Predicting Content Virality in Social Cascade

Ming Cheung, James She, Lei Cao
HKUST-NIE Social Media Lab
Department of Electronic and Computer Engineering
Hong Kong University of Science and Technology, Hong Kong
{cpming, eejames, lcaoab}@ust.hk

***Abstract*—Predicting why and how certain content goes viral is attractive for many applications, such as viral marketing and social network applications, but is still a challenging task today. Existing prediction algorithms focus on predicting the content popularity without considering the timing. Those algorithms are based on information that may be uncommon or computationally expensive. This paper proposes a novel and practical algorithm to predict the virality of content. Instead of predicting the popularity, the algorithm predicts the time for the social cascade size to reach a given viral target. The algorithm is verified by the data from a popular social network - Digg.com and 2 synthesize datasets under different conditions. The results prove that the algorithm can achieve the lower bound with a practical significance for the time to reach the viral target.**

***Keywords*—prediction, popularity, social cascade, social network prediction, popularity, social network**

I. Introduction

Social media and social networks are part of our lives. Sharing text, video and other media content have become a daily activity for many people. Websites such as Digg.com enable people to share content on different social networks and make the content go viral. With an accurate prediction on how viral content can be is important for applications such as viral marketing and many social network applications. One of the possible reasons behind viral content is the viral spreading on social networks. Fig. 1 shows the number of votes for a popular story from Digg.com in the first 140 minutes after the publication. This piece of viral content was eventually voted for about 10000 times over the first 140 minutes. However, this type of success is not guaranteed and difficult to predict. Some content attracts the attention of millions, while most not. Predicting the virality, the tendency of content spread widely, is still challenging due to the nature of the content, and the influence among users on social networks.

The measurement of virality can be conducted by assessing the popularity, the total number of watches, shares or votes for. The virality prediction can be based on the early popularity [1][2][3], title [4][5], user interactions [6][7] or mathematical model [8]. While most of the algorithms focus on the final popularity as the virality, the prediction in this paper is the time for the number of infected nodes to reach a given viral target. The proposed algorithm focuses on the fast growing initial stage as shown in Fig. 1. It is based on the observation of the cascade size in terms of sharing/ voting/ viewing through a social network. People who shared/ voted /viewed content are defined as infected node on the network. The content spreads as a social cascade through the social network by the notification system. A larger cascade size implies a higher virality of content.

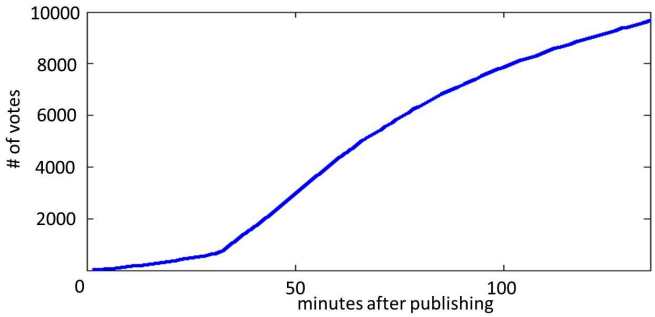


Fig. 1: Virality of a popular story in Digg.com [8]

It is proven that the basic reproduction number can be well modeled by the network structure and the cascade behavior [9]. The proposed algorithm, based on the basic reproduction number, is relatively simple. It is tested with three sets of data. The first one is a real dataset from Digg.com [8], a common social network for sharing. The second dataset is generated by the forest fire model. The third dataset is generated by the Kronecker graph. Social cascades are generated by the independent cascade model in the second and third datasets. The time for the number of votes to reach the viral target in each dataset is estimated and evaluated by comparison with the ground truth. It is proven that the algorithm can predict the virality by estimating the time required to reach the viral target with the basic reproduction number. The approach also provides a self-correction with new data to capture the dynamic of the cascade and the properties of users in the social network. It can be applied to real applications such as viral marketing and popular content detection for cloud computing.

