An Improved Linear Threshold Model

Xiaohong Zhang[1], Nanqun He[1], Kai Qian[1], Wanquan Yang[2], and Jianji Ren[1](✉)

[1] College of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454000, China
{xh.zhang,renjianji}@hpu.edu.cn, 211809010012@home.hpu.cn, qkai237@126.com

[2] Henan College of Survey and Mapping, Zhengzhou 451464, China
ywanquan@126.com

Abstract. Linear threshold model is one of the widely used diffusion models in influence maximization problem. It simulates influence spread by activating nodes for an iterative way. However, the way it makes activation decisions limits the convergence speed of itself. In this paper, an improvement is proposed to speed up the convergence of Linear threshold model. The improvement makes activation decisions with one step ahead considering the nodes which will be activated soon. To assist in activation decision making, the improvement introduces a new state and updates state transition rules. The experiment results verify the performance and efficiency of the improvement.

Keywords: Linear threshold model · Performance · Influence maximization · Convergence speed

1 Introduction

Social network platforms such as Facebook, WeChat, Twitter, etc., have become widely used mediums that people communicate with each other. As one of the most important problems in social network analysis, the problem of influence maximization has attracted tremendous attentions [15,20,26]. It aims to find a small set of influential nodes so that the influence of those nodes can be spread most widely. Although the problem is first raised in the field of marketing, it exists in many other fields such as political movements [16], rumor controlling [29,31], and so on.

Diffusion models are critical for solving the influence maximization problem since they can simulate the spread of influence, Linear threshold model (LT model) [17] is one of the most widely used diffusion models. It simulates influence spread by activating nodes in an iterative way. In each iteration, a activation decisions are made only according to the nodes activated in previous iterations. The nodes activated in current iteration are not counted for any activation decision made in the same iteration, which limits the convergence speed

X. Wang et al. (Eds.): 6GN 2020, LNICST 337, pp. 333–344, 2020.
https://doi.org/10.1007/978-3-030-63941-9_25

of LT model. Although many efforts have been made on the improvement of LT model [2,7,23,27], how to improve the convergence speed of LT model is still an open issue.

In this work, we focus on the problem of the convergence speed of LT model. To deal with the problem, an improvement is proposed to activate nodes with one step ahead considering the nodes which will be activated. Concretely, the improvement makes activation decisions according to not only active nodes but also the inactive nodes to be activated soon. To identify those inactive nodes, the improvement introduces a new state for them, that is, ready-active state, and new state transition rules.

The rest of this paper is organized as follows: related work is reviewed in Sect. 2 followed by motivation in Sect. 3. The proposed approach is elaborated in Sect. 4 and evaluated in Sect. 5. Finally, the paper is concluded in Sect. 6.

2 Related Work

Influence maximization is one of the most important problems in the areas of social network analysis, and has attracted much attentions in recent years. Domingos and Richardson [8,24] first study the influence maximization problem in probabilistic environments. Kempe et al. [17] prove that the influence maximization problem is an NP-hard problem, and propose LT model to simulate the process of information diffusion. In LT model, each node of a social network is only in one of the following two states: inactive state and active state. LT model starts with the assumption that all the nodes are in the inactive state except the set of seed nodes which are initialized in the active state.

Many researches have been carried out to select the set of seed nodes. Kempe et al. [17] propose a simple greedy algorithm approximating the optimum with a factor of $(1-1/e)$. Various variants of the greedy algorithm, e.g. CELF [18], CELF++ [12], constrained greedy algorithm [33], etc., have been proposed to improve the efficiency of the simple greedy algorithm. In addition to greedy algorithms, many different algorithms, e.g., centrality based algorithms [9,10,25], community based algorithms [3,19,28,32], influence estimation based algorithms [21,22], etc., are also exploited to identify the set of seed nodes. Besides the work on the selection of seed nodes, many efforts are put on the improvement of LT model.

