Community-Aware Prediction of Virality Timing Using Big Data of Social Cascades

Alvin Junus, Ming Cheung, James She and Zhanming Jie
HKUST-NIE Social Media Lab, Hong Kong University of Science and Technology
{jalvin, cpming, eejames, zjieaa}@ust.hk

***Abstract*—Predicting the virality of contents is attractive for many applications in today's big data era. Previous works mostly focus on final popularity, but predicting the time at which content gets popular (virality timing), is essential for applications such as viral marketing. This work proposes a community-aware iterative algorithm to predict virality timing of contents in social media using big data of user dynamics in social cascades and community structure in social networks. From the continuously generated big data, the algorithm uses the increasing amount of data to make self-corrections on the virality timing prediction and improve its prediction. Experimental results on viral stories from a social network, Digg, prove that the proposed algorithm is able to predict viralty timing effectively, with the prediction error bounded within 30% with 20% of data.**

***Index Terms*—virality timing, virality prediction, community structure, social cascade, big data, social networks**

I. Introduction

Social media are now an integral part of our lives. People post and share content on social networks such as YouTube, Facebook and Twitter everyday. The content shared can be in the form of video, news, or images, among many others. While most online contents do not reach a lot of people, some can become viral and reach thousands, or even millions. One example is Gangnam Style, the popular Korean song which became a global hit, which is viewed more than 1 billion times in YouTube in a year. Fig. 1 (a) shows the popularity growth of the song. Another example is a story in Digg, a social network for sharing news or stories, which received more than 10000 votes in the first 140 minutes after it was published, as shown in Fig. 1 (b).

Virality of a piece of content can be taken as the targeted number of views, shares or votes for the content. In today's big data era, various information on online contents can be collected. Predicting content virality is attractive in many applications, e.g., in online marketing, predicting the number of audience reached gives a useful measure of the effectiveness of the marketing campaign. However, in such applications, the time at which the advertised content becomes popular is more important than the final popularity, i.e. knowing the virality timing - the time at which the content can reach the desired target - allows the campaign organiser to adjust the marketing duration. This translates to cost-effectiveness of the marketing campaign.

Most of the previous works focus on the final popularity of content, but do not consider its virality timing. The challenge here is to accurately predict the virality timing of online

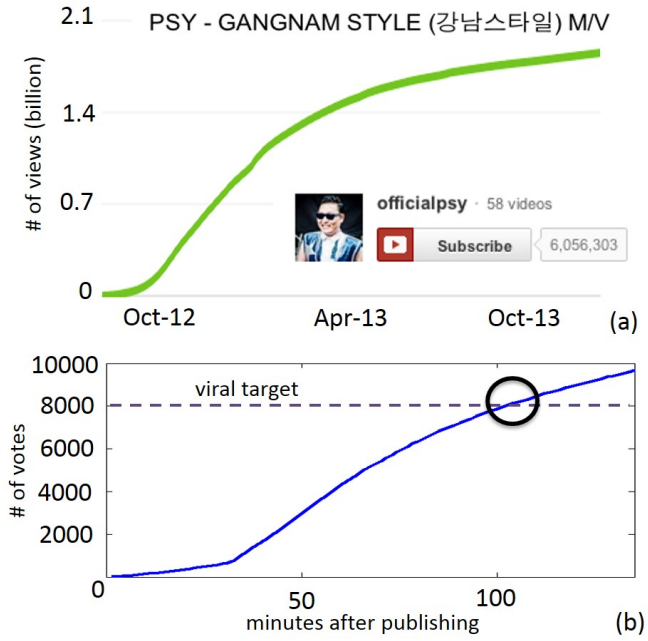

Fig. 1: Examples of viral content: (a) Gangnam Style, (b) popular story on Digg

contents in the early stages of their growths. This paper aims to solve the problem by using users' sharing dynamics and the community structure in social networks to model the population growth in time associated to a piece of content, and thus predict virality timing for the piece of content.