The main contributions of this paper can be summarized as follows: 1) an approach to predict the time needed to reach a viral target; 2) self-correction to handle the dynamic of the cascade and user properties in the social network; and 3) experiments on 3 sets of data shows the approach works well on real data and synthesized data.

This paper is organized as follows: The related work is summarized in section II. The methodology of predicting the virality is presented in section III, followed by experiments and results in section IV. Section V concludes the paper.

II. Related Work

The prediction of virality can be based on: popularity in the early stage, the content and the social interaction. It is proven that the early popularity has a strong correlation with final virality [1] and is used in [2] and [3] to predict the final

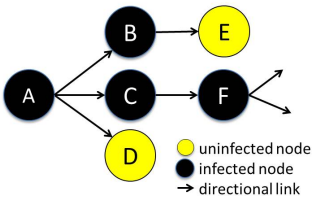


Fig. 2: Social cascade

virality. The algorithms based on popularity in the early stage are relatively simple, but they provide no insight into the content or the social interaction. In a content-based approach, the features of the content are used with standard classification techniques, such as support vector machine, to predict the virality. In [4], multiple factors such as authors and titles are used to classify the content into different virality classes. A similar algorithm is proposed in [5]. Attributes such as the release date and actors in a movie are used to predict its virality. While a content-based approach has enabled the classification of the virality, it also presents several problems. It can be technically complex and only applicable under certain assumptions. Another prediction approach is based on the social interactions. In [6] and [7], the user interactions and behaviors are captured to predict the virality. The behavior of users on Digg.com is modeled by [8] mathematically. The authors take into account the user interface and how it affects the user behavior. The model describes how the number of votes received by stories changes over time. However, the model may be computationally expensive and may not be always available.

Most of the algorithms focus on predicting the final popularity as the virality. However, those algorithms are either limited by assumption or computationally expensive. The proposed algorithm is based on the observation on the cascade size in the fast growing initial. The cascade size is predicted by the concept of the basic reproduction number that is relatively simple.

III. Predicting the Virality

In this section, the details of the prediction algorithm will be shown. First the social cascade is defined, followed by the basic reproduction number and the prediction method.

A. Social Cascade

A social cascade is the process of information diffusion in a social network [9]. For a social cascade in Flickr, user B likes a photo P through a social cascade in the social network if A, who also marked P as a favorite before B, was a contact of B before B marks P as a favorite. Two users must have a connection, bidirectional or unidirectional, on a social graph in order to have a cascade. Fig. 2 shows an example of a social cascade. Node A is the initial node of the cascade. Node A connects to 3 nodes: B, C and D. Node B and C are infected by node A. Node F is in their connected nodes and the cascade size, the total number of nodes in a cascade, will grow. The growth of a cascade is affected by the network structure and the properties of the nodes. The infection of an important node may result in a larger cascade size.

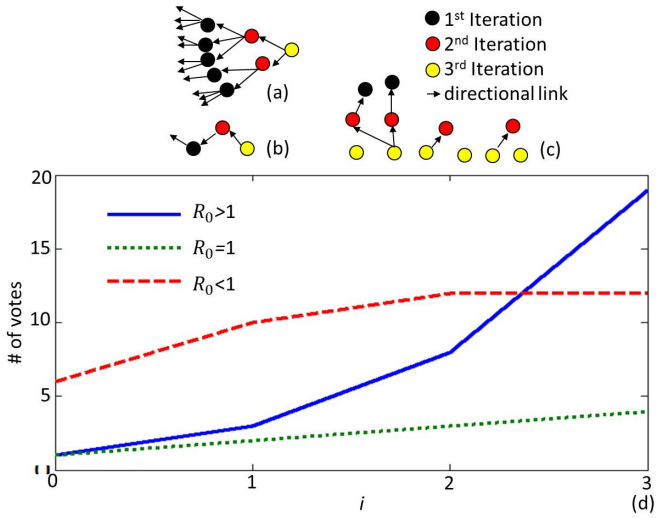


Fig. 3: Social cascades when (a) $R_0 > 1$. (b) $R_0 = 1$. (c) $R_0 < 1$. (d) the prediction curves.

