



# Influential Nodes Detection in Dynamic Social Networks

Nesrine Hafiene<sup>1,2</sup>(✉), Wafa Karoui<sup>1,3</sup>, and Lotfi Ben Romdhane<sup>1,2</sup>

<sup>1</sup> Laboratoire MARS LR17ES05, ISITCom, Université de Sousse,  
4011 Sousse, Tunisie

hafiene.nesrine@gmail.com, karoui.wafa@gmail.com,  
lotfi.ben.romdhane@gmail.com

<sup>2</sup> ISITCom, Université de Sousse, 4011 Sousse, Tunisie

<sup>3</sup> ISI, Université de Tunis El Manar, 2080 Tunis, Tunisie

**Abstract.** The influence maximization problem aims to identify influential nodes allowing to reach the viral marketing objectives on social networks. Previous researches are mainly concerned with the static social network analysis and the development of algorithms in this context. However, when network changes, those algorithms must be updated. In this paper, we offer a new interesting approach to study the influential nodes detection problem in changing social networks. This approach can be considered to be an extension of a previous static algorithm SND (Semantic and structural influential Nodes Detection). Experimental results prove the effectiveness of SNDUpdate to detect influential nodes in dynamic social networks.

**Keywords:** Dynamic social networks · Influence maximization · Influence propagation · Influential nodes

## 1 Introduction

In recent years, with the increasing popularity of social networks like Facebook and Twitter, more and more scientists who study the influence maximization problem pay their attention to this field. Indeed this problem has drawn much attention and many researches have proposed various algorithms to detect influential nodes, most of these methods are based on static social networks. However, real social networks keep changing during time. New connections between users are created and some other users lose contact. Since many influential nodes can be detected by carefully studying the relationships among the links, these ones play an important role in dynamic social network analysis.

A social network looks like a graph structure consisting of nodes and edges, where nodes represent users and edges represent the interactions between the connected users. Users develop their connections to each other, while interactions between them vary over time. A changing social network consists of social

---

MARS—Modeling of Automated Reasoning Systems.

© Springer Nature Switzerland AG 2019

W. Abramowicz and R. Corchuelo (Eds.): BIS 2019, LNBIP 354, pp. 62–73, 2019.

[https://doi.org/10.1007/978-3-030-20482-2\\_6](https://doi.org/10.1007/978-3-030-20482-2_6)

networks observations at different time stamps  $\{G_1, G_2, \dots, G_n\}$  and contains not only a set of relationships between nodes, but also information on how these relationships change in each time stamp.

To resolve the above drawbacks, we propose an extension of SND to tackle the problem of important nodes detection under changing social networks. The remainder of this paper is organized as follows. The related works are reviewed in Sect. 2. In Sects. 3 and 4, the proposed extension SNDUpdate for maximizing the influence propagation is described. The details of the experiments are presented in Sect. 5. Finally, the paper is concluded in Sect. 6.

## 2 Related Work

This section discusses the review of different researches done on dynamic social networks. In this paper, we concentrate on the problem of influential nodes detection in edges changing social networks.

### 2.1 Methods Based on a Non-linear Model

Aggarwal et al. proposed [7] the first paper which deals with the problem of information flow authority determination in dynamic networks based on temporary interactions. Moreover, this paper studies both the problem of influence propagation, and that of tracking back from a given pattern of spread the influential nodes. They considered how to discover a set of nodes having the highest influence within a time. They modelled the influence spread as a non-linear system which is very different from triggering models like the Linear Threshold model or the Independent Cascade model. The algorithm in [7] is heuristic and the produced results have not any provable quality guarantee. There are only a few papers focusing on Influence Maximization under changing networks. Aggarwal et al. [7] focuses on finding a seed set at time  $t$ , that maximizes the influence spread at some  $[t + \Delta]$  given the dynamics of the evolution of networks over interval time  $[t, t + \Delta]$ . In this paper we consider to maximize the influence under a serie of snapshots taken from a social network. Zhuang et al. [18] study the influence maximization problem under dynamic networks where the changes can be only detected by occasionally probing some nodes. The main idea of their paper is how to choose a subset of nodes so that the actual influence propagation is improved.