Various researches concern on competitive environments in which more than one player competes with each other to influence the most nodes. He et al. [13] proposed a competitive LT model to block the influence of competing products. Bozorgi et al. [3] extend LT model to give each node a decision-making ability about incoming influence spread. Galhotra et al. [11], Zhang et al. [30] and Yang et al. [29] concern on the improvement of LT model to allow a node to be either positive or negative in the speed process of information or influence. Zhang et al. [14,34] and Caliò et al. [5] revise LT model to spread influence according to trust/distrust relationships. Chan et al. [6] propose the non-progressive LT model to deal with the case in which active nodes may become inactive. Although much efforts have been made on the improvement of LT model, the convergence problem of the model is still an open problem.

For the convenience of presentation, all the notations in this work are described in Table 1.

Table 1. Notations

Notations	Description
$\boldsymbol{G}$	A directed graph
$\boldsymbol{E}$	The edge set of $\boldsymbol{G}$
$\boldsymbol{V}$	The vertex set of $\boldsymbol{G}$
v	A vertex, e.g., v_i represents the i_{th} vertex of $\boldsymbol{V}$
θ_a	Activation threshold of a node
$\boldsymbol{N_i^a}$	The active neighbor set of v_i
$\boldsymbol{N_i^r}$	The ready-active neighbor set of v_i
$State(v_i)$	The state of node v_i
Inf^{To}	The influence to a node
Inf_p^{To}	The potential influence to v_i
$Inf(v_i, v_j)$	The influence of v_i to v_j

3 Motivation

LT model is proposed to simulate the spread of influence by Kempe et al. [17]. In LT model, a social network is denoted as a directed graph $G = (V, E)$, where V represents the set of nodes, and E expresses the set of directed edges. Each node can only be in one of the following two states: the active state and the inactive state. $((\forall v_i) v_i \in V)$, v_i can transmit from the inactive state to the active state if function (1) is satisfied. Otherwise, it stays in the inactive state.

$$\Sigma_{v_j \in N_{v_i}^a} Inf(v_j, v_i) \geq \theta_a \tag{1}$$

LT model starts with a small set of seeds. All the nodes regarded as seeds are initialized in the active state while all the other nodes in inactive state. LT model updates the states of nodes in an iterative way and converges if no inactive nodes are activated any more. It decides whether an inactive node can be converted to the active state according to the influence from its neighbors which are activated in previous iterations. The influence from the nodes being activated in the current iteration is not counted, which limits the convergence speed of LT model.

In this work, we aim to make an improvement on LT model to accelerate its convergence. The improvement makes activation decisions according to the influence from not only active neighbors but also the inactive neighbors which will change into the active state.

4 LT Model with One Step Forward Looking

In this section, we elaborate the improvement of LT model. The improvement aims to speed up the convergence speed of LT model with one step ahead considering the nodes which will be activated. Concretely, in each iteration, the improvement calculates the influence to an inactive node according to its active neighbors and inactive neighbors which will be activated soon. If the calculated influence overpasses θ_a, the inactive node changes into the active state.

4.1 State Transition

To activate a node, the improvement calculates the influence from its active neighbors, and some of its inactive neighbors. Here, the inactive neighbors indicate the inactive nodes which will change from the inactive state to the active state in the next iteration. To distinguish those inactive neighbors form other inactive neighbors, we introduce the ready-active state.