The cascade size indicates the virality of content. A similar mechanism is used in Digg.com. The friends of a voter can see the story that the voter votes for and decide if they will vote for the story. The social cascade in Digg.com provides real data to investigate the accuracy of the algorithm. The proposed algorithm focuses on the size of the social cascade. The time needed to reach a viral target is predicted by the basic reproduction number (introduced in the next sub-section) and compared to the ground truth. A similar procedure is used in the cascade generation in the synthesized data.

B. Basic Reproduction Number

From [9], 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. An example is Fig. 3(a) and the solid line in Fig. 3(d). The cascade size grows fast with the generation. When R_0 = 1, it is the critical case where the cascade grows linearly with the generation, as shown in Fig. 3(b) and the dotted line in Fig. 3(d). If $R_0 < 1$, the number of infected for each generation will be decreasing and the cascade will fizzle out before many nodes are infected. An example is the shown in Fig. 3(c) and the broken line in Fig. 3(d).

Although the concept of R_0 is to model the growth of a cascade, it can be approximated if the number of uninfected nodes is much larger than the number of infected nodes [9]. The theory of epidemiological models from [10] shows that the basic reproduction number on a network is given by:

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

where $\rho_0 = \beta\gamma\overline{k}$. β and γ are the transmission rate and the duration of the infection respectively. k is the node degree, and $\overline{k}$ represents the mean value of the node degree. As stated in [9], the basic reproduction number R_0 can be obtained by

counting the number of infected nodes directly by the seed. Eq.(1) is tested with more than 1000 pictures from Flickr with an accurate result. By estimating the value of R_0, it is possible to predict the time required to reach the viral target.

However, by taking the R_0 from the number of infected nodes directly from the seed, the value may not be accurate. R_0 captures the general behavior of the cascade, but not the individual behavior. Nodes may have a different influence on the network, so that an infected important node may result in infecting a larger number of others. The prediction of a cascade may be affected by this dynamic and the error may be accumulated. Instead of using a fixed R_0, an iterative approach is used to estimate a more accurate R_0 with additional data. The R_0 of iteration i is represented by $R_{0,i}$.

C. The Proposed Prediction Algorithm

The focus of the paper is to predict the time needed for content to reach a viral target (i.e., the targeted cascade size). This is calculated by scraping data periodically to form iterations. The number of iterations required for reaching the viral target is estimated and the time is calculated correspondingly. The parameters are defined in Table I.

TABLE I: Definition of parameters

Parameters	Definition
n	viral target (targeted cascade size)
i	iteration number
n_i	number of nodes currently infected
Δn_i	number of newly infected nodes
t_i	time duration for each iteration i (generally assumed $t_i = t_{t+1} = t = a$ constant duration)
$R_{0,i}$	basic reproduction number at iteration i
$i(n)$	predicted iterations needed to reach n
i_{GT}	number of iterations to viral target (ground truth)
$t_{i(n)}$	minimum time needed to reach n
E_p	percentage error in prediction

For iteration i, Δn_i is obtained from the scraped data, and $R_{0,i}$ is calculated. With $R_{0,i}$ known, n', the total number of infected nodes after $i(n')$, can be calculated by:

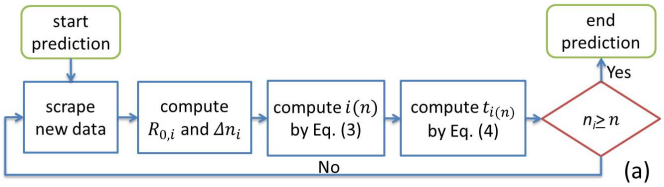
$$n' = n_i + \sum_{j=1}^{i(n')-i} (\Delta n_i \cdot R_{0,i}^j) \quad (2)$$

By setting $n' = n$, the sum of a geometric series and changing the subject to $i(n)$, (2) becomes (details can be found in appendix):

$$i(n) = i - 1 + log_{R_{0,1}}(\frac{\Delta n_i \cdot R_{0,i} + (n - n_i)(R_{0,i} - 1)}{\Delta n_i}) \quad (3)$$