### 2.2 Methods Based on Metrics

During the years, some centrality metrics have been proposed to estimate node importance. It is well-known that degree centrality, betweenness centrality and closeness centrality are three basic centrality measures to identify influential nodes. However, all of these centrality measures are designed for static networks and they looked over the feature that networks are usually dynamic. Motivated by this deficiency, to apply those metrics in changing networks we added information about the time of edges evolution.

Kitsak et al. [1] proposed *k-shell* decomposition and they discover that the influential nodes are those positioned in the core of the network. The *k-shell* decomposition assigns many nodes in the same K-shell. Basaras et al. [6] proposed an alternative measure, the  $\mu$ -power community index, that is an combination of coreness and betweenness centrality,  $\mu$ -PCI is calculated in a totally localized manner and thus is appropriated for any type of networks ignoring its size or dynamicity. Wei et al. [15] proposed two classes of dynamic metrics to estimate temporal evolution of agents with regard to persistence and emergence. Here, the network activities are measured per time stamp using static network metrics such as degree centrality, authority centrality or clustering coefficients. Ren et al. [14] proposed a new approach to identify influential nodes in a complex network called Evidential K-shell centrality based on edge weight. The authors study only undirected networks, while many surveys could be done on identifying influential nodes in directed networks. As reported by Wang et al. [5], nodes in the network can be evaluated by centrality metrics. Therefore, they defined influential nodes in a dynamic community as nodes which have the most important centrality scores higher than the other nodes and have comparatively long life in this community.

### 2.3 Methods Based on a Diffusion Model

Chen et al. [8], included time in the influence diffusion model to track influential nodes in changing social networks. The main idea is to choose a budgeted subset to maximize the influence propagation  $\varphi$  time stamps later. However, they assume that the dynamic network is completely observed. This is impossible in many real situations. Ohsaka et al. [16] studied a related problem, maintaining some RR sets over a stream of networks updates under the IC model such that approximation influence maximization can be completed with a fixed probability. Tong et al. [13] proposed the Dynamic Independent Cascade (DIC) model by extending the classic IC model, DIC is able to better take control of the dynamic aspects of real social networks. Liu et al. [4] proposed an incremental approach, IncInf, which can efficiently locate the top- $k$  influential nodes in changing social networks based on previous information. Song et al. [3] proposed Upper Bound Interchange Greedy (UBI) algorithm for the Influential Nodes Tracking (INT) problem, in which they found the seed set that augments the influence under  $G^{t+1}$  based on the seed set  $S^t$  that they have effectively found in  $G^t$ . Then they proposed UBI+ algorithm that enhances the computation of the upper bound and achieves better influence propagation. The authors in [9] modelled a changing network as a stream of edge weight updates. In [10], the authors have systematically tackle two essential tasks of tracking *top - k* influential nodes. Their goal is to control the error incurred in their algorithm [9], so that even without prior knowledge about the data, they can still obtain meaningful results by setting a relative error threshold.

### 3 Preliminaries and Problem Statement

In this section, we first introduce the previous diffusion model for static networks. We then present our extension algorithm as a generalization of the influential nodes detection problem to dynamic social networks.

#### 3.1 Problem Statement

The previous algorithm SND aims to detect influential nodes for only static social networks. However, real social networks are dynamic so both the structure and also the influence propagation associated with the edges are constantly changing. As a result, according to the evolution of the network structure and the influence propagation, the leader nodes that maximizes the influence propagation should be changed. In this paper, we model the dynamic social network as a group of snapshot graphs where nodes remain the same while the edges in each snapshot graph change over different time intervals. We denote the snapshot graph as  $G^t = (V, E^t)$ , where  $V = \{v_1, \dots, v_i, \dots, v_n\}$  represents nodes and  $E$  represents edges appearing during time intervals. Notation  $G^t$ ,  $t = 0, \dots, T$  defines the snapshot graph over time. Our main idea is to identify a set of leading nodes, denoted as  $\mathbb{NL}^t$ ;  $t = 1, \dots, T$ , that maximizes the influence propagation in each of the snapshot graph  $G^t$ . Each user is represented by an attributes vector  $X_i = (x_{i1}, \dots, x_{ij})$ , where  $x_{ij}$  is the value taken by the attribute  $j$  of the vertex  $v_i$ . In SND this value is binary, either 1 (if the user likes a center of interest) otherwise 0. This approach exploits, on the one hand, the relations between the vertices of the network and, on the other hand, the attributes that characterize them. Table 1 lists the notations to be used extensively in the sequel of this paper.