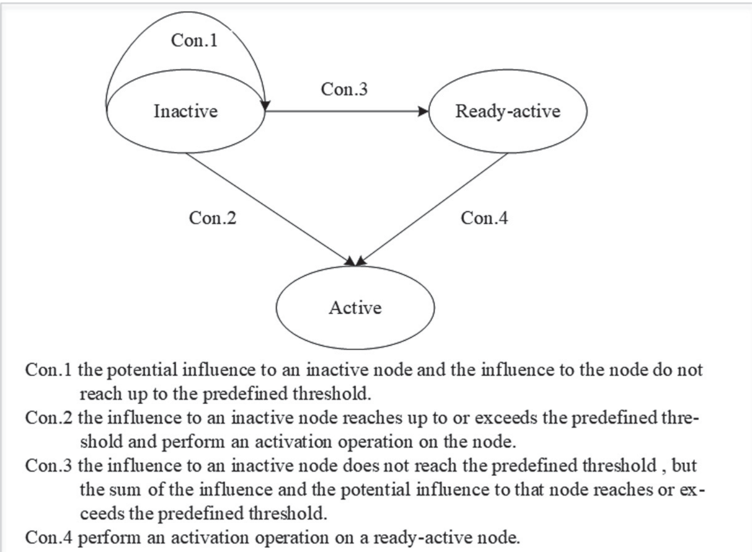


Fig. 1. Node state transition

Definition 1. ***Ready-active state.*** $(\forall v_i)((v_i \in V)$ *and* $State(v_i) = inactive)$, *v_i is in the ready-active state if v_i satisfies the following conditions: (1) the influence from all the active neighbors of v_i reaches up to or overpasses the predefined threshold, and (2) no activation operation has been performed on v_i.*

Definition 2. ***Potential influence.*** $(\forall v_i)((v_i \in V)$ *and* $State(v_i) = inactive)$, *the potential influence of* v_i *indicates the influence from its ready-active neighbors. It is calculated by function (2).*

$$Inf_p^{To} = \Sigma_{v_j \in N_i^r} Inf(v_j, v_i) \tag{2}$$

In the function, $inf(v_j, v_i)$ represents the influence from v_j to v_i, N_i^r represents the ready-active neighbor set of v_i. It can be calculated according to the existing influence models. After adopting the ready-active state, state transition is performed according to the following rules:

Rule 1. Perform an activation operation on an inactive node if the influence from the active neighbors of that node reaches or overpasses an predefined threshold, and change that node into the active state.

Rule 2. Change an inactive node into the ready-active state if the influence from its active neighbors does not reach the predefined threshold, but the influence from its active neighbors and the potential influence from its ready-active neighbors reach or exceed the predefined threshold.

Rule 3. Change the state of a ready-active node into the active state after performing an activation operation on that node.

Figure 1 shows the state transition in the improvement. According to the figure, an inactive node can be transited into the ready-active state or active state. It can also keep being in the inactive state. If Con.1 is satisfied, it keeps being in the inactive state. If Con.2 is satisfied, it changes into the active state. If Con.3 is satisfied, it changes into the ready-active state. A ready-active state can only change into the active state only if Con.4 is satisfied.

4.2 Influence Propagation

The improved model propagates influence in an iterative way as LT model does. However, it is different from LT model by considering ahead the inactive node to be activiated soon, that is, the actve-ready nodes. It first initializes seed nodes in the active state and all other nodes in the inactive state and starts iterations. In each iteration, it calculates the influence and the potential influence to each inactive node, and performs state transition according to Rules 1 to 3. The iteration converges if the difference in the number of the nodes activated during two adjacent iterations is less than a predefined threshold.

Algorithm 1 shows the influence propagation in the improved model. We take Fig. 2 as an example to illustrate the influence spread with the improved model. In the initial state, v_2 and v_5 are regarded as seed nodes, the states of the two nodes are active, and the remaining nodes are inactive.