The rest of the paper is organised as follows. Related work in the topic is discussed in Section II. Section III presents the concept of social cascades, and how they, along with community structure, relate to virality timing. The proposed algorithm is detailed in Section IV, and is followed by experimental results that evaluate its performance in Section V. Finally, Section VI concludes the paper.

II. Related Work

Content properties and users' social interactions in the early stages of popularity growth have been proven to be highly correlated with the final content popularity [1], and are investigated for their effectiveness in predicting content virality [2]. Content features like author, tags, length and retweet count are proven to contribute to content virality [3] [4] [5]

[6]. These works present viable approaches to predict content virality, but they do not consider the time at which the content becomes viral (virality timing). A commonly found approach in the literature is time series analysis [7] [8], which proves that temporal patterns correlate with content popularity, but so far has not been employed in addressing the issue of virality timing. Recent work formulates virality prediction as a sequence of binary classification problems while a cascade is tracked over time and identifies features that contribute to content popularity [9]. While it presents some interesting findings, the prediction is still based on the final popularity, and does not focus on predicting the virality timing.

The main focus of most of the existing works is on the final content virality. However, it is also imperative to consider when such popularity can be reached. Our previous work incorporated cascade dynamics in social networks to predict virality timing [10]. This paper is an extension of the work and incorporates the underlying community structure of social network to predict virality timing. This is achieved through an iterative algorithm that considers community structure and social network dynamics obtained from the big data generated by user interactions, and self-corrects its prediction in each iteration. To the best of our knowledge, this is the first attempt to predict virality timing using both social network dynamics and community structure.

III. Predicting Virality Timing

In this section, social cascade and basic reproduction number are defined. Their relationship with virality timing is explained, followed by a description of how community structure can be used to measure virality timing.

A. Social Cascades

A social cascade is a process of information diffusion in a social network [11]. An example of a social cascade in Flickr is shown in Fig. 2, in which each node (i.e., a user) is sharing a common content (i.e., photo) to other nodes in a social graph. Node A, who first likes a photo P, is the initial node (seed) and is considered as generation 1 in a social cascade. For a social cascade to form, two users must have a social connection (e.g. friends or followers) in the social graph first, either bi-directional or uni-directional. In Fig. 5, nodes B and C also like photo P after node A does, and they already have social connections with node A before liking P. Node A is said to infect nodes B and C, and both nodes are considered as generation 2 in the social cascade. Similarly, node F is considered as generation 3 after being infected by node C. The concept of social cascades is not just applicable to one social network. Digg adopts a similar mechanism as in Flickr. The friends of a voter can see the story that the voter votes for and can vote for the story as well, thus forming a social cascade.

Exposure to top news from followed people or friends, as shown in Fig. 3, allows an ongoing cascade to grow. However, social network mechanisms such as recommended or recently popular news enable nodes which are not connected to the infected nodes to also be infected. In this case, these nodes

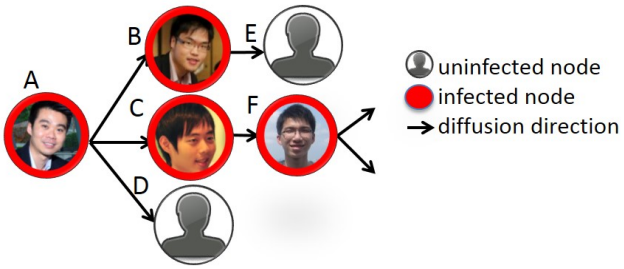

Fig. 2: Social cascade.