**Table 1.** Notation explanation

Notations	Descriptions
$G^t = (V, E^t)$	The snapshot graph
$V$	The nodes in $G^t$
$E^t$	The edges in $G^t$
$\Delta t$	The distance between two consecutive snapshots
$b$	The number of $E^t$ in $G^t$
$C_i$	The community in $G^t$
$M$	Number of communities
$NL^t$	The leaders nodes in $G^t$
$NA^t$	The active nodes in $G^t$
$N_{inf}^t$	The influential nodes in $G^t$
$A_v$	The active nodes

## 4 Proposed Algorithm

The main objective of SNDUpdate is to detect the most influential nodes in a dynamic social network. It exploits the structural and semantic aspects of the network. For this reason, the main idea is to propose an approach that contains two phases. Indeed, the first phase of SNDUpdate explores the structural aspect of the network and the second phase concentrates on the semantic aspect. In the sequel of this paper, each user is described by a set of interests that are represented as an attributes vector. In the previous work, the influential nodes are detected in a static social network and thus the weight of the link between two nodes remains unchangeable. In this paper, the network is dynamic so the structure of the network changes and thus the link between two nodes belonging to a snapshot graph  $G^t$  can be removed in the snapshot graph  $G^{t+1}$  which generates a modification in the value of the semantic similarity between two nodes as well as a modification in the set of leader nodes.

### 4.1 Phase 1: Community Detection

In this phase, we propose to use Combo [11]. This algorithm handles the community detection problem which is able to deal with various objective functions. The majority of search strategies take one of the next steps to improve the quality of partition: merging two communities, splitting a community into two and moving nodes between two distinct communities. Combo covers all these possibilities.

### 4.2 Phase 2: Influential Nodes Detection

The first part of the phase 2 enables to generate a set of leader nodes. A node is a leader if its degree centrality is greater or equal then its neighbors. The degree centrality enables to measure the total of the node connections with its neighbors. In this paper, we model the dynamic social network as a group of snapshot graphs while the edges in each snapshot graph change over different time intervals and thus the leader nodes change in each snapshot graph. This equation is used to calculate the degree centrality of a node  $v$  of a given snapshot graph  $G^t = (V, E^t)$ :

$$dc(v) = \frac{deg(v)}{|V| - 1} \quad (1)$$

In the second part of this phase, we use the diffusion model in [2] to define the inactive nodes that can be activated. In  $G^t$ , the link between two nodes is associated with a weight which is defined by the semantic similarity of their information. The semantic similarity related to each snapshot graph  $G^t$  is given by following equation:

$$sim_{u,v}^t = \frac{Common(u,v)}{long(u) + long(v)} \quad (2)$$

Every node  $v$  in a partition  $\mathcal{P} = \{C_1, \dots, C_r\}$  is related to a degree of centrality. Taking into consideration the degree of centrality of each node in  $G^t$  and a set of active nodes, a node  $v$  becomes active if the total of the similarity or the total weight of its active neighbors overrides the threshold value  $dc(v)$  associated with a node  $v$ . Our diffusion model is defined as follows:

$$\sum_{u \in \mathbb{A}_v} w_{v,u}^t > dc(v) \quad (3)$$

When we apply our diffusion model in each snapshot  $G^t$ , we can get a set of active nodes  $\mathbb{NA}^t$  which allows us to identify the set of influential nodes. Our main objective is to identify from the active nodes those that maximize the influence propagation. In each community, we determine its influence degree which is given by the Eq. 4 and is equal to the number of active nodes in each community ( $V_{active}$ ) divided by the sum of nodes in the graph  $G(V)$ .

