An Influence Maximization Algorithm Based on Real-Time and De-superimposed Diffusibility

Yue Ren[1], Xinyuan Zhang[1], Liting Xia[1], Yongze Lin[1,2], Yue Zhao[1], and Weimin Li[1(✉)]

[1] School of Computer Engineering and Technology, Shanghai University, Shanghai, China
randomvar788@gmail.com, zxy_zhangxinyuan@163.com, xia_lt@163.com, yongze_lin@163.com, {yxzhao,wmli}@shu.edu.cn

[2] Shanghai Key Laboratory of Computer, Software Evaluating and Testing, Shanghai, China

Abstract. Influence maximization is to find a small number of seed nodes in the network that maximize their influence on the network. Existing algorithms select a seed node with the greatest influence. This will inevitably have an influence on mutual coverage, which will have a more or less negative impact on the final results and reduce the performance of the algorithm. In this paper, Node Diffusibility is proposed, and it is updated in real time and eliminated the deviation caused by its overlay. On the basis of traditional calculation of node influence, more attention was paid to the influence of a node's neighboring nodes rather than to the characteristics of the nodes themselves. The proposed algorithm was evaluated by experiments conducted on selected real data sets. Compared with the classical ranking-based algorithms, MaxDegree and PageRank, the proposed algorithm achieved better results in terms of efficiency and time complexity.

Keywords: Social network · Influence maximization · Diffusibility

1 Introduction

With the continuous development of network technology, social networks have become more and more widely used in real life, which has changed the way people communicate or share information. A social network is a complex network that consists of many individuals and their connections. When a person gets a product or a message, he could recommend it to others. Some of them would accept the message and spread to more people nearby under the effect of "Word of Mouth". In this way, the message will be spread from several individuals to some groups. The social information platform is booming and its market value is increasing. For example, it has a strong practical significance in virus marketing [1, 2] and public opinion control. How to spend the least cost to get the most extensive dissemination range, namely to obtain the maximum influence has become the most important thing for information publishers. This is also

X. Wang et al. (Eds.): CollaborateCom 2019, LNICST 292, pp. 537–548, 2019.
https://doi.org/10.1007/978-3-030-30146-0_37

the most critical part of the information dissemination process. Based on the social communication model, this paper will simulate the process of information diffusion in social networks and discuss the influence maximization [3, 4].

The spread and diffusion of social networks have a long history of social science. In recent years, many scholars have conducted deep research on these topics. Social networks have become a research hot spot at present, mainly including the information dissemination modeling in the social network [5], community detection, the calculation of user influence, and the study of the influence maximization. Richardson et al. [6] introduced the issue of influence maximization and defined it specifically in social networks. Kempe et al. [7] studied this problem in detail and abstracted it into a discrete optimization problem to simulate the information transmission process. Many algorithms, such as Greedy algorithm [8, 9] and Heuristic algorithm [10, 11], are been used to solve the problem of influence maximization in social networks. The algorithm of influence maximization is to select some high-impact seed nodes through some appropriate methods and maximize the influence by spreading the messages from these seed nodes.

In the study of the influence maximization algorithm for social networks, Kemple and Kleinberg proposed Greedy Algorithm, which selects the node that can bring the maximum influence benefits each time. However, there exists a problem during the process of influence maximization. The selected seed nodes with the largest influence inevitably have the influence of mutual coverage. In order to address these issues node diffusibility is defined. Then, based on the traditional calculation of node influence we paid more attention to the influence of a node's neighboring and updated the diffusion in real time instead of just to the characteristics of the node itself. Finally, an algorithm is proposed to maximize the influence of real-time diffusibility based on the Linear Threshold Model [12].

The rest of this paper is organized as follows. Section 2 introduces the related work. Section 3 presents a new concept, the algorithm framework, and two optimization methods. Section 4 presents the experimental results. Finally, Sect. 5 concludes this study by highlighting our main contribution and future research work.

2 Related Work

Kempe et al. [7] first established the model of influence maximization, which aims at finding the most influential K nodes on a specific dissemination model. They pro-posed Greedy Algorithm, and simulated the information dissemination process of K rounds in the whole network diagram. The marginal influence of nodes was calculated for selecting seed nodes in each round, and the most influential node could be gained. However, this process is very time-consuming, and the local optimum cannot reach the optimal result of the final dissemination.