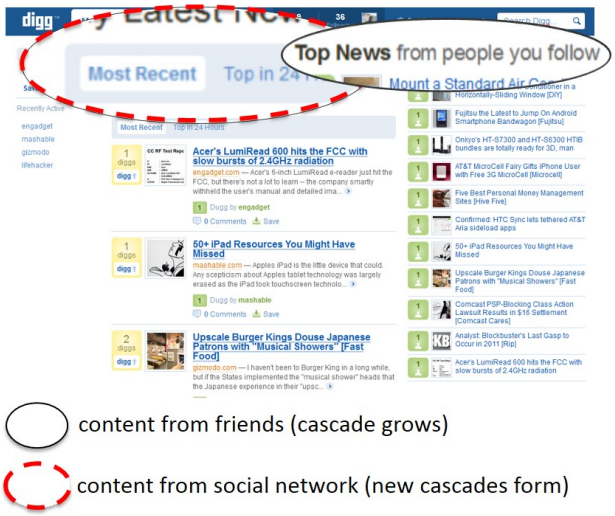

Fig. 3: Highlighted/recommended content from mechanisms that continue a cascade (solid line) and create multiple cascades (broken line).

will be considered to be seeds of new social cascades. Fig. 4 shows this phenomenon. Cascade size, i.e. the number of infected nodes in a cascade, is a measure of the content's virality. The proposed algorithm measures a piece of content's virality by summing the size of all of its social cascades. By measuring cascade size in time, a content's popularity growth can be known, and its virality timing can be predicted.

B. Basic Reproduction Number

The *basic reproduction number*, R_0, is defined as the expected number of secondary infections resulting from an infected node in a cascade. In epidemiological models, if $R_0 > 1$, one infected node will infect more than one node and the cascade size will grow. Examples are Fig. 5 (a) and the solid line in Fig. 5 (d). The cascade grows fast with the generation. $R_0 = 1$ is the critical case where the cascade grows linearly with the generation, as shown in Fig.5 (c) and the dotted line in Fig.5 (d). If $R_0 < 1$, the number of infected nodes will decrease for each subsequent generation and the cascade will fizzle out before it can infect many nodes. Examples are shown in Fig.5 (b) and the dashed line in Fig.5 (d). A viral piece of content will infect more nodes in one generation, resulting in a higher R_0.

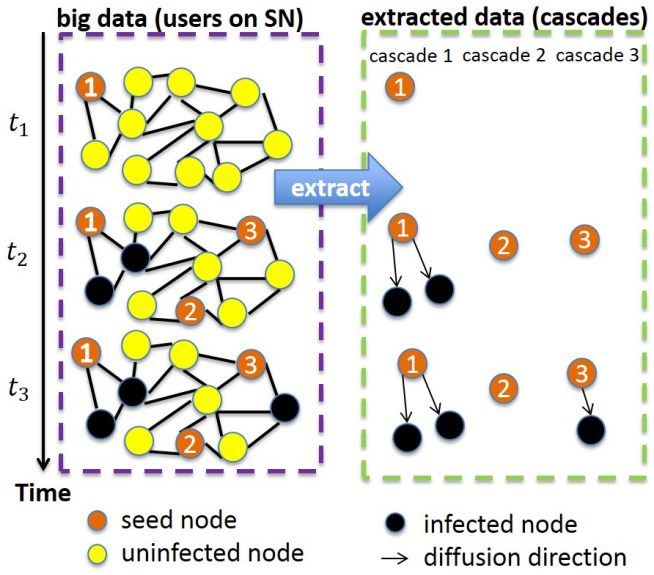

Fig. 4: Extracting social cascades from big data

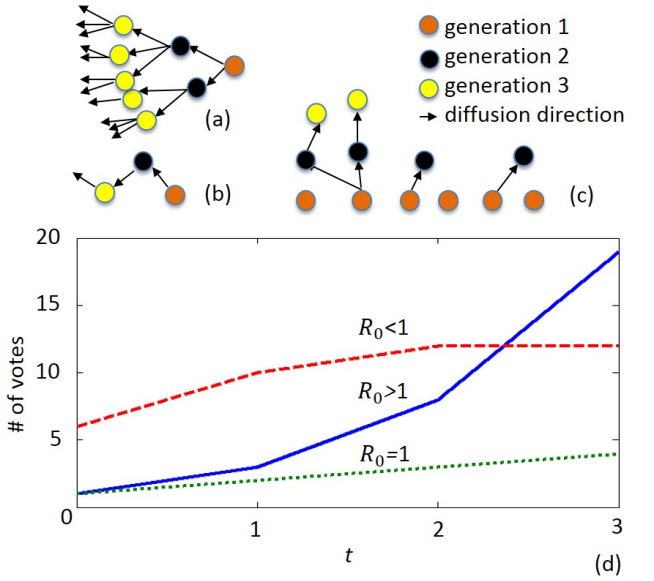

Fig. 5: Social cascades when (a) $R_0 > 1$, (b) $R_0 < 1$, (c) $R_0 = 1$, and (d) the prediction curves.