$$R(C_r) = V_{active}/V \quad (4)$$

Firstly, the proposed approach determines influential nodes from the community that contains the value of the highest influence degree. Secondly, our objective is to calculate for each active node its closeness centrality given by the Eq. 5. It represents the total of the length of the shortest paths between the node and all other nodes in the graph. The closeness centrality is defined by the following equation:

$$CCenter(v_i) = \frac{1}{\sum_{v_j \in V} |ShortPath(v_i, v_j)|} \quad (5)$$

Once we have calculated the closeness centrality of every leader node, we classify these nodes according to an increasing order. The influential node is that admits the highest closeness centrality. The algorithm of SNDUpdate is described in Algorithm 1. Firstly, it detects communities using Combo algorithm. Secondly, on each snapshot graph  $G^t$ , it generates a set of nodes playing the role of a leading nodes. Once we have generated the set of the leader nodes that are the initiators, we apply our diffusion model. Finally, we apply our diffusion model to determine the set of active nodes and then identify the influential nodes. In the proposed algorithm we find the set of the active nodes that maximizes the influence within each snapshot graph  $G^t$  with updating  $b$  (Update b) the function which allow us to update the value of edges over each time stamp  $t$ . About the SNDUpdate time complexity, because it is difficult to analyse the complexity of dynamic edges update operations [16], we will only analyze each step in a static snapshot. SNDUpdate consists of T snapshots consisting each one of two phases. The first phase admits a complexity of  $O(V^2 \log M)$  as we used the Combo algorithm. The second phase is composed of two steps. The first generates leader nodes and admits a complexity of  $O(MV^2)$ . The second calculates the temporary shortest path of active nodes and corresponds to a complexity of  $O(E^t(NA^t) \log(NA^t))$ .

**Algorithm 1.** SNDUpdate

---

**Data:** An initial snapshot graph  $G^t = (V, E^t)$ ,  $T$ ,  $b$ ,  $\Delta t$ : the distance between two consecutive snapshots;

**Result:** A set  $N_{inf}^t$  of influential nodes at  $t = 1, \dots, T$ ;

```

1   $N_{inf}^t \leftarrow \emptyset$ ;
2   $NL^t \leftarrow \emptyset$ ;
3   $NA^t \leftarrow \emptyset$ ;
4  Begin
5      for  $t=0$  to  $T$  do
6          for  $b$  edges in  $G^t$  do
7              Apply Combo to detect communities;
8              foreach community  $C$  in  $G^t$  do
9                  Determine the degree centrality of each node in  $G$  applying
10                     the equation 1 ;
11                     foreach node  $v \in V$  of a community  $C$  do
12                         if  $isLeader(v)$  then
13                              $NL^t \leftarrow NL^t \cup \{v\}$ ;
14                         end
15                     end
16                     foreach Node  $u$  in  $C$  do
17                         Allocate a weight for each node that represents the degree
18                         of centrality;
19                     end
20                     foreach Edge  $(u, v)$  in  $C$  do
21                         Determine the similarity  $sim(u, v)$  using equation 2;
22                          $w_{u,v}^t \leftarrow sim_{u,v}^t$ ;
23                         if  $(\sum_{u \in A_v} w_{v,u}^t > dc(v))$  then
24                              $NA^t \leftarrow NA^t \cup \{u\}$ ;
25                         end
26                     end
27                      $M = |C|$ ;
28                     while  $M \neq \emptyset$  do
29                         Determine the degree of influence of each community applying
30                         the equation 4;
31                          $A_{C_{max}} \leftarrow \{active\ nodes\ in\ C_{max}\}$ ;
32                         while  $(C_{max} \neq \emptyset)$  et  $(A_{C_{max}} \neq \emptyset)$  do
33                             Calculate the closeness centrality of each node in  $A_{C_{max}}$ 
34                             using equation 5;
35                             Classify these nodes in an ascending order of closeness
36                             centrality;
37                              $v_{max} \leftarrow$  the node having the highest closeness centrality;
38                              $N_{inf}^t \leftarrow N_{inf}^t \cup \{v_{max}\}$ ;
39                         end
40                     end
41                     return  $N_{inf}^t$ ;
42 end
43  $t \leftarrow t + \Delta t$ ;
44 Update  $b$ ;
45 End

```

---

## 5 Experiments

In this section, we estimate the effectiveness and efficiency of our proposed SNDUpdate to identify the influential nodes in dynamic social networks on three real networks. The experimental results demonstrate that our algorithm is both efficient and effective.