The time needed to reach the viral target can be estimated by:

$$t_{i(n)} = i(n) \cdot t \quad (4)$$



Algorithm

for $i(n) < n$ **do**
  scape new data
  compute $R_{0,i}$ and Δn_i
  compute $i(n)$ by Eq. (3)
  compute $t_{i(n)}$ by Eq. (4)
end for

(b)

Fig. 4: (a) Flowchart and (b) Algorithm

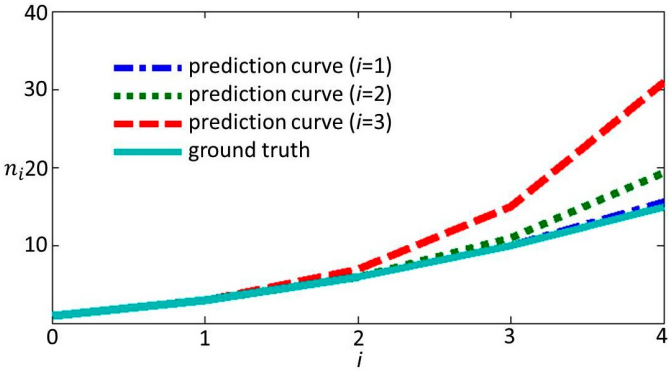


Fig. 5: Example of prediction curves

Fig. 4 summarizes the steps of the algorithm. The first step of the proposed algorithm is scraping new data, followed by calculating Δn_i and $R_{0,i}$. $i(n)$ and $t_{i(n)}$ are estimated in the next step. Again, prediction based on the current data may not capture the dynamic of the cascade and the properties of the social network. The prediction of $t_{i(n)}$ may be large as the error is accumulated. The prediction is updated in each iteration to capture the dynamic of the cascade and network. If the number of currently infected nodes is greater than the viral target, the operation will be stopped. Otherwise, it loops back to the data scraping. Fig. 5 shows the prediction curves and the ground truth of an example. n is set to 15 on a cascade initialized with 1 seed. The cascade reaches n at iteration 4, with n_i equal to 3, 6, 10 and 15 in all iterations. The corresponding $R_{0,i}$ is 2, 1.5, 1.5 and 1.4. The prediction curves are calculated accordingly and are shown in Fig. 5 accordingly. The prediction stops when the current number of infected nodes is greater than the viral target when i = 4.

In this paper, 3 datasets are used to verify the proposed algorithm. The experiment and the results will be presented in the next section.

IV. Experimental Results

Real data from Digg.com and synthesized data are used to investigate the algorithm. The data from Digg.com provides a good ground for testing. In order to test the algorithm in

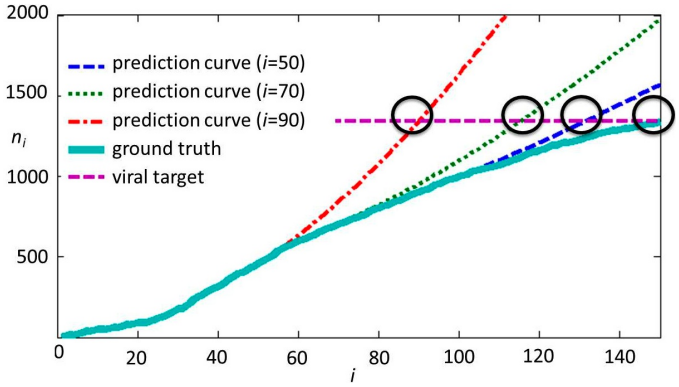


Fig. 6: i and n_i of infected node (1st story)