Set Covering Greedy Algorithm [13] is another Influence Maximization Algorithm. Once a node is selected as a seed node, all its neighboring nodes will be marked as covered. The algorithm chooses the uncovered node with the highest degree each time, that is, the node with the largest coverage. However, the coverage mentioned here is not equal to activation, so the experimental results are not good for influence maximization.

Heuristic algorithm based on node centrality is a method to reduce the complexity. To evaluate the centrality of nodes in the network [14], many algorithms are proposed, such as Degree Centrality [15, 16], Closeness Centrality [17], Betweenness Centrality [18] and PageRank [19, 20]. Degree Centrality is the most common and simplest measurement. The greater the Degree Centrality of the node, the more important the node is in the network [15]. Closeness Centrality is another index to measure the centrality of nodes by calculating the path length of each node to other nodes. If the path length of one node to other nodes is small, the influence of this node may be greater, also the information diffused by this node will disseminate more easily [17]. Betweenness Centrality is related to the shortest paths of two nodes in the network. If plenty of these shortest paths pass through one node, this node is considered to have high Betweenness Centrality [18]. However, PageRank is different from each of the three ways mentioned above [19]. It's used to evaluate the influence of Web pages. It can also be understood as a method to measure the importance of nodes. Kitsak et al. [21] proposed k-core algorithm to evaluate the dissemination influence of nodes, and proposed Maximum Core Algorithm Based on Coverage and Maximum Degree Algorithm. K- core decomposition measures the centrality of one node by its location in the network. If the centrality of a node is large, it can be gained that this node is in the core position of the network, and its influence may be greater. Cao Qiuxin et al. [22] proposed Core Covering Algorithm, which combined the k-core algorithm and degree centrality to calculate the influence of each node. However, these traditional node centrality index always ignore the characteristics of its neighboring nodes. Degree Discount [3] is an optimization of it. Chen et al. pointed out that when some of one node's neighboring nodes are seed nodes, the degree of the node should be discounted to avoid the overlap of influence.

In addition, there exists some Influence Maximization Algorithm Based on Community Discovery [23]. In social networks, people will form many communities be-cause of various interests and hobbies. Social networks can be divided into many small aggregation areas according to certain characteristics through some behaviors of people. Heuristic algorithm is not always so effective. In addition, the greedy algorithm adds seed node every time and calculates the marginal impact of all inactive nodes, which makes the algorithm run for a long time. Therefore, a better evaluation of node influence needs to be studied. In this paper, we pay attention to the influence of a node's neighboring nodes and eliminates the superimposed influence between neighboring nodes. The aim of this study is to achieve better results with the less computational time cost.

3 Influence Maximization Algorithm Based on Real-Time and De-superimposed Diffusibility

In this section, node diffusibility, a new metric that measures the importance of the node, is defined. It takes the overall impact of the node and its neighbors into account. Loss coefficient is added to simulate the loss caused by information diffusion. Based on experiments, it is found that the nodes with larger diffusibility have higher diffusion range coincidence. Also, node diffusibility is updated in real time to get the most realistic dissemination process. Besides, eliminating the negative effect caused by diffusibility superposition is also considered to get more precise conclusions.

3.1 Node Diffusibility

In an information dissemination network, each node has a different situation and status. Thus, these nodes play different roles in information dissemination. Therefore, it is of great significance to judge the status and importance of a node in the network. Traditional node importance metrics, such as Degree Centrality and Node Influence, usually focus on some factors directly related to one node while ignoring the features of the relevant nodes connected with it. Generally, there exists a fact that the degree of a certain node is very large, but the degree of its neighboring nodes is relatively small. Thus, its influence cannot reach a high level.

To solve the above problems, a new metric Node Diffusibility was proposed. Based on the traditional nodes influence, we considered the influence of the neighboring nodes of one node. The influence of the node spreading i layers is positively correlated with the influence of its neighboring node spreading $i-1$ layers ($i > 1$). According to the simulation experiments, the necessary condition for one node to be activated is that the gained influence from the surrounding activated nodes reaches its own threshold. So the diffusion of information over each layer must be accompanied by a certain loss. To simulate the loss caused by dissemination, a variable called Loss Coefficient was defined to quantify it.