In the first iteration, v_2 has an influence on v_1 exceeds θ_a and v_1 transitions to an active state. Similarly, v_4 also becomes active. The sum of influence of v_2 on v_3 and the potential influence of v_4 on v_3 exceeds θ_a, and v_3 is changed into

Algorithm 1. Influence propagation in the improved model

Input:: seeds, G(V,E)
Output: total number of the nodes activated

```
actives ← seeds
do
    num=actives.size();
    inactives ← obtain the inactive neighbors of the nodes in seeds
    for each node in inactive do
        calculate the influence to that node, that is Inf^{To};
        if Inf^{To} ≥ θ_a then
            activate the node and updata the state of that node to be active;
            insert that node to actives
        else
            calculate the potentional influence to that node, that is Inf_p^{To}
            if Inf^{To} + Inf_p^{To} ≥ θ_a then
                activate v_i
                insert v_i to actives
            end if
        end if
    end for
while (actives.size()–num ≥ ξ)
return num
```

the ready-active state, the activation operation is performed on v_3, and v_3 is changed into the active state. The remaining nodes keep being inactive.

In the second iteration, the influence to v_6 comes from v_3 overpasses θ_a, and v_6 is changed into the active state. The sum of the influence of node v_5 on v_7 and the potential influence of v_6 on v_7 exceeds θ_a, so v_7 changes into the ready-active state, the activation operation is performed on v_7, and v_7 is changed into an active state. The remaining nodes keep being inactive.

In the third iteration, v_7 has an influence on v_8 that exceeds θ_a, so v_8 changes into the active state. Similarly, v_9 is also activated. The other node states remain unchanged.

In the fourth iteration, no nodes can be activated, so the iteration converges.

5 Experiment and Evaluation

In this section, we elaborate the abundant experiments conducted to evaluate the improved model by comparing with LT model, and discuss the experiment results.

5.1 Data Sets and Environment

We use six-real-world data sets in our experiments. All these data sets are collected from Stanford Large Network Dataset Collection [1]. All these data sets are in different scales and expressed in directed graphs. The details of those data sets are described below.

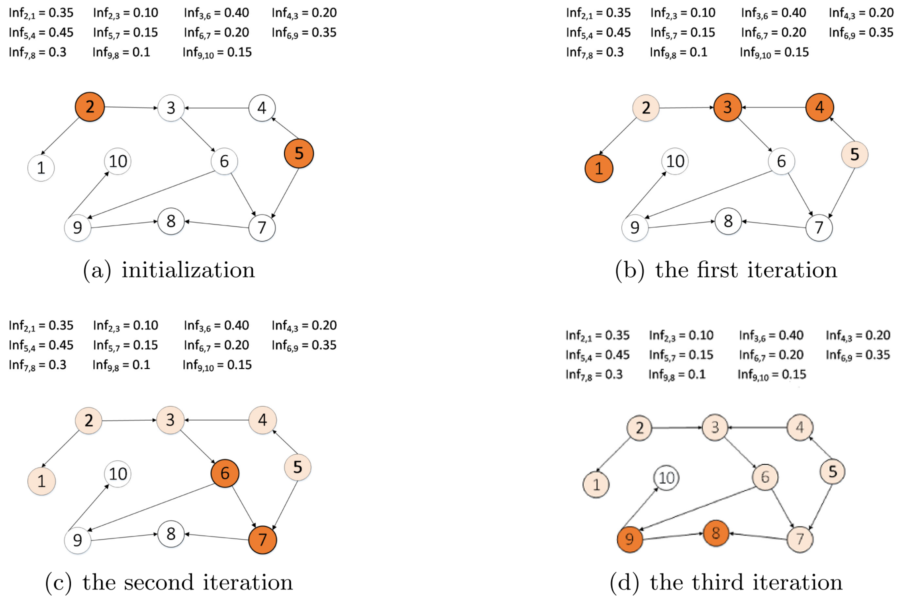
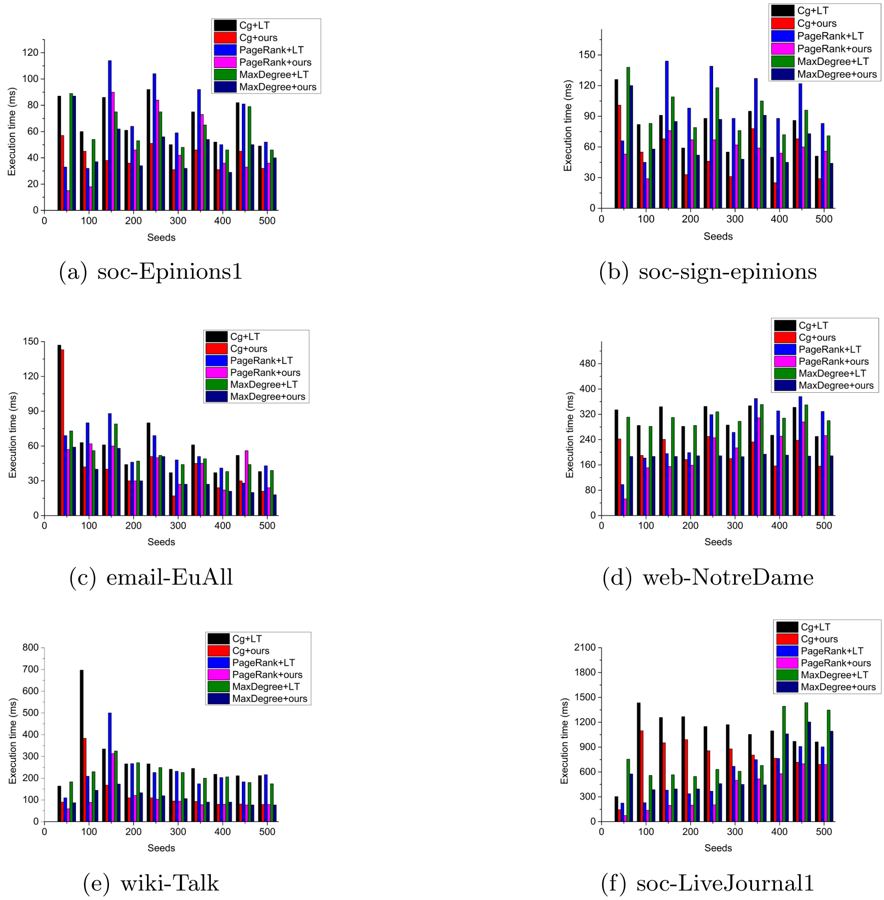