The theory of epidemiological models from [12] shows that R_0 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, 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. [11] states that the basic reproduction number R_0 can be obtained by counting the number of infected nodes directly from the seed. Eq. 1 is tested with more than 1000 shared pictures in Flickr over different social cascades, with an accurate result.

The growth of a cascade is affected by different factors: content, seed, resharer, cascade structure and temporal features [9], 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. By obtaining an estimate of the value of R_0, it is also possible to model the cascades' behaviors. Cascades' growth in time can be modelled and thus virality timing can be predicted. However, R_0 obtained at the early stages of the content's popularity growth may not be an accurate representation of the eventual popularity, as early infections may not accurately capture the actual infection dynamics of the content. This limitation motivates the iterative structure of the algorithm, where R_0 is updated in each successive iteration as more available data can better capture the actual dynamics. Thus, $R_0(t)$ is used to denote the basic reproduction number in an iteration at time t.

C. Community Structure and Content Virality

Similar users in a social network tend to connect with one another and form a community. Users belonging to the same community are more likely to be friends with one another, and also have many mutual friends [13]. Recent work shows that the more viral a piece of online content is, the more communities it will penetrate [14]. This observation presents qualitatively an additional angle to measure content virality, which can potentially improve the performance of the previous prediction algorithm. A viral content may infect nodes in different communities, and nodes in those communities will infect others in the same community or even nodes in other infected communities.

With the basic reproduction number, the qualitative evaluation of virality can be turned into quantitative, i.e., by assigning each community with its own R_0, the infection dynamics in each community can be captured. Cascades' growth in each community can then be modelled to obtain the virality timing prediction. Thus, $R_{0,c}(t)$ is used to denote the basic reproduction number of community c in an iteration at time t. Fig. 6 shows how a piece of content can spread within a single community, and also from one community to another in a simple social network through weak ties between communities. In this case, a social cascade may extend to different communities. The first node in a different community infected in this way is considered as the seed of a new cascade in the community. This is because each community is assigned its own R_0, and for R_0 to model the content's popularity growth in the community, all infections in the community, but not those outside, need to be considered.

With a community detection algorithm, nodes in a social network can be grouped into separate communities. Cascade growth in each community can then be modeled, and by summing up a content's popularity growth in all communities, its virality timing can be predicted.

IV. The Proposed Community-Aware Prediction Algorithm

In this work, users' sharing dynamics and community structure in social networks are used to predict virality timing, the time at which a certain content will reach a desired number of target

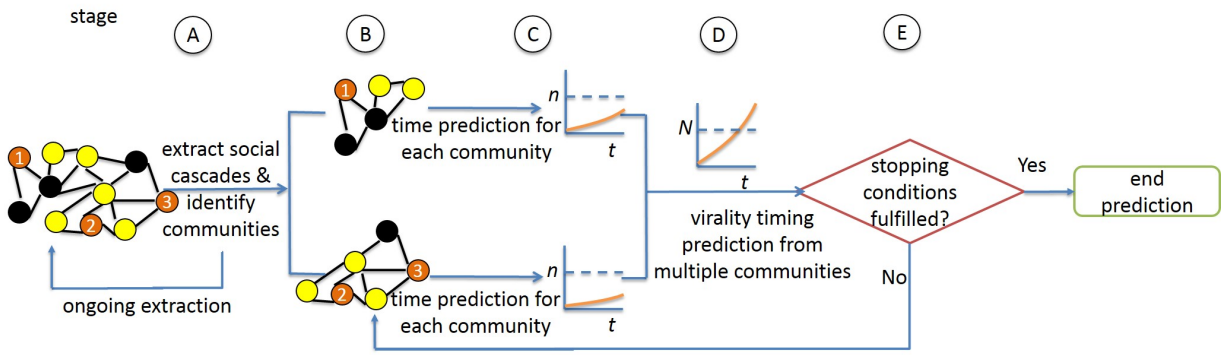

Fig. 7: Stages of the proposed algorithm.

