An Influence Maximization Method Based on Effective K-Core for Social Networks

Yang Wenqi[1]

[1]*School of Computer Science and Technology, China University of Mining and Technology, China*

[1]*yangwenqi@cumt.edu.cn*

Abstract—
Neighbours usually play an important role in the measurements of node influence. The number of neigbhours to a node is called its degree, which is a frequently adopted centrality. In order to solve the problem that traditional degree-based influence maximization algorithms fail to identify effective neighbours, this paper proposes a K-core based social network influence maximization method named K-core algorithm (EKCA). The proposed method first introduces the concept of K-core. Then it calculates the core of nodes based on K-core decomposition. Last, it uses coreness instead of degree as a standard to select effective neighbours. The proposed method could describe the position of nodes in the network more accurately, and thus better for the influence maximization problem. Experiments on networks with various sizes show that the proposed method can select nodes that spread more influence than the degree-based influence maximization algorithms.

Keywords—*social network, influence maximization, independent cascade model, K-core, effective neighbour*

I. INTRODUCTION

With the developments of social network services, more and more users tend to exchange their opinions and information on the Internet. Consequently, the research on the spread of social influence has become a hot spot in the field of social network analysis. Online social networks bring various possibilities for marketing such as using social networks for viral marketing [1] which often brings a huge impact at a small cost. Maximizing the spread of influence is the key to solving this type of problems [2]. It can be widely used in marketing strategies, targeted advertising, public opinion prediction and controlling, etc [3].

Influence maximization is a hot issue in the field of social network analysis [4]. Researchers have proposed many algorithms to solve the problem of influence maximization, the most among which are propagation-based algorithms and topology-based algorithms. The algorithm based on propagation has a good effect but suffers from high time complexity and thus cannot handle large networks. However, many traditional algorithms based on topological structure use degree to measure the effectiveness of a node's neighbours, which will inevitably select some invalid neighbours and lead to the misjudgment for node influence.

To solve the limitation of the commonly used degree centrality which leads to the problem of selecting seed nodes with invalid neighbours, this paper proposes an effective K-core algorithm (EKCA) that exploits K-core instead of the degree for the influence maximization problem. Moreover, the proposed method evaluates a node's influence by calculating the effective K-core in its neighbourhood. Experimental results show that the proposed method has a higher influence spread than other degree-based algorithms.

II. RELATED WORKS

Kempe et al. [5] first proved that influence maximization is an NP-hard problem, and proposed a general greedy algorithm based on propagation for hill-climbing greedy. The algorithm has good accuracy that is guaranteed to be about 63% of the optimal solution. However, since selecting the node that brings the maximal marginal influence needs to traverse the entire network, the efficiency of the algorithm could be rather poor on networks with a large scale. The optimized greedy algorithm CELF (Cost-Effective LazyForward) [6] solves the problem of low efficiency. The algorithm exploits the submodularity of the function to reduce the time for evaluating a node's influence. The experimental results show that CELF could select nodes with an improvement in the speed of nearly 700 times. However, CELF still takes several hours to run a network with tens of thousands of nodes. The StaticGreedy algorithm proposed by Cheng et al. [7] optimizes the running time by saving static snapshots for calculating the marginal gain. But there is still a problem of long running time in large-scale networks. The above are all propagation-based algorithms. It can be seen that this kind of algorithm needs to optimize the influence spreading process.

Another kind of algorithm is based on the network topology, which focuses on the topology-related centralities of the network. This kind of algorithm does not need to consider the optimization of the influence spreading process, thus the running time is very short. However, the influence spread of these algorithms is relatively small, and they are unstable on different network structures. In the early research, centralities such as degree and distance [8] were used as evaluation parameters. The DegreeDiscount algorithm [9] modifies the degree and improves the effect of the degree index, while the improvement is limited. Liu et al. [10] proposed the LIR algorithm by finding the node with the local maximal degree. However, If the network is relatively smooth, the nodes selected by this algorithm may be on the border, which seriously reduce the influence spread. Nguyen et al. [11] rescreened the selected first batch of seed nodes to filter out non-adjacent nodes as the final seed nodes, and proposed the probability-based multi-hop diffusion algorithm (hereinafter referred to as the pBmH algorithm). The process of rescreening nodes consumes extra time, and if the nodes that meet the requirements can be directly selected, the time complexity will be greatly reduced.