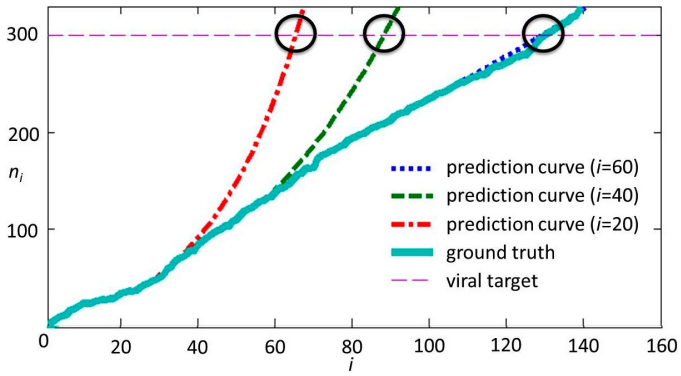


Fig. 7: i and n_i of infected node i (2nd story)

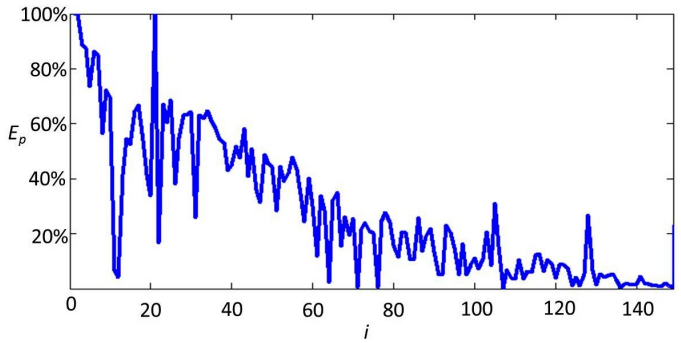


Fig. 8: E_p in different iterations (1st story)

controlled conditions, data generated by the forest fire model and the Kronecker graph are also used. The social cascades are generated with different diffusion rates on the generated social graph by the independent cascade model (ICM). With ICM, an infected node infects its neighbor with a given probability. Those infected neighbors will continue to infect their neighbors, and hence a cascade is formed.

The prediction is based on the size of a cascade. From one node, the size of cascade grows through the social network. E_p is used to evaluate the performance of the algorithm. The evaluation is based on the difference between the ground truth, and is defined in the following:

$$E_p = |i(n) - i_{GT}|/i_{GT} \tag{5}$$

This is defined as the absolute difference between the ground truth and prediction divided by the ground truth. It provides an evaluation of the prediction performance. A smaller E_p implies a better performance.

A. *Digg.com*

The dataset was scraped from web pages in Digg.com by the authors of [8]. The data includes stories from 2 years: 2006 and 2009, in which there were 1251 stories and 89643 users. Digg.com is a website where users can submit stories. It allows users to track their friends' activities, such as the submissions and the voting. A newly submitted story can be voted on, and becomes visible to the voter's friends so that they can also vote. Similar to the definition of a social cascade in [9], the user must be a follower of a voter before the user can vote for the same story. Votes that are not from a social cascade are discarded. This makes sure that the cascade is the reason for the voting and not other mechanisms of Digg.com. In particular, two popular stories are selected for the testing. The ground truth is based on the observation of the cascade size through the social network. The virality of the stories makes it as a good sample for the experiment in which the first reach 1300 in less than 4 hours. The results of the 2 stories are in Fig. 6 and Fig. 7. The estimated R_0 can provide insight into how the cascade grows. $i(n)$ is a good estimation of the true value. For 1st story in Fig. 6, E_p for iterations 50, 70 and 90 is 44.6%, 25.4% and 13.2% respectively. The prediction at iterations before 60 underestimate the growth of the viral size as the cascade is at slow growth phase. When $i > 60$, the cascade enters the explosive phase, and E_p is greatly reduced. For 2nd story in Fig. 7, the prediction at iterations 30 and 60 are not in accordance with the early observed data but 90. The prediction at iteration 90 fits the curve well with a small E_p. The results of the 2 stories show a similar trend: when more data is available, the prediction is more accurate. From Fig. 6, E_p in the early prediction is large and E_p is reduced when more data is available. For the prediction in Fig. 7, the same observation can be found. Prediction at iteration 90 is a better than the predictions before it. The meaning is that when more data is available, the prediction is more accurate as $R_{0,i}$ captures the dynamic of the social network better. However, the usefulness of the prediction is reduced, as the prediction is closer to the current iteration.