The estimation formula of node diffusibility is as follows:

$$db(u,i) = db(u,i-1) + \sum\nolimits_{v \in U'} db(v,i-1) \times ls \times b(u,v) \tag{1}$$

$$db(u,1) = \sum\nolimits_{v \in U'} b(u,v) \tag{2}$$

where *db(u, i)* represents the diffusibility of node u spreading i layers; U' is the neighboring node of u; the ls represents the loss coefficient; *b(u, v)* represents the effect of the active node u on the neighboring node v, calculated by *1/(d(v))*; *d(v)* represents the degree (in-degree in the directed graph) of the node v.

Algorithm 1. Node Diffusibility Calculating (NDC)

Input: graph: G(V, E),Θ, size of initial dissemination set: k, loss coefficient: ls, n: consider node spreading n layers
Output: initial dissemination set: s

1. Set s_0=Ø
2. **For each** node u in graph G do
   $b_{uv} \leftarrow 1/indegree\ of\ u$ /*calculate the effect of the active node u on the neighboring node v*/
3. **For each** node u in graph G **do**
   For each node v in neighbors of node u **do**
   For each i in n: /* consider node u spreading n layers */
   $db(u,i) \leftarrow db(u,i-1) + db(v,i-1) \times ls * b_{uv}$ /**db(u,i)* represents the diffusibility of node u spreading i layers */
   End For
   End For
   End For
4. Sort nodes in G by its value of *db(u,n)*
5. Select *top-k* nodes with large diffusibility to join the set s_0

Algorithm 1 is node diffusibility calculating algorithm (NDC). The effect of node can be calculated, and the loss coefficient is defined to simulate the loss of effect caused by dissemination. The algorithm selects top k nodes that have larger diffusibility. Though we need to consider several layers of dissemination, the time complexity of NDC is *O(E)* by memorization. The space complexity is mainly consumed on the storage of the network, which is *O(E)*. *E* is the number of edges.

3.2 Influence Maximization Based on Real-Time and De-superimposed Diffusibility

In order to apply the diffusibility in the dissemination model, some verified experiments were carried out. By combining with the Linear Threshold Model, the top-k nodes of the diffusibility were selected as the seed nodes, and the number of nodes that can be activated could be gained. Through experiments, it is found that the diffusion range of the nodes with large diffusibility is high. Thus, the following two improvements were propose:

(1) Update the node diffusibility in real time

As known that each seed node has an influence on its neighboring nodes. And several such influences will contribute a part of the total influence of the seed node. Considering the following situation, one node v can reach node u after being diffused through n layers. After the node u is selected as a seed node, it is obvious that the node u does not have the ability to provide a contribution value for the influence of the node v. Therefore, under the premise of maintaining the original diffusibility base, the influence contribution value of node u on node v should be subtracted. The specific formula is as follows:

$$db(v,i) = db(v,i) - \sum\nolimits_{v \in U'} b(v,u) \times ls^i \tag{3}$$

The result of this formula indicates the influence of node v which can diffuse to node u in i layers.

(2) Eliminate the superposition effects of diffusibility

Obviously, the effect of the activated node u on node v is through the indirect dissemination of the nodes that are on the path of node u to node v. Consider the following situation, node u has been selected as the seed node, assuming that node u has an effect on node v through node p, this effect is the indirect impact of node u on node v, which is included in the direct impact of node p on node v. Therefore, we need to subtract the partially superposed influence of node u on node v when calculating the diffusibility of node p. The specific formula is as follows:

$$db(p,c) = db(p,c) - db(u,c-i) \times ls^{i+1} \tag{4}$$

Algorithm 2. Influence Maximization Algorithm Based on Real-time and De-superimposed Diffusibility (RDD)

Input: graph: G(V, E),Θ, size of initial dissemination set: k, loss coefficient: ls, n: consider node spreading n layers
Output: initial dissemination set: s

1. Set s_0=Ø
2. **For each** node *u* in graph G do
   $b_{uv} \leftarrow 1/indegree\ of\ u$ /*calculate the effect of the active node u on the neighboring node v*/
3. **For each** node *u* in graph G **do**
   For each node *v* in neighbors of node *u* **do**
   For each *i* in *n*: /* consider node *u* spreading *n* layers */
   $db(u,i) \leftarrow db(u,i-1)+db(v,i-1)\times ls * b_{uv}$ /**db(u,i)* represents the diffusibility of node *u* spreading *i* layers */
   End For
   End For
   End For
4. **Loop** following steps for *k* times
5. Select node *u* with the largest diffusibility to join the set s_0
6. **Recursion** following steps for *n* depth: /* update the node diffusibility in real time*/
   Parameters: node *u*, depth *i*
   For each node *v* in neighbors of node *u* **do**
   $db(v,n) = db(v,n) - b_{uv} * ls^{i}$
   Recursion with set parameters *u* to *v*
   End For
   End Recursion
7. **Recursion** following steps for *n* depth: /*eliminate the superposition effects of diffusibility*/
   Parameters: node *now*, depth *i*
   For each node *v* in neighbors of node *now* **do**

