizes are small, and only a few of them are big. Note that in the synthesized dataset, the number of cascades is equal to the number of seeds, i.e., 20. As a result, $n_a(t)$ is always equal to 350, as the samples are taken at 1,400, 2,800 and so on. Another parameter in the experiment is $d_a(t)$, the calculated cascade duration of cascade a at time t. As shown in Fig. 7, the peak time of infection is 240 s and 60 s on Digg and Twitter, respectively. However, the values in Table I are are bigger to give the best estimation of the generation index of a node. Note that in the synthesized dataset, the generation index of a node does not need to be estimated.

7. CONCLUSION AND FUTURE WORKS

There have been many previous works that predict content virality (popularity), but only a few on the timing of reaching such virality. This work has intensively measured social cascades of viral content from Twitter and Digg to describe and evaluate correlations between cascade dynamics and virality timing. A practical self-correcting iterative algorithm was proposed to predict the viral timing by using data on cascade dynamics, followed by an extensive verification on the proposed algorithm with data from Twitter, Digg and synthesized datasets with extreme conditions to prove the pre-

Table II: Mean Values of Major Parameters

Parameters	Digg	Twitter	forest fire	Kronecker
N	2,000		14,000	
t_{GT}	122,000s	2,960s	86.7	52.0
t_{GT} (Low infection rate)	NA		59.3	40.6
t_{GT} (High infection rate)	NA		114	63.3
$R_0(t)$	1.01	1.31	1.21	1.214
$L(t)$	846	735	20	20
$n_a(t)$	1.22	1.35	350	350
$\Delta n_a(t)$	44.8	10.9	6.81	13.4
$d_a(t)$	1,007s	278s	NA	

diction's effectiveness. The experiments show that the algorithm can predict virality timing with an error bounded to be within a 40% deviation from the actual virality timing with 50% of the required data (i.e., halfway to the viral target) available, which is better than a regression model fitting using Frechet distribution. These contributions create a long-term impact in this research field and real-world applications for marketing, advertising, entertainment, etc., in today's era of big data.

One of the possible future works is to test the proposed algorithm in other social networks to verify whether the same observation and conclusion can be found in networks with a similar mechanism that forms social cascades. The proposed algorithm may be able to provide insights into the prediction. Another possible future work to improve the effectiveness of predictions could incorporate sharing probability of individual users of content, which is proven to affect information diffusion and social contagion [Pei et al. 2012]. Community structures in a social cascade as well as tie strengths from user interactions may be considered in $R_0(t)$ to achieve a possible better prediction performance. The approaches taken in each stage of the algorithm can also be investigated further to improve the algorithm's effectiveness.

REFERENCES

Eytan Bakshy, Itamar Rosenn, Cameron Marlow, and Lada Adamic. 2012. The role of social networks in information diffusion. In *Proceedings of the 21st International Conference on World Wide Web*. ACM, 519–528.

Roja Bandari, Sitaram Asur, and Bernardo A. Huberman. 2012. The Pulse of News in Social Media: Forecasting Popularity.. In *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media*.

Raquel A. Baos, Javier Borge-Holthoefer, Ning Wang, Yamir Moreno, and Sandra Gonzlez-Bailn. 2013. Diffusion dynamics with changing network composition. *Entropy* 15, 11 (2013), 4553–4568.

Christian Bauckhage, Fabian Hadiji, and Kristian Kersting. 2015. How Viral Are Viral Videos?. In *Ninth International AAAI Conference on Web and Social Media*.

Christian Bauckhage, Kristian Kersting, and Fabian Hadiji. 2013. Mathematical Models of Fads Explain the Temporal Dynamics of Internet Memes.. In *ICWSM*.

Christian Bauckhage, Kristian Kersting, and Bashir Rastegarpanah. 2014. Collective attention to social media evolves according to diffusion models. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*. International World Wide Web Conferences Steering Committee, 223–224.

Tom Broxton, Yannet Interian, Jon Vaver, and Mirjam Wattenhofer. 2013. Catching a viral video. *Journal of Intelligent Information Systems* 40, 2 (2013), 241–259.

Yannet Interian Jon Vaver Broxton, Tom and Mirjam Wattenhofer. 2013. Catching a viral video. *Journal of Intelligent Information Systems* 40, 2 (2013), 241–259.

Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. 2007. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*. ACM, 1–14.

Meeyoung Cha, Alan Mislove, Ben Adams, and Krishna P. Gummadi. 2008. Characterizing social cascades in flickr. In *Proceedings of the First Workshop on Online Social Networks*. ACM, 13–18.

Meeyoung Cha, Alan Mislove, and Krishna P. Gummadi. 2009. A measurement-driven analysis of information propagation in the flickr social network. In *Proceedings of the 18th International Conference on World Wide Web*. ACM, 721–730.

Justin Cheng, Lada Adamic, P. A. Dow, Jon M. Kleinberg, and Jure Leskovec. 2014. Can cascades be predicted?. In *Proceedings of the 23rd International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 925–936.

Xu Cheng, Cameron Dale, and Jiangchuan Liu. 2007. Understanding the characteristics of internet short video sharing: YouTube as a case study. *Multimedia, IEEE Transactions on* 5 (2007), 1184–1194.

Ming Cheung, James She, and Lei Cao. 2013. Predicting Content Virality in Social Cascade. In *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*. IEEE, 970–975.

Peng Cui, Shifei Jin, Linyun Yu, Fei Wang, Wenwu Zhu, and Shiqiang Yang. 2013. Cascading outbreak prediction in networks: a data-driven approach. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 901–909.

Gunther Eysenbach. 2011. Can tweets predict citations? Metrics of social impact based on Twitter and correlation with traditional metrics of scientific impact. *Journal of Medical Internet Research* 13, 4 (Dec 19 2011), e123.

Flavio Figueiredo, Jussara M. Almeida, Marcos A. Gonalves, and Fabrcio Benevenuto. 2014. On the dynamics of social media popularity: a YouTube case study. *ACM Transactions on Internet Technology (TOIT)* 14, 4 (2014), 24.

Wojciech Galuba, Karl Aberer, Dipanjan Chakraborty, Zoran Despotovic, and Wolfgang Kellerer. 2010. Outtweeting the twitterers-predicting information cascades in microblogs. In *Proceedings of the 3rd Conference on Online Social Networks*. USENIX Association, 3–3.

Sharad Goel, Ashton Anderson, Jake Hofman, and Duncan Watts. 2013. The structural virality of online diffusion. *Preprint* 22 (2013), 26.

Amit Goyal, Francesco Bonchi, and Laks V. Lakshmanan. 2010. Learning influence probabilities in social networks. In *Proceedings of the third ACM International Conference on Web Search and Data Mining*. ACM, 241–250.

Mark Granovetter. 1978. Threshold models of collective behavior. *American journal of sociology* (1978), 1420–1443.

Marco Guerini, Jacopo Staiano, and Davide Albanese. 2013. Exploring image virality in google plus. In *Social Computing (SocialCom), 2013 International Conference on*. IEEE, 671–678.

Nathan O. Hodas and Kristina Lerman. 2014. The simple rules of social contagion. *Nature Scientific reports* 4 (2014).

Liangjie Hong, Ovidiu Dan, and Brian D. Davison. 2011. Predicting popular messages in twitter. In *Proceedings of the 20th International Conference Companion on World Wide Web*. ACM, 57–58.

Maximilian Jenders, Gjergji Kasneci, and Felix Naumann. 2013. Analyzing and predicting viral tweets. In *Proceedings of the 22nd International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 657–664.

Lu Jiang, Yajie Miao, Yi Yang, Zhenzhong Lan, and Alexander G Hauptmann. 2014. Viral video style: a closer look at viral videos on YouTube. In *Proceedings of International Conference on Multimedia Retrieval*. ACM, 193.

Alvin Junus, Ming Cheung, James She, and Zhanming Jie. 2015. Community-Aware Prediction of Virality Timing Using Big Data of Social Cascades. (2015).

Janette Lehmann, Bruno Gonalves, Jos J. Ramasco, and Ciro Cattuto. 2012. Dynamical classes of collective attention in twitter. In *Proceedings of the 21st International Conference on World Wide Web*. ACM, 251–260.