In order to obtain the general performance of the algorithm, the prediction of each iteration is evaluated by E_p. The result is a better evaluation of the algorithm. It is summarized in Fig. 8. It is observed that E_p decreases with the iteration. More available data improves the result on the prediction. However, at some iterations, E_p may be increased. At i = 14, there are no newly infected nodes, hence no prediction can be made. E_p is set to 1 in this case. E_p is small at i = 13 and 15. It could be solved by choosing another duration of iteration.

Synthesized data will be used in the next section to simulate social cascades in different conditions. The infection rate of the cascades is controlled and a deeper study will be conducted. This provides the general behavior of the cascade.

B. *Forest fire and ICM*

The forest fire proposed in [12] is used to generate a social graph for the cascade. The graph fulfills properties such

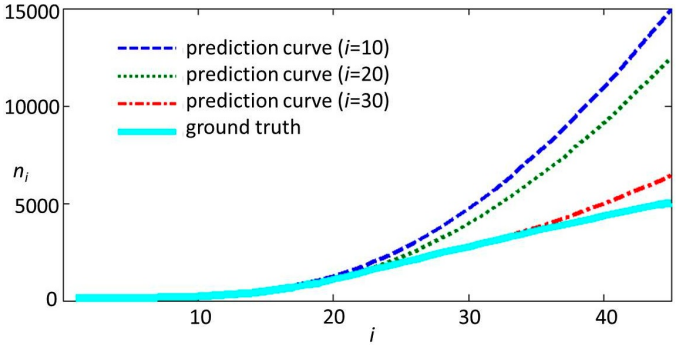


Fig. 9: i and n_i in the forest fire model

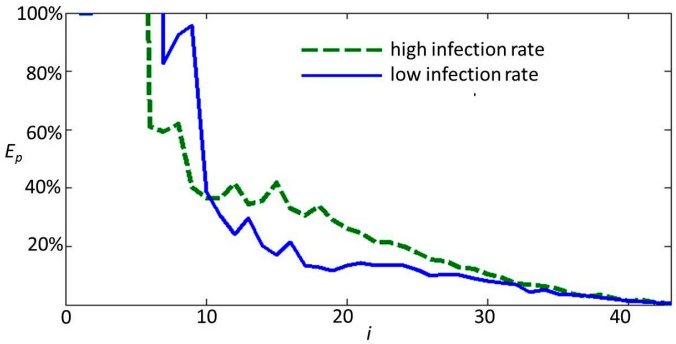


Fig. 10: E_p in the forest fire model

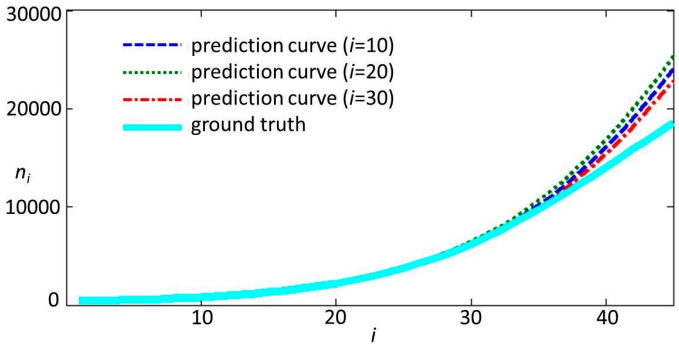


Fig. 11: i and n_i in the Kronecker graphs

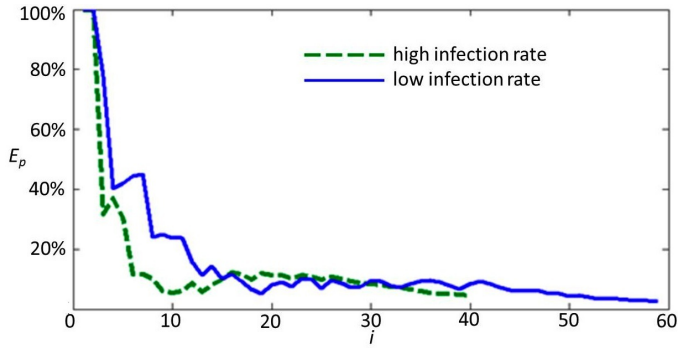


Fig. 12: E_p in the Kronecker graphs

as densification laws, shrinking diameters and others social graph properties. It provides a good reference for testing the prediction algorithm. 50000 nodes with 268657 edges are generated. A social cascade is created on the social graph with a probability of infection by ICM. The algorithm is also tested on an iterative approach. Fig. 9 shows the prediction curve and the ground truth of the size of a generated cascade. The prediction is on iteration 10, 20 and 30. It is observed that the accuracy of the prediction improves with each iteration, that is, the amount of data available.

In order to obtain a better evaluation on the performance on the algorithm, 20 trials are conducted for high and low growth rates with 400 random nodes as the seeds. $t_{i(n)}$ is calculated and compared with the true value in each iteration and the average E_p is obtained. The result is summarized in Fig. 10. A similar conclusion to that in Fig. 9 can be drawn. The later iteration has a smaller E_p as more data is available. There is a similar effect on the high and low infection rate: E_p drops dramatically after the first few iterations.

C. Kronecker Graphs and ICM

The Kronecker graph [11] is a social graph generator to produce graphs exhibiting the full range of properties observed in prior works [12]. Similar to the forest fire model, the graph also fulfills social graph properties. In the experiment, a social graph with 59049 nodes with 9765625 edges is generated. The cascades are generated by a similar setting to the one in the forest fire model. The algorithm is tested on an iterative approach.

Fig. 11 shows the prediction curve and the ground truth of the size of a generated cascade. The prediction is on iterations 10, 20 and 30. It is observed that the predictions match the ground truth well.