$$db(v,n) = db(v,n) - db(u,n-i) \times ls^{i+1}$$

Recursion with set parameters *now* to *v*
End For
End Recursion
8. **End Loop**

The result of this formula (4) indicates the influence of node *p* which can diffuse to node *u* in *c* layers, and node *u* can diffuse to node *v* in *i* layers.

When a node is selected as a seed node, the nodes within *n* layers should be updated as above. The pseudo algorithm is shown in Algorithm 2.

Algorithm 2 is influence maximization algorithm based on real-time and de-superimposed diffusibility (RDD). This algorithm is designed based on Algorithm 1 by

updating the node diffusibility in real time and eliminating the superposition effects of diffusibility, which makes influence maximization more effective. The average time complexity of RDD is $O(E*(E/V)^{(n-1)})$, and the space complexity is $O(E)$. E is the number of edges. V is the number of nodes. n is the number of layers considered. RDD consumes little time than traditional heuristic algorithms like MaxDegree when n is not large, but it is much faster than Greedy Algorithm.

4 Experiments

4.1 Data Set

The experiment was conducted to verify the effectiveness of the algorithm proposed.

The first data set is Gnutella peer-to-peer network, which is derived from data sets published in the social networking field for various tests [24]. It is a snapshot of a series of Gnutella peer-to-peer file sharing networks. Nodes represent hosts in the Gnutella network topology. And edges represent connections between Gnutella hosts. The second data set is PGP network [25], which is a list of edges of the giant component of the network of users of the Pretty-Good-Privacy algorithm for secure information interchange. The data set is described in Table 1.

Table 1. The information of data set

	Gnutella p2p network	PGP network
Nodes	8717	10680
Edges	31525	24316
Average clustering coefficient	0.0067	0.26598
Number of triangles	1142	164.9K
Fraction of closed triangles	0.002717	0.377912

The effectiveness of the algorithm is reflected by the number of nodes that the selected seed nodes can affect through dissemination in final, which means the range that nodes can influence.

4.2 Results and Analysis

Experiments were conducted based on the linear threshold model. And the formula of buv is $buv = \frac{1}{d(v)}$. The value of the threshold is set as 0.8 for each node.

Considering the different number of layers, the results are shown in Figs. 1 and 2, respectively, with the loss coefficient being 0.2. It can be seen that only one layer of dissemination is considered to be less effective. The effect is much more remarkable on Gnutella p2p network. For 2–4 layers, the differences are insignificant. This is because the effect is weakened after the transmission of multi-layers. Therefore, there is no need to consider too many layers.

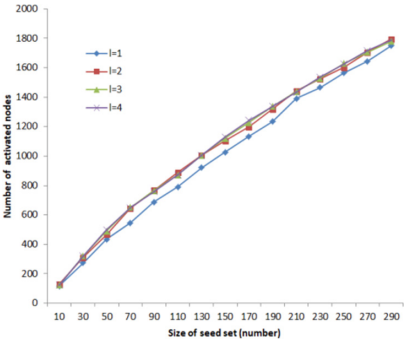

Fig. 1. The algorithm effectiveness of different layers on Gnutella p2p

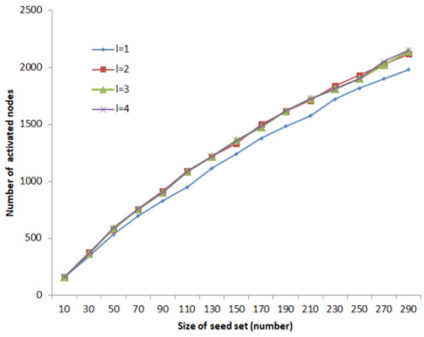

Fig. 2. The algorithm effectiveness of different layers on PGP

Considering the different value of loss coefficient, the result is shown in the Figs. 3 and 4 when the number of layers is 2. It can be seen that the effect of three different value of the loss coefficient is insignificantly different. On PGP network, the difference of effect among each loss coefficient is much smaller. The reason is that the algorithm RDD has been optimized. Loss coefficient has less effect on the algorithm. As for the Gnutella p2p network, when the loss coefficient is 0.2 the dissemination is slightly better than others, the loss coefficient is set as 0.2 in other experiments of this paper.