Kristina Lerman and Rumi Ghosh. 2010. Information Contagion: An Empirical Study of the Spread of News on Digg and Twitter Social Networks. *Proceedings of the 4rd ACM International Conference on Web Search and Data Mining* 10 (2010), 90–97.

Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. 2010. Kronecker graphs: An approach to modeling networks. *The Journal of Machine Learning Research* 11 (2010), 985–1042.

Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2005. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. ACM, 177–187.

Jure Leskovec, Mary McGlohon, Christos Faloutsos, Natalie S. Glance, and Matthew Hurst. 2007. Patterns of Cascading behavior in large blog graphs.. In *SDM*, Vol. 7. SIAM Conference on Data Mining, 551–556.

Robert M. May and Alun L. Lloyd. 2001. Infection dynamics on scale-free networks. *Physical Review E* 64, 6 (2001), 066112.

Sen Pei, Lev Muchnik, Jos S. Andrade Jr, Zhiming Zheng, and Hernn A. Makse. 2012. Searching for super-spreaders of information in real-world social media. *Nature Scientific Reports* 4 (2012).

Henrique Pinto, Jussara M Almeida, and Marcos A Gonçalves. 2013. Using early view patterns to predict the popularity of youtube videos. In *Proceedings of the sixth ACM international conference on Web search and data mining*. ACM, 365–374.

Kazumi Saito, Ryohei Nakano, and Masahiro Kimura. 2008. Prediction of information diffusion probabilities for independent cascade model. In *Knowledge-Based Intelligent Information and Engineering Systems*. Springer, 67–75.

David A. Shamma, Jude Yew, Lyndon Kennedy, and Elizabeth F. Churchill. 2011. Viral Actions: Predicting Video View Counts Using Synchronous Sharing Behaviors.. In *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media*.

Eric Sun, Itamar Rosenn, Cameron Marlow, and Thomas M. Lento. 2009. Gesundheit! Modeling Contagion through Facebook News Feed.. In *Proceedings of the 3rd International AAAI Conference on Weblogs and Social Media*.

Gabor Szabo and Bernardo A. Huberman. 2010. Predicting the popularity of online content. *Commun. ACM* 53, 8 (2010), 80–88.

Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. 2009. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 807–816.

Senzhang Wang, Yan Zhao, Hu Xia, Yu Philip S., and Li Zhoujun. 2015. Burst time prediction in cascades. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*. ACM.

Lilian Weng, Filippo Menczer, and Yong-Yeol Ahn. 2013. Virality prediction and community structure in social networks. *Nature Scientific reports* 3 (2013).

Philip Wolfe. 1959. The secant method for simultaneous nonlinear equations. *Commun. ACM* 2, 12 (1959), 12–13.

Fang. Wu and Bernardo A. Huberman. 2007. Novelty and collective attention. *Proceedings of the National Academy of Sciences of the United States of America* 104, 45 (Nov 6 2007), 17599–17601.

Jaewon Yang and Jure Leskovec. 2011. Patterns of temporal variation in online media. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*. ACM, 177–186.

Bai Yu and Hong Fei. 2009. Modeling social cascade in the flickr social network. In *Fuzzy Systems and Knowledge Discovery, 2009. FSKD'09. Sixth International Conference on*, Vol. 7. IEEE, 566–570.

Honglin Yu, Lexing Xie, and Scott Sanner. 2014. Twitter-driven youtube views: Beyond individual influencers. In *Proceedings of the ACM International Conference on Multimedia*. ACM, 869–872.

Honglin Yu, Lexing Xie, and Scott Sanner. 2015. The Lifecyle of a Youtube Video: Phases, Content and Popularity. In *Ninth International AAAI Conference on Web and Social Media*.

Linyun Yu, Peng Cui, Fei Wang, Chaoming Song, and Shiqiang Yang. 2015. From micro to macro: Uncovering and predicting information cascading process with behavioral dynamics. In *Data mining (ICDM), 2015 IEEE international conference on*. IEEE, 559–568.

Xiaoyan Zhang and John A. Rice. 2003. Short-term travel time prediction. *Transportation Research Part C: Emerging Technologies* 11, 3 (2003), 187–210.

Qingyuan Zhao, Murat A Erdogdu, Hera Y He, Anand Rajaraman, and Jure Leskovec. 2015. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1513–1522.