### 5.1 Experiment Settings

In our experiments, we use three real social networks as shown in Table 2. For each data set, we used 3 different social networks, updated edges, and thus executed the experiments 3 times. For the 3 examples, however the original networks and updated edges are different, the combination of the 3 groups snapshots is similar to the data set itself. Let  $b$  denote the number of edges in each snapshot graph, using different parameters,  $b$  and  $\Delta t$ , we can create a family of snapshots with many properties for our next experiments. In Table 2, to simulate dynamic networks, we partitioned all edges into 3 time stamps: 25% of edges in time stamp  $t = 5$ , 50% of edges in time stamp  $t = 10$  and 85% of edges in time stamp  $t = 15$ . The evolution of edges can reflect that the real-world social networks change rapidly during the considered time periods.

**Table 2.** Datasets and edge information

Networks	Nodes	Edges	Edge information		
Flixster	99,825	978,265	<b>t = 5</b>	<b>t = 10</b>	<b>t = 15</b>
			1,222,831	1,467,398	1,809,790
NetHept	15,634	62,836	78,545	94,254	116,247
CA-HEP-PH	12,008	237,010	296,263	355,515	438,469

Table 2 shows the number of edges in each snapshot graph created from networks. We make a group of snapshot graphs from three networks by changing the number of edges with a constant time difference  $\Delta t = 5$  min. The metrics applied in this paper to evaluate the performance of our algorithm are: influence propagation and running time. Influence propagation is the total number of influenced nodes activated by leader nodes and the running time is the time to identify the most influential nodes.

### 5.2 Experiment Results

In this paper, we compare our algorithm with two static influence maximization algorithm IMM, Degree and one dynamic algorithm DIM [19]. As shown in Figs. 1 and 2, the influence propagation of SNDUpdate algorithm outperforms that of IMM and DIM algorithms on Flixster and NetHept datasets. Varying



the network size, Degree algorithm can not detect influential nodes anymore, while SNDUpdate finds better values of the influence propagation than those of both IMM and DIM. According to the results of this evaluation, while increasing the size of the network and updating the number of edges at each time stamp, SNDUpdate is still able to have a better influence propagation. Thus, it covers a large number of influential nodes in any type of networks. As conclusion, on the Flixster and NetHept networks, SNDUpdate is more efficient since the diffusion strategy based on edges change can narrow the search space of influential nodes.