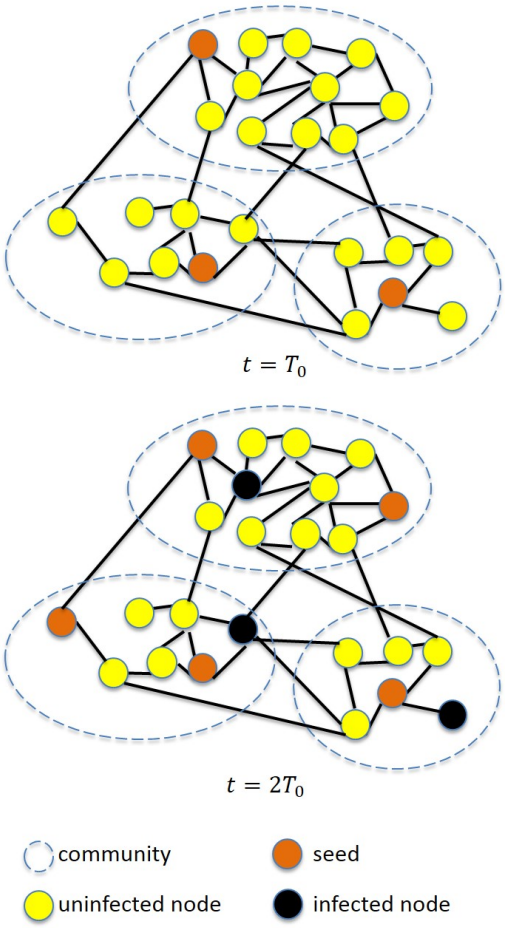

Fig. 6: Infections spreading to different communities

audience. A 5-stage iterative algorithm is designed to achieve this purpose. Fig. 7 shows each stage of the algorithm. Big data on user activities and community structure of the social network is continuously extracted automatically. Information on the community structure is continuously updated as new users are identified. The extracted data is used to update R_0 periodically. The R_0 of each community is then used to model how infection spreads in the community, and the prediction for each community is summed up to get the popularity growth in time of a single content in the social network. Consequently, virality timing prediciton for the content can be obtained. The algorithm then checks whether stopping conditions have been satisified, and if not, loops back to updating community dynamics.

A. *Extracting Information from Big Data*

Extraction of information from the big data of social network is an ongoing process. For a piece of content of interest, the algorithm continuously scans the social network and extracts information related to the content, such as the user reposting the content, the original poster, and its time. In this way, cascades formed in the social network can be extracted. When new users are extracted, they are placed into their respective communities accordingly. The Louvain algorithm is used here for large-scale community detection [16]. While community detection is essential, which algorithm to be selected is not the focus here.

B. *Updating Community Dynamics*

Unlike Stage A, this stage and all the subsequent stages are gone through periodically, with intervals T_0. As mentioned previously, R_0 can be obtained by counting the number of infected nodes directly from the seed. Fig. 6 shows that infections may penetrate through different communities through weak ties between communities. Nodes infected in this way are considered as seeds of new cascades for R_0 to model the content's popularity growth in the community accurately. A limitation of the previous algorithm is the assumption that there are an infinite number of nodes to be infected in the social network. Community detection addresses this limitation as the size of each community can be obtained in Stage A and a valid limit of each community's prediction value can be set.