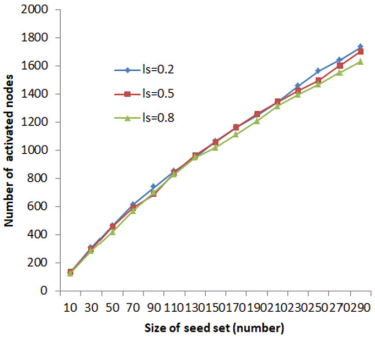

Fig. 3. The algorithm effect of different loss coefficient on Gnutella p2p

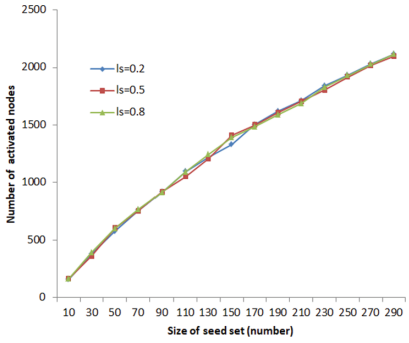

Fig. 4. The algorithm effect of different loss coefficient on PGP

Test the Pre-optimized Algorithm (NDC) and the Optimized Algorithm (RDD) The loss coefficient is set to 0.2 on these two data set, and the effect is shown in Figs. 5 and 6 when the number of layers is 2. It can be seen that the overlap of influence has a large effect on the result when there are more nodes selected in the seed set.

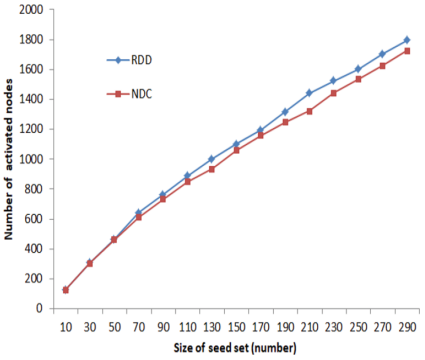

Fig. 5. The algorithm effect on Gnutella p2p (ls = 0.2)

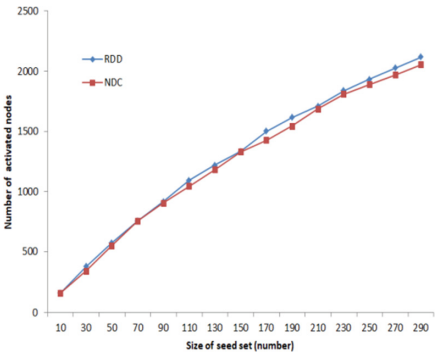

Fig. 6. The algorithm effect on PGP (ls = 0.2)

When the loss coefficient is large, the optimization still maintains good performance. But the performance before the optimization is greatly reduced as shown in Figs. 7 and 8. On PGP network, when the loss coefficient is 0.2, the influence has no significant effect. If we set the loss coefficient to 0.8, the optimization performs better.

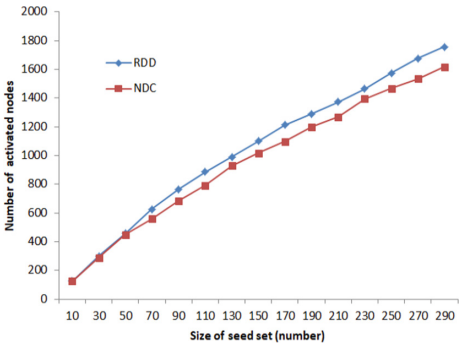

Fig. 7. The algorithm effect on Gnutella p2p (ls = 0.8)

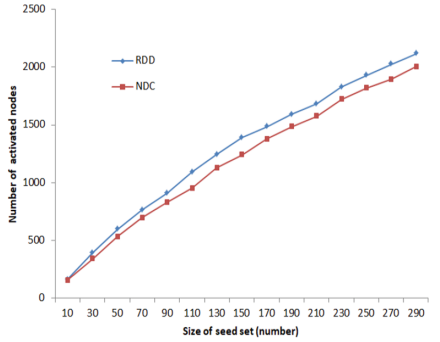

Fig. 8. The algorithm effect on PGP (ls = 0.8)

Comparison of Different Algorithms' Effect

We compared the proposed algorithm with two existing classical ranking-based algorithms, MaxDegree and PageRank. And the results are shown in Figs. 9 and 10, respectively. For Gnutella peer-to-peer network, the effectiveness of the algorithm is obviously better than the two existing algorithms. The difference between RDD and PageRank's effectiveness is not great on PGP network, but PageRank needs matrix calculation, which consumes a lot of space.