Fig. 2. Information dissemination based on the improved model

- soc-Epinions1 data set: This data is collected according to the trust relationship interaction on the website. It includes 75879 nodes and 508837 edges. Each vertex describes a reviewer and each edge denotes the trust relationship between two reviewers.
- soc-sign-epinions data set: This data set is extracted from Epinions.com. It contains about 131828 nodes and 841372 edges. Each vertex represents a user, and edge describes one user described by one vertex trusting the other user described by the other vertex.
- email-EuAll data set: This network is generated according to the email data from a large European research institution. It includes 265214 nodes and 420045 edges. Each node corresponds to an email address. Each edge describes that at least one email is sent from one node to the other node.
- web-NotreDame data set: This data set is collected from the web site of the University of Notre Dame. It includes 325729 nodes and 1497134 edges. Nodes represent pages from the web site and edges represent hyperlinks between those pages.
- wiki-Talk data set: This data set is collected from Wikipedia which is a free encyclopedia written collaboratively by volunteers around the world. It includes 2394385 nodes and 5021410 edges. A node represents a Wikipedia user. An edge represents one user at least edits a talk page of the other node.
- soc-LiveJournal1 data set: This data set is collected from a free online community with almost 10 million members. It includes 4847571 nodes and 6899373 edges. Nodes represent members and edges represent friendship of members.