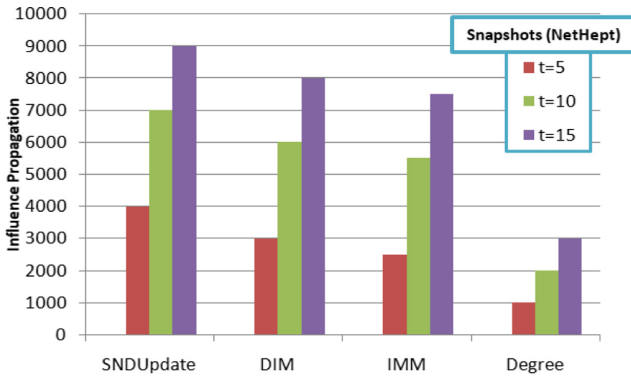


Fig. 1. Detecting influential nodes under NetHept network

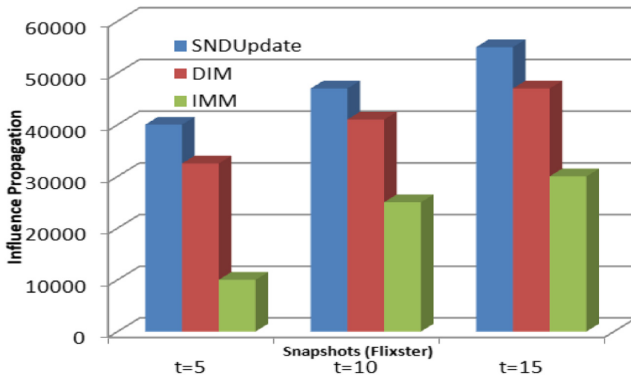


Fig. 2. Detecting influential nodes under Flixster network

### 5.3 Comparison with Methods Based on Metrics

We compared our algorithm SNDUpdate with three algorithms based on metrics which are: K-shell, MDD [17] and CN [12]. Our study is based on metrics to detect influential nodes in dynamic social networks. In the K-shell method, some nodes with large degree are not under consideration. When examining the exhausted degree in the network decomposition, the ranking of these nodes are ameliorated by the MDD algorithm. That is the reason why MDD method outperforms the K-shell method. In spite of the fact that the MDD method can improve the K-shell method in running time, it is still not the best way to deal with the problem of identifying influential nodes. According to this evaluation, when updating the size of the network SNDUpdate still has a better running time than the three other algorithms K-shell, CN and MDD. We explain this result by the fact that our method covers a considerable number of influential nodes in any type of networks. As indicated in Table 3, SNDUpdate is the best one regarding to the running time.

**Table 3.** Evaluation of the running time in different algorithms

Networks	K-shell	CN	MDD	SNDUpdate
CA-HEP-PH	2037	3003	8772	100
NetHept	4048	5929	16,195	120
Flixster	8110	10,269	32,442	1000

### 5.4 Comparison with Methods Based on a Diffusion Model

To detect influential nodes, all the algorithms are running under a diffusion model: Independent Cascade model (IC), Linear Threshold model (LT) and the Weighted Cascade (WC). Our diffusion model is based on the semantic similarity between nodes. We evaluate the running time of SNDUpdate, DIM and IMM algorithms on NetHept and Flixster networks in three snapshots. As illustrated in Figs. 3 and 4, the running time of SNDUpdate is better than the other two algorithms on the two networks. Among the three compared algorithms, we can easily find that IMM is very slow, while SNDUpdate performs well in terms of running time on the two networks. The reason is that IMM is a static algorithm so, when the network changes, DIM and SNDUpdate which are two dynamic algorithms run faster than the static algorithm IMM on the two networks. On Flixster network, the SNDUpdate running time increases, but is still better than the running time of DIM algorithm on large dynamic network. On NetHept and Flixster networks, SNDUpdate is faster than DIM algorithm. As conclusion, we can see that SNDUpdate has the best running times. The reason is that we use a diffusion model that improves the influence propagation as well as the running time of SNDUpdate. Note that IMM and DIM cannot detect influential nodes behind any diffusion model because they do not know the influence propagation of each node.

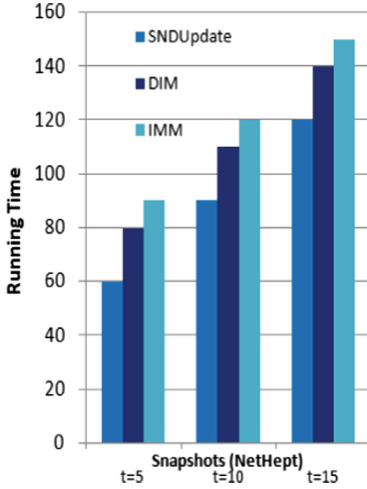


Fig. 3. Running time on NetHEPT

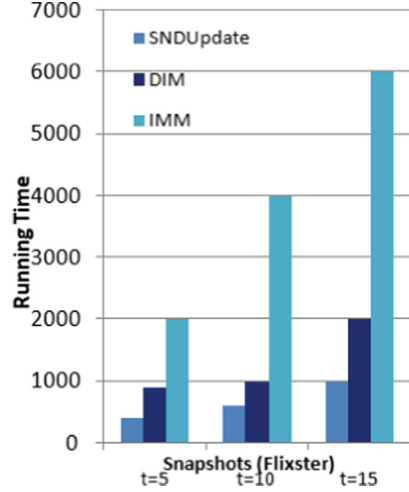


Fig. 4. Running time on Flixster

## 6 Conclusion

In this paper, we study the problem of identifying influential nodes in a changing social networks. We propose an approach called SNDUpdate algorithm and describe how to integrate it into a classical SND algorithm. The goal of this method is to detect influential nodes in dynamic social networks where edges are evolving over time. To prove the performance of the proposed approach, we compare it with approaches based on metrics and diffusion model in three real networks. Experimental studies carried out on the selected networks shown how it is possible to obtain good results in the detection of nodes that maximize the influence propagation. Empirical studies show that our algorithm has great improvement on the influence propagation compared with IMM, Degree and DIM algorithms. In the future, we plan to predict the change of influential nodes where both nodes and edges evolve in dynamic social networks.