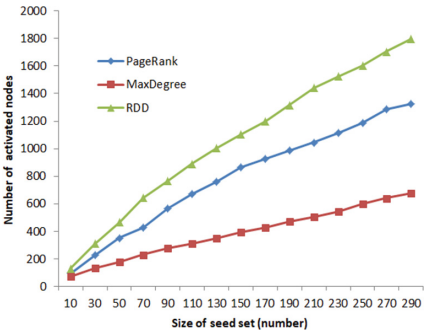

Fig. 9. Comparison of different algorithms' effect on Gnutella p2p

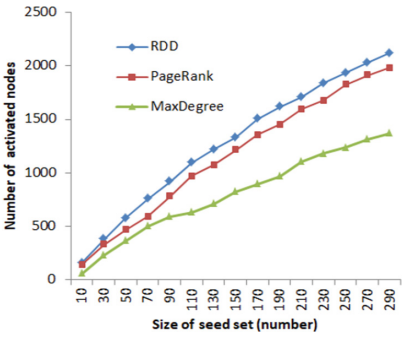

Fig. 10. Comparison of different algorithms' on PGP

Comparison of Time Complexity

When the threshold is set to 0.8 and the number of layers to 2, we can gain the comparisons of the time complexity among MaxDegree, PageRank and proposed algorithm. We found that the cost of time mainly happened in calculating the number of final activated nodes. The algorithm MaxDegree only calculates out-degree of nodes one time, and the cost of time mainly happened on calculating the number of final activated nodes. Therefore, the algorithm MaxDegree can be used as a benchmark for this time-consuming. When considering two layers of nodes, the algorithm consumes less and can get better results as shown in Fig. 11.

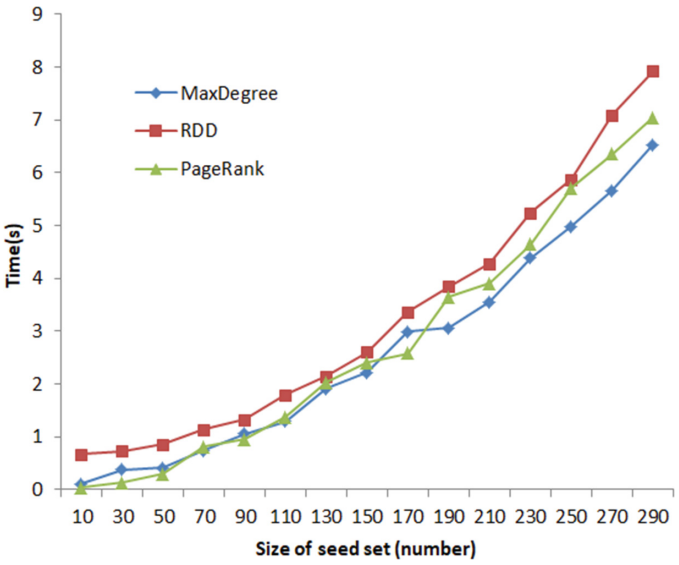

Fig. 11. The comparison of time complexity

5 Conclusion

This paper proposed a new concept, node diffusibility, and a measure metric of node influence. Node diffusibility takes the overall impact of the node and its neighbors into account. Moreover, in order to apply the diffusibility in the dissemination model, an influence maximization algorithm based on real-time and de-superimposed diffusibility was proposed. The algorithm reduces the coincidence of influence of seed node effectively. And the related experiments verified that the method based on the linear threshold model is effective. The results demonstrated that our proposed algorithm works well.

Our future work will focus on the effects of the algorithm on special networks such as weighted graphs, the influence of information timeliness on the result. Also, the influence of node characteristics on the effects of dissemination will be taken into account.