(a) soc-Epinions1

(b) soc-sign-epinions

(c) email-EuAll

(d) web-NotreDame

(e) wiki-Talk

(f) soc-LiveJournal1

Fig. 3. Execution time on different data sets

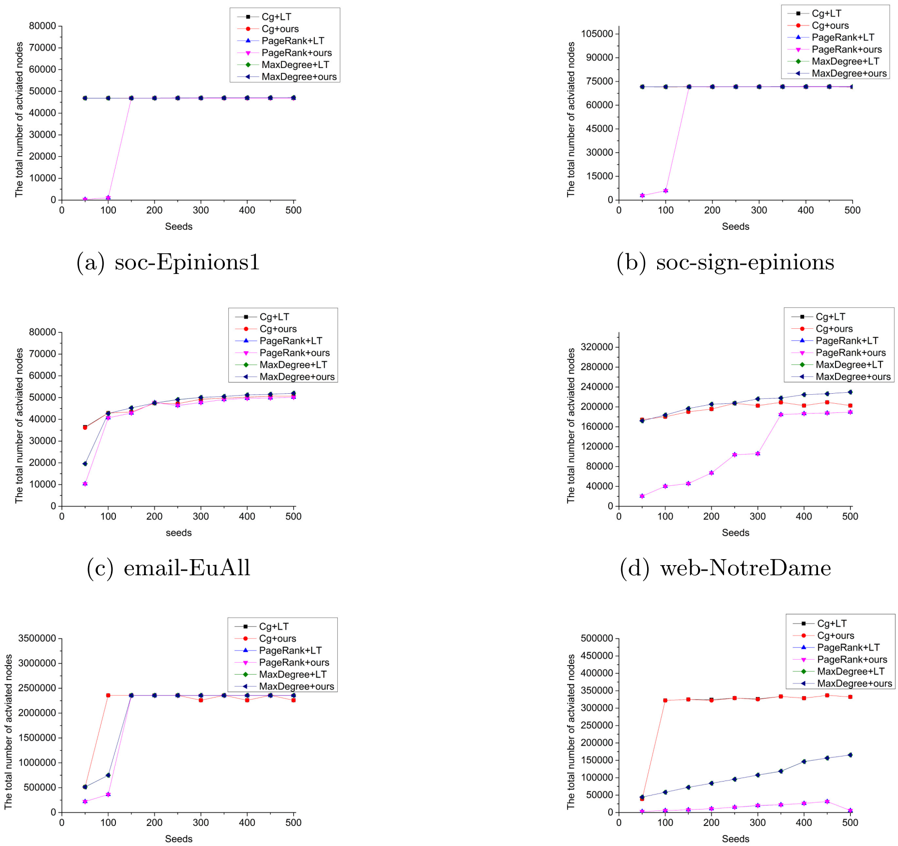
Table 2. Configuration of the big data platform

Hardware layer		HDFS		Spark	
Audit/Node	24	File block	256	Parallel tasks/actuators	7
Memory/Node	32G	Number of copies	3	Working node A	19
Hard disk/Node	3TB	Data node	19	Actuator number/work node	1
Tocal nodes	20	Nodes	19	Worker	1

All the experiments are conducted on the big data platform which is deployed on the cluster consisting of 20 severs. The details of the platform are described in Table 2. We choose execution time and the total number of activated nodes as the metrics to evaluate our work.

5.2 Result Discussion

The evaluation of our work is performed by comparing with LT model when different seed selection algorithms are adopted. Cg denotes the seed selection algorithm based on influence spread in a constrained greedy way algorithms [33]. MaxDegree and PageRank describe the algorithm selecting seeds according to Max degree policy [17] and PageRank algorithm [4], respectively.



(a) soc-Epinions1

(b) soc-sign-epinions

(c) email-EuAll

(d) web-NotreDame

(e) wiki-Talk

(f) soc-LiveJournal1

Fig. 4. Actived node comparison

Cg+ours, MaxDegree+ours and PageRank+ours denote the results of the corresponding seed selection algorithms combined with our model, respectively. Cg+LT, MaxDegree+LT and PageRank+LT describe the results of those algorithms with LT model, respectively. All the results shown in this subsection are the average results of 100 independent runnings.