C. *Virality Timing Prediction for Each Community*

In the third stage, community structure and R_0 of a community are used to obtain a content popularity growth prediction in the community. The predicted number of infected nodes in community c at time t for a future time t', $n'_c(t,t')$, can be modelled by the sum of a geometric series [15]:

$$n'_c(t,t') = n_c(t) + \sum_{j=1}^{k(t,t')} \Delta n_c(t) \cdot (R_{0,c}(t))^j, \qquad (2)$$

where $n_c(t)$ is the current number of infected nodes, and $\Delta n_c(t)$ is the number of newly infected nodes at time t in community c. The number of samplings that would have occured at future time t', $k(t,t')$, is given by:

$$k(t,t') = \lfloor \frac{t'-t}{T_0} \rfloor. \tag{3}$$

As the size of each community is known, the prediction value of any community c can be limited to the size of the community s_c.

$$n'_c(t,t') = \begin{cases} s_c & n'_c(t,t') \geq s_c \\ n'_c(t,t') & n'_c(t,t') < s_c \end{cases} \tag{4}$$

D. Predicting Virality Timing from Multiple Communities

The growth prediction for each community is summed up to obtain the growth prediction for the content at time t', $N'(t,t')$.

$$N'(t,t') = \sum_{c=1}^{L(t)} n'_c(t,t'), \tag{5}$$

where $L(t)$ is the number of communities identified as at the current time t. Hence, the predicted virality timing, $t(N)$, is the soonest time at which $N'(t,t')$ will reach or go beyond a given viral target N, which could be computed by solving the following:

$$t(N) = \arg\min_{t'} N'(t,t') \geq N \tag{6}$$

E. Stopping Conditions

The algorithm stops if at least one of two stopping conditions (C1 or C2) are fulfilled:

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

$$N(t) \geq N \tag{7}$$

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

$$t \geq T_{out} \tag{8}$$

In applications where time determines costs and expenses, e.g. viral marketing, the operation should be terminated to reduce incurred expenses if the viral target still has not been reached after a reasonable time T_{out} has passed. Parameters used in the algorithm are summarized in Table I.

TABLE I: Parameters in algorithm

Parameters	Definition
$R_0(t)$	basic reproduction # at time t
$N(t)$	# of infected nodes at time t for content
$L(t)$	# of communities identified at time t
$n_c(t)$	# of infected nodes at time t in community a
$\Delta n_c(t)$	# of newly infected nodes at time t in community a
t'	a future time for the prediction
$k(t,t')$	# of samplings up to future time t'
T_0	prediction interval
$n'_c(t,t')$	predicted $n_a(t)$ at time t for future time t'
s_c	size of community a
$N'(t,t')$	predicted $N(t)$ for future time t'
N	viral target
$t(N)$	minimum time needed to reach N
T_{out}	user-defined timeout period

V. Experimental Results

The performance of the algorithm is evaluated in this section. A benchmark error value E is defined as the percentage of deviation from the ground truth (i.e., the actual time to reach the viral target) [17]:

$$E = |t(N) - t_{GT}|/t_{GT}. \tag{9}$$

where $t(N)$ is the predicted virality timing and t_{GT} is the ground truth. A smaller E implies a more accurate virality timing prediction. Measurements of E over time show a trade-off between available data and prediction error. More data is available in a prediction at a later time and can reduce E, but the prediction's value may be less significant as it is closer to the ground truth. A good algorithm should give a prediction within a reasonable bound of error range at an earlier time such that the prediction will be useful for practical applications.

A. Real Dataset from Digg

Real data from a social network, Digg, is used to evaluate the proposed algorithm's performance. The dataset contains information on seed users, resharers, and the timings of each post or reshare. 3553 stories from 2006 and 2009 are covered, and 139410 users are involved. A smart local moving algorithm for the Louvain algorithm [18] is used to detect communities in the dataset based on users' friendship. 791 communities are formed in this way, with the largest community containing 35311 users.