Twenty trials are conducted for high and low growth rates with 400 random nodes as the seeds to obtain a better evaluation of the algorithm. $t_{i(n)}$ is calculated and compared with the true value in each iteration and the average E_p is obtained. The result is summarized in Fig. 12. A conclusion similar to the prediction in the forest fire model can be drawn. The later iteration has a smaller E_p as more data is available. There is a similar effect on the high and low infection rate: E_p drops dramatically after the first few iterations. The percentage error of the prediction is below 20% when there is 20% of the data. Based on the result, it is possible to predict the time for the first 20% of the data with less than 20% error.

The same trend is observed from the experiments on the 3 datasets. From Fig. 6 and Fig. 7, E_p is reduced for the predictions in later iterations. This can be observed from the E_p of Fig. 8. In Fig. 10 and Fig. 12, E_p is also reduced with time. The prediction based on a larger amount of data is more accurate. However, the prediction is less useful as the current iteration is closer to the time to reach the viral target.

D. Discussions

Although the algorithm has demonstrated a promising result, further investigations are required: 1) the trade off between the accuracy and the time, and 2) the duration of iteration. As shown in the results, a larger data set improves the accuracy but the time value is lost as it is closer to the time to reach the viral target. An early prediction requires less data, but may have a larger error. A study for the optimum point in terms of the accuracy and the time is needed.

The duration of iteration also affect the accuracy. On the one hand, a shorter duration may underestimate R_0 which the predicted time is longer than the actual time. On the other hand, the viral target is reached in a small number of iterations with a long duration. The error may be large in this case. The selection of the iteration duration is another subject for the investigation.

V. Conclusion

This paper has introduced a novel approach to predict the content virality by estimating the time required for the viral target with the basic reproduction number, which provides insight into cascades on social networks. The prediction is a self-correction approach. The algorithm is tested with 3 sets of data: real data from Digg.com and cascades generated by the Kronecker graph and the forest fire model. The tests show that the algorithm can be a lower bound and a good prediction of the virality for cascades, giving a time prediction for the size of the cascade to reach the viral target. The test on the real data shows how the algorithm works, and the test on the synthesized data shows how the algorithm works in extreme conditions. According to the tests, it is possible to predict the time from the first 20% of data with less than 20% error. Based on the incoming data, the prediction accuracy can be improved to handle the dynamic of social network and cascade. The algorithm has been proven to be a good estimation with a practical significance for the time to reach the viral target.