Figure 3 shows the execution time of our model and LT model on different data sets. In the figure, the x axis represents the total number of seeds selected by different algorithms. The y axis expresses the execution time of our model and LT model spreading influence with the selected seeds. According to the figure, our model converges faster than LT model on all data sets no matter which algorithm is used to select seeds.

More details, our model performs best on Soc-Epinions1 data set, soc-sign-epinions data set, wiki-Talk data set and soc-LiveJournal1 data set with 450 seeds, 350 seeds, 500 seeds and 50 seeds selected by PageRank algorithm, respectively. It performs best on email-EuAll data set and web-NotreDame data set with 450 seeds selected by MaxDegree algorithm. Our model improves the performance of LT model by about 59%, 53%, 54%, 46%, 63% and 67% on Soc-Epinions1 data set, soc-sign-epinions data set, email-EuAll data set, web-NotreDame data set, wiki-Talk data set and soc-LiveJournal1 data set in the best case.

In the worst case, our model still obtains obvious performance improvement on four data sets, that is, Soc-Epinions1 data set, web-NotreDame data set, wiki-Talk data set and soc-LiveJournal1 data set. It improves the performance by 13% at least and 37% at most on these four data sets. Our model exhibits the similar performance with LT model on soc-Epinion1 data set and email-EuAll data set.

Averagely, our model obtains performance improvement on 32.7%, 33.75%, 34.01%, 31.6%, 54.38% and 32.3% on the six data sets, respectively.

To evaluate the ability to spread influence, we compare the models on six data sets with the same seeds, respectively. The corresponding results are described in Fig. 4, where the x axis and y axis represents the amount of seeds and that of the activated nodes, respectively. According to the figure, our model activates the same number of nodes as LT model, that is, our model exhibits the similar ability to spread influence to LT model. That means our model improves the performance of influence spread while guaranteeing the efficiency when comparing with LT model.

6 Conclusion

LT model is one of the diffusion models widely used in the solutions of influence maximization. In this work, we focus on the convergence problem of LT model. We analyze the reasons limiting the convergence speed of the model and propose an improvement of that model. The improved model makes activation decisions according to not only active nodes but also the inactive nodes to be activated soon. The experiment results on abundant datasets verify the performance and efficiency of the improved model. In the future, we will research the application of Linear threshold model in competitive environments.

References

1. http://snap.stanford.edu/data/ (2020)
2. Borodin, A., Filmus, Y., Oren, J.: Threshold models for competitive influence in social networks, pp. 539–550 (2010)
3. Bozorgi, A., Samet, S., Kwisthout, J., Wareham, T.: Community-based influence maximization in social networks under a competitive linear threshold model. Knowl. Based Syst. **134**, 149–158 (2017)
4. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Comput. Netw. ISDN Syst. **30**, 107–117 (1998)
5. Caliò, A., Tagarelli, A.: Complex influence propagation based on trust-aware dynamic linear threshold models. Appl. Netw. Sci. **4**(1), 1–41 (2019). https://doi.org/10.1007/s41109-019-0124-5
6. Chan, T.H.H., Ning, L.: Influence maximization under the non-progressive linear threshold model. arXiv Social and Information Networks (2015)
7. Chen, W., et al.: Influence maximization in social networks when negative opinions may emerge and propagate, pp. 379–390 (2011)
8. Domingos, P., Richardson, M.: Mining the network value of customers. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 57–66. ACM (2001)
9. Freeman, L.C.: A set of measures of centrality based on betweenness. Sociometry **40**(1), 35–41 (1977)
10. Freeman, L.C.: Centrality in social networks conceptual clarification. Social Netw. **1**(3), 215–239 (1978)
11. Galhotra, S., Arora, A., Roy, S.: Holistic influence maximization: combining scalability and efficiency with opinion-aware models, pp. 743–758 (2016)
12. Goyal, A., Lu, W., Lakshmanan, L.V.S.: CELF++: optimizing the greedy algorithm for influence maximization in social networks, pp. 47–48 (2011)
13. He, X., Song, G., Chen, W., Jiang, Q.: Influence blocking maximization in social networks under the competitive linear threshold model, pp. 463–474 (2012)
14. Hosseinipozveh, M., Zamanifar, K., Naghshnilchi, A.R.: Assessing information diffusion models for influence maximization in signed social networks. Expert Syst. Appl. **119**, 476–490 (2019)
15. Huang, H., Shen, H., Meng, Z., Chang, H., He, H.: Community-based influence maximization for viral marketing. Appl. Intell. **49**(6), 2137–2150 (2019). https://doi.org/10.1007/s10489-018-1387-8
16. Katz, E., Lazarsfeld, P.F., Roper, E.: Personal influence : the part played by people in the flow of mass communications. Am. Sociol. Rev. **17**(4), 357 (1956)
17. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence in a social network. In: Proceeding of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2003)
18. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., Vanbriesen, J.M., Glance, N.: Cost-effective outbreak detection in networks, pp. 420–429 (2007)
19. Li, X., Cheng, X., Su, S., Sun, C.: Community-based seeds selection algorithm for location aware influence maximization. Neurocomputing **275**, 1601–1613 (2018)