## References

1. Kitsak, M., Gallos, L., Havlin, S.: Identification of influential spreaders in complex networks. *Nature Phys.* **6**, 888–893 (2010)
2. Hafiene, N., Karoui, W.: A new structural and semantic approach for identifying influential nodes in social networks. In: *IEEE/ACS International Conference of Computer Systems and Applications AICCSA*, pp. 1338–1345 (2017)
3. Chen, X., Song, G., He, X., Xie, K.: On influential nodes tracking in dynamic social networks. *IEEE Trans. Knowl. Data Eng.* **29**, 359–372 (2015)
4. Liu, X., et al.: On the shoulders of giants: incremental influence maximization in evolving social networks. *Complexity* 1–14 (2017)

5. Wang, T., Dai, W., Jiao, P., Wang, W.: Identifying influential nodes in dynamic social networks based on degree-corrected stochastic block model. *Int. J. Mod. Phys. B* **30**(16), 1–18 (2016)
6. Basaras, P., Katsaros, D., Tassiulas, L.: Detecting influential spreaders in complex, dynamic networks. *Computer* **46**, 24–29 (2013)
7. Aggarwal, C.C., Lin, S., Yu, P.S.: On influential node discovery in dynamic social networks. In: *International Conference on Data Mining*, pp. 636–647 (2012)
8. Chen, W., Lu, W., Zhang, N.: Time-critical influence maximization in social networks with time-delayed diffusion process. In: *International Conference on Data Mining*, pp. 636–647 (2012)
9. Yang, Y., Wang, Z., Pei, J., Chen, E.: Tracking influential nodes in dynamic networks. *IEEE Trans. Knowl. Data Eng.* **29**, 2615–2628 (2017)
10. Yang, Y., Wang, Z., Jin, T., Pei, J., Chen, E.: Tracking top-k influential vertices in dynamic networks. *IEEE Trans. Knowl. Data Eng.* **29**, 1–14 (2018)
11. Sobolevsky, S., Ratti, C., Campari, R.: General optimization technique for high-quality community detection in complex networks. *Phys. Rev.* **90**, 1–19 (2014)
12. Feng, S., Wang, L., Sun, S., Xia, C.: Synchronization properties of interconnected network based on the vital node. *Non Linear Dyn.* **93**(2), 335–347 (2018)
13. Tong, G., Weili, W., Tang, S., Du, D.-Z.: Adaptive influence maximization in dynamic social networks. *IEEE/ACM Trans. Netw.* **25**(1), 112–125 (2017)
14. Ren, J., Wang, C., Liu, Q., Wang, G., Dong, J.: Identify influential spreaders in complex networks based on potential edge weights. *Int. J. Innov. Comput. Inf. Control* **12**(2), 581–590 (2016)
15. Wei, W., Carley, K.: Measuring temporal patterns in dynamic social networks. *J. ACM Trans. Knowl. Discov. Data* **10**(1), 1–27 (2015)
16. Ohsaka, N., Akiba, T., Yoshida, Y., Kawarabayashi, K.: Dynamic influence analysis in evolving networks. *J. Proc. VLDB Endow. VLDB* **9**(12), 1077–1088 (2016)
17. Zeng, A., Zhang, C.-J.: Ranking spreaders by decomposing complex networks. *Phys. Lett.* **377**, 1031–1035 (2013)
18. Zhuang, H., Sun, Y., Tang, J., Zhang, J., Sun, X.: Influence maximization in dynamic social networks. In: *International Conference on Data Mining*, pp. 636–647 (2013)
19. Wang, Y., Zhu, J., Ming, Q.: Incremental influence maximization for dynamic social networks. In: Zou, B., Han, Q., Sun, G., Jing, W., Peng, X., Lu, Z. (eds.) *ICPCSEE 2017. CCIS*, vol. 728, pp. 13–27. Springer, Singapore (2017). [https://doi.org/10.1007/978-981-10-6388-6\\_2](https://doi.org/10.1007/978-981-10-6388-6_2)