Fig. 8 shows growth predictions of the previous algorithm in [10] and the proposed algorithm for the most viral story in the dataset, when there is 40% and 80% data available, respectively. The viral target N is set to be 15000. It can observed that E decreases as more data becomes available. E for the proposed community-aware algorithm is 10.4% less than that of the previous algorithm as shown in Fig. 8 (a), while the improvement in E decreases as more data becomes available, as shown in Fig. 8 (b).

In order to compare the general performance of the two algorithms, E of both algorithms for predictions of the 10% most viral stories are plotted in Fig. 9. Similar trends can be observed: E decreases as more data becomes available, and the performance improvement of the proposed community-aware algorithm decreases with more data. For the same viral stories, the proposed community-aware algorithm can give predictions with 31% lower E than the previous algorithm. Based on Fig. 9, it can be concluded that the prediction error of viral stories for the proposed community-aware algorithm can be bounded within 30% with 20% of data.

VI. Conclusion and Future Work

This work has extended [10] to predict the time at which a piece of content can reach a given viral target. An iterative

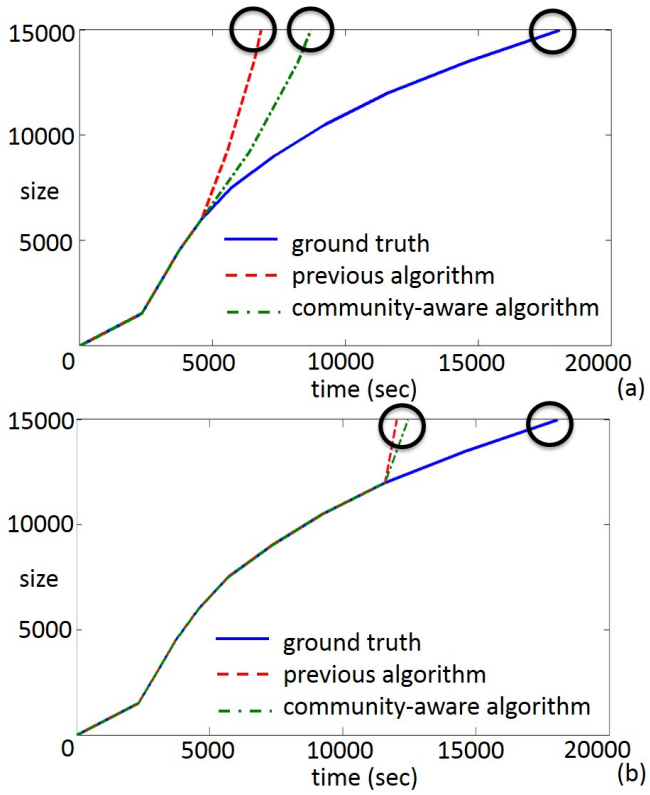

Fig. 8: Time and $N(t, t')$ of the most viral content in Digg when there is: a) 40% and b) 80% data available.

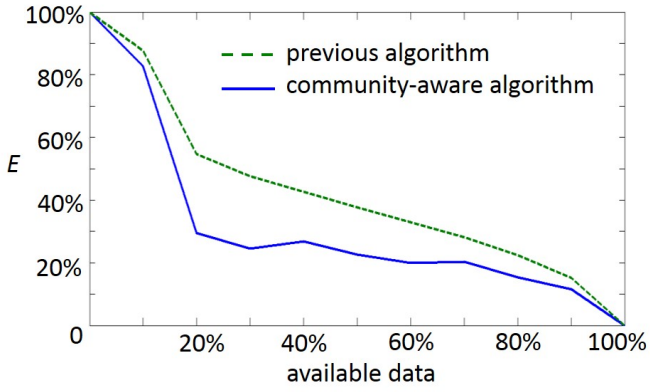

Fig. 9: The values of E for viral contents in Digg

and self-correcting algorithm is proposed to predict virality timing by taking advantage of an awareness of community structure in a social network from the big data of user sharings in the social network. Experimental results using real dataset from Digg prove that the algorithm works well in practical situations, in which the prediction error is reduced by 31% compared to the previous algorithm, and is bounded within 30% with 20% of data. More datasets from different social networks can be used in an extension of this work to obtain a more comprehensive evaluation of the algorithm.