Acknowledgment. The research presented in this paper is supported by the National Key R&D Program of China (No. 2017YFE0117500) and the National Natural Science Foundation of China (No. 61762002).

References

1. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. ACM Trans. Web (TWEB) **1**(1), 5 (2007)
2. Bhattacharya, S., Gaurav, K., Ghosh, S.: Viral marketing on social networks: an epidemiological perspective. Stat. Mech. Appl., Physica A (2019)
3. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp. 199–208 (2009)
4. Aslay, C., Lakshmanan, L.V.S., Lu, W., et al.: Influence maximization in online social networks. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, ACM, pp. 775–776 (2018)
5. Zhang, Y.C., Liu, Y., Zhang, H.F., et al.: The research of information dissemination model on online social network. Acta Phys. Sin. **60**, 050501 (2011)
6. Richardson, M., Domingos, P.: Mining knowledge-sharing sites for viral marketing. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp. 61–70 (2002)
7. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp. 137–146 (2003)
8. Goyal, A., Lu, W., Lakshmanan, L.V.S.: Celf++: optimizing the greedy algorithm for influence maximization in social networks. In: Proceedings of the 20th International Conference Companion on World Wide Web, ACM, pp. 47–48 (2011)
9. Sánchez-Oro, J., Duarte, A.: Iterated greedy algorithm for performing community detection in social networks. Future Gener. Comput. Syst. **88**, 785–791 (2018)
10. Liu, G., Wang, Y., Orgun, M.A., et al.: A heuristic algorithm for trust-oriented service provider selection in complex social networks. In: 2010 IEEE International Conference on Services Computing, IEEE, pp. 130–137 (2010)

11. He, Q., Wang, X., Huang, M., et al.: Heuristics-based influence maximization for opinion formation in social networks. Appl. Soft Comput. **66**, 360–369 (2018)
12. Pathak, N., Banerjee, A., Srivastava, J.: A generalized linear threshold model for multiple cascades. In: 2010 IEEE International Conference on Data Mining, IEEE, pp. 965–970 (2010)
13. Estevez, P.A., Vera, P., Saito, K.: Selecting the most influential nodes in social networks. In: 2007 International Joint Conference on Neural Networks, IEEE, pp. 2397–2402 (2007)
14. Ma, Q., Ma, J.: Identifying and ranking influential spreaders in complex networks with consideration of spreading probability. Physica A Stat. Mech. Appl. **465**, 312–330 (2017)
15. Opsahl, T., Agneessens, F., Skvoretz, J.: Node centrality in weighted networks: generalizing degree and shortest paths. Soc. Networks **32**(3), 245–251 (2010)
16. Freeman, L.C.: Centrality in social networks conceptual clarification. Social Networks **1**(3), 215–239 (1978)
17. Okamoto, K., Chen, W., Li, X.-Y.: Ranking of closeness centrality for large-scale social networks. In: Preparata, Franco P., Wu, X., Yin, J. (eds.) FAW 2008. LNCS, vol. 5059, pp. 186–195. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-69311-6_21
18. Goh, K.I., Oh, E., Kahng, B., et al.: Betweenness centrality correlation in social networks. Phys. Rev. E **67**(1), 017101 (2003)
19. Ding, Y., Yan, E., Frazho, A., et al.: PageRank for ranking authors in co-citation networks. J. Am. Soc. Inf. Sci. Technol. **60**(11), 2229–2243 (2009)
20. Frahm, K.M., Shepelyansky, D.L.: Ising-PageRank model of opinion formation on social networks, p. 121069. Stat. Mech. Appl., Physica A (2019)
21. Kitsak, M., Gallos, L.K., Havlin, S., et al.: Identification of influential spreaders in complex networks. Nat. Phys. **6**(11), 888 (2010)
22. Cao, J.X., Dong, D., Xu, S., et al.: A k-core based algorithm for influence maximization in social networks. Chin. J. Comput. **38**(2), 238–248 (2015)
23. Wang, Y., Cong, G., Song, G., et al.: Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp. 1039–1048 (2010)
24. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graph evolution: densification and shrinking diameters. ACM Trans. Knowl. Discovery from Data (ACM TKDD) **1**(1), 2 (2007)
25. Boguná, M., Pastor-Satorras, R., Díaz-Guilera, A., et al.: Models of social networks based on social distance attachment. Phys. Rev. E **70**(5), 056122 (2004)