20. Liu, W., Chen, X., Jeon, B., Chen, L., Chen, B.: Influence maximization on signed networks under independent cascade model. Appl. Intell. **49**(3), 912–928 (2018). https://doi.org/10.1007/s10489-018-1303-2
21. Lu, W., Zhou, C., Wu, J.: Big social network influence maximization via recursively estimating influence spread. Knowl. Based Syst. **113**, 143–154 (2016)
22. Lu, Z., Fan, L., Wu, W., Thuraisingham, B., Yang, K.: Efficient influence spread estimation for influence maximization under the linear threshold model. Comput. Soc. Netw. **1**(1), 1–19 (2014). https://doi.org/10.1186/s40649-014-0002-3
23. Pathak, N., Banerjee, A., Srivastava, J.: A generalized linear threshold model for multiple cascades, pp. 965–970 (2010)
24. 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, pp. 61–70. ACM (2002)
25. Sabidussi, G.: The centrality index of a graph. Psychometrika **31**(4), 581–603 (1966)
26. Saxena, B., Kumar, P.: A node activity and connectivity-based model for influence maximization in social networks. Soc. Netw. Anal. Min. **9**(1), 1–16 (2019). https://doi.org/10.1007/s13278-019-0586-6
27. Trpevski, D., Tang, W.K., Kocarev, L.: Model for rumor spreading over networks. Phys. Rev. E **81**(5), 056102 (2010)
28. Wang, Y., Cong, G., Song, G., Xie, K.: Community-based greedy algorithm for mining top-k influential nodes in mobile social networks, pp. 1039–1048 (2010)
29. Yang, L., Li, Z., Giua, A.: Containment of rumor spread in complex social networks. Inf. Sci. **506**, 113–130 (2020)
30. Zhang, H., Dinh, T.N., Thai, M.T.: Maximizing the spread of positive influence in online social networks, pp. 317–326 (2013)
31. Zhang, R., Li, D.: Identifying influential rumor spreader in social network. Discrete Dyn. Nat. Soc. **2019**, 1–10 (2019)
32. Zhang, X., Zhu, J., Wang, Q., Zhao, H.: Identifying influential nodes in complex networks with community structure. Knowl. Based Syst. **42**, 74–84 (2013)
33. Zhang, X., Li, Z., Qian, K., Ren, J., Luo, J.: Influential node identification in a constrained greedy way. Physica A **557**, 124887 (2020). https://doi.org/10.1016/j.physa.2020.124887
34. Zhang, Y., Wang, Z., Xia, C.: Identifying key users for targeted marketing by mining online social network, pp. 644–649 (2010)