A number of directions are possible for future continuation of the work. Characterization of weak and strong ties between nodes can model information diffusion in a social network better, and can characterize infections within and between communities, allowing a better prediction result. Different approaches on how a community is defined, e.g., based on user interests, conversation between users, etc, are also worth exploring, as different definitions will result in different communities being formed, thus affecting the prediction result.

ACKNOWLEDGMENT

This work is supported by the HKUST-NIE Social Media Lab, HKUST.

REFERENCES

[1] G. Szabo and B. Huberman, "Predicting the popularity of online content" Commun. ACM 53(8), August 2010, pp. 80-88.
[2] T. Wu, M. Timmers, D.D. Vleeschauwer and W.V. Leekwijck, "On the Use of Reservoir Computing in Popularity Prediction," Second International Conference on Evolving Internet, 2010, pp.19-24.
[3] C. Yun, "Performance evaluation of intelligent prediction models on the popularity of motion pictures," IEEE 4th International Conference on Interaction Sciences (ICIS), 2011, 16-18 Aug. 2011, pp.118-123.
[4] X. Cheng, C. Dale and J. Liu, "Understanding the characteristics of internet short video sharing: YouTube as a case study," CoRR, vol. 0707.3670, 2007.
[5] M. Jenders, G. Kasneci, and F. Naumann, "Analyzing and predicting viral tweets," in Proceedings of WWW Companion, 2013, pp. 657-664.
[6] S. Asur, B. A. Huberman, G. Szabo, and C. Wang, "Trends in social media: persistence and decay", in Proceedings of the 5th International AAAI Conference on Weblogs and Social Media, 2011.
[7] J. Yang and J. Leskovec, "Patterns of temporal variation in online media", in Proceedings of the 4th ACM International Conference on Web Search and Data Mining, 2011, pp. 177-186.
[8] J. Lehmann, B. Gonalves, J. J. Ramasco, and C. Cattuto, "Dynamical classes of collective attention in twitter", in Proceedings of the 21st International Conference on World Wide Web, 2012, pp. 251-260.
[9] J. Cheng, L. Adamic, P. A. Dow, J. M. Kleinberg and J. Leskovec, "Can cascades be predicted?" in Proceedings of the 23rd International Conference on World Wide Web, 2014, pp. 925-936.
[10] M. Cheung, J. She and L. Cao, "Predicting the content virality in social cascade", The IEEE International Conference on Cyber, Physical and Social Computing, Aug. 2013, pp. 970-975.
[11] M. Cha, A. Mislove, B. Adams and K. Gummadi, "Characterizing social cascades in flickr," in Proceedings of the first Workshop on Online Social Networks, 2008.
[12] R. M. May and A. L. Lloyd, "Infection dynamics on scale-free networks," Physical Review E64.6: 066112, 2001.
[13] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," in Proceedings of the National Academy of Sciences 99.12, 2002, pp. 7821-7826.
[14] L. Weng, F. Menczer and Y. Ahn, "Virality prediction and community structure in social networks," Nature Scientific Reports 3:2522, 2013.
[15] W. Galuba, K. Aberer, D. Chakraborty, Z. Despotovic and W. Kellerer, "Outtweeting the twitterers-predicting information cascades in microblogs," in Proceedings of the 3rd Conference on Online Social Networks, 2010, pp. 3-3.
[16] V.D. Blondel, J.-L Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks", Journal of Statistical Mechanics: Theory and Experiment, 10, 10008, 2008.
[17] X. Zhang and J. A. Rice, "Short-term travel time prediction," Transportation Research Part C: Emerging Technologies, vol. 11, 2003, pp. 187-210.
[18] Waltman, L., and Van Eck, N.J., "A smart local moving algorithm for large-scale modularity-based community detection," European Physical Journal B, 86(11), 471, 2013.