Acknowledgment

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

References

[1] M. Cha, H. Kwak, P. Rodriguez, Y-Y Ahn and S. Moon, "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 (IMC '07). ACM, 2007.

[2] G. Szabo and B. Huberman, "Predicting the popularity of online content" Commun. ACM 53, 8 (August 2010)

[3] T. Wu, M. Timmers, D.D. Vleeschauwer and W.V. Leekwijck, "On the Use of Reservoir Computing in Popularity Prediction," Evolving Internet (INTERNET), 2010 Second International Conference on , vol., no., pp.19-24, 20-25 Sept. 2010

[4] S. Kim, S. Kim and H. Cho, "Predicting the Virtual Temperature of Web-Blog Articles as a Measurement Tool for Online Popularity," Computer and Information Technology (CIT), 2011 IEEE 11th International Conference on , vol., no., pp.449-454, Aug. 31 2011-Sept. 2 2011

[5] C. Yun, "Performance evaluation of intelligent prediction models on the popularity of motion pictures," Interaction Sciences (ICIS), 2011 4th International Conference on , vol., no., pp.118-123, 16-18 Aug. 2011

[6] J. Lee; S. Moon and K. Salamatian, "An Approach to Model and Predict the Popularity of Online Contents with Explanatory Factors," Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on, vol.1, no., pp.623-630, Aug. 31 2010-Sept. 3 2010

[7] D. Shamma, J. Yew, L. Kennedy and E. Churchill, "Viral Actions: Predicting Video View Counts Using Synchronous Sharing Behaviors" International AAAI Conference on Weblogs and Social Media 2011.

[8] K. Lerman and R. Ghosh, "Information Contagion: an Empirical Study of Spread of News on Digg and Twitter Social Networks." In Proceedings of 4th International Conference on Weblogs and Social Media (ICWSM).

[9] M. Cha, A. Mislove, B. Adams and K. Gummadi, "Characterizing social cascades in flickr" In Proceedings of the first workshop on Online social networks (WOSN '08).

[10] R. M. May and A. L. Lloyd, "Infection Dynamics on Scale-Free Networks." Physics Review E, 2001.

[11] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos and Z. Ghahramani, "Kronecker Graphs: An Approach to Modeling Networks" J. Mach. Learn. Res. 11 (March 2010)

[12] J. Leskovec, J. Kleinberg and C. Faloutsos, "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 (KDD '05)

Appendix

From Eq.(2):

$$n' = n_i + \sum_{j=1}^{i(n')-i} (\Delta n_i \cdot R_{0,i}^j) \tag{6}$$

By using the formula of the sum of geometric series and set $n' = n$:

$$n' = n_i + \Delta n_i \cdot R_{0,i} \frac{(1 - R_{0,i})^{i(n)-i}}{1 - R_{0,i}} \tag{7}$$

By changing the subject to $R_{0,i}$:

$$R_{0,i}^{i(n)-i+1} = \frac{\Delta n_i \cdot R_{0,i} + (n - n_i)(R_{0,1} - 1)}{\Delta n_i} \tag{8}$$

By taking log with base $R_{0,i}$ on both sides:

$$i(n) - i + 1 = \log_{R_{0,i}}(\frac{\Delta n_i \cdot R_{0,i} + (n - n_i)(R_{0,i} - 1)}{\Delta n_i}) \tag{9}$$

By changing the subject to $i(n)$, Eq. (8) becomes:

$$i(n) = i - 1 + \log_{R_{0,i}}(\frac{\Delta n_i \cdot R_{0,i} + (n - n_i)(R_{0,i} - 1)}{\Delta n_i}) \tag{10}$$

which is Eq.(3).