The above two types of algorithms have their different advantages and disadvantages. The algorithms based on propagation can ensure their accuracy, while they are inefficient and not suitable for large-scale networks. The algorithms based on topology have higher efficiency, while the spread of influence is not as good as the algorithms based on propagation. Moreover, their performance is unstable on different network structures [12].

Avoiding the Rich-club phenomenon is an effective way to improve the performance of influence maximization algorithms based on topology. The Rich-club phenomenon is that the selected nodes have a large number of common neighbours during the propagation process, which will severely limit the spread of influence. For example, in the degree-based influence maximization methods, the selected nodes with a larger degree have many common neighbours, which can only affect a limited number of nodes in the propagation process. Although the LIR algorithm solves the Rich-club phenomenon, it has limited influence in some networks. The pBmH algorithm needs to re-screen the selected seed nodes and filter out the nodes that are connected with selected seed nodes.

When selecting seed nodes, if replacing the degree with a more effective parameter K-core as well as considering the effectiveness of different neighbours, the Rich-club phenomenon could be effectively avoided. Meanwhile, nodes with greater influence could also be selected with high efficiency. Therefore, this paper will handle the influence maximization problem from the perspective of the K-core.

III. **SPREADING MODEL**

For a specific social network, finding the set of influential nodes is highly relevant to the propagation model. A social network is usually denoted as a graph *G(V, E)* composed of *n* nodes and *m* edges, where nodes represent social individuals and edges represent the social relationships between individuals. Independent Cascade Model and Linear Threshold Model are two frequently adopted propagation models.

In these two models, nodes have two states, i.e. active and inactive. A node can turn from the inactive state to the active state, but not vice versa. As the number of active neighbours of an inactive node increases, the node tends to be active.

A. Independent Cascade (IC) Model

The independent cascade model is a probabilistic model. For each node in the network *G*, there are two states: active and inactive. Each node can only change from the inactive state to the activated state, and each node can be activated by its neighbours. For each edge $e(u,v) \in E$, the specific influence probability $p(u,v) \in (0,1]$ means that *u* influences *v* through the edge *e(u,v)* with the probability of *p(u,v)*. Given the initial set of activated nodes *S0*, the propagation process proceeds in the following way. When the propagation reaches the *t+1* step, the nodes activated in the *t* step (*St*) try to activate their neighbours according to the probability *p(u,v)*. The activated nodes in this step would be added to *St+1*. This process repeats until no new nodes are activated. In the whole process, a node could only try to activate each of its neighbours once. If the activation is successful, its neighbour will change from inactive to active, and if the activation fails, no more attempts will be made.

B. Linear Threshold (LT) Model

Under the linear threshold model, each node *v* contains an activation threshold θv that is randomly selected uniformly from the interval *[0,1]*. In addition, LT model stipulates that the sum of all incoming weights is at most 1, and the influence of other incoming nodes on it is thus cumulative. When the influence exceeds the threshold, the node is activated. The linear threshold model reflects the cumulative effect of influence.

IV. **ALGORITHM DESCRIPTION**

A. K-core

The K-core concept was proposed by Seidman [13] in the paper "Network Structure and Minimum Degree" in 1983. It can be used to describe network characteristics that cannot be captured by degree. The K-core reveals the structural properties and hierarchical properties derived from the special topology structure.

Given a network *G(V, E)* where *V* is the set of nodes and *E* is the set of edges, the relevant definitions are as follows.

Definition 1. K-core. The degree of any node *v* in the set *C⊆V* is not less than *k*, and the derived largest subgraph *Gc (C, E|C)* is called K-core. In other words, the subgraph obtained by recursively removing nodes with a degree less than k in the graph as well as their connecting edges is the K-core of the graph.

Definition 2. Coreness. If node *v* belongs to *k*-core but not *(k+1)*-core, then the coreness of node *v* is *k*.

An important feature of K-core is its connectivity. If the K-core of the graph is k-connected, then there are k non-cross paths between any two nodes. Therefore, a larger coreness indicates better connectivity of the nodes.

Algorithm 1. K-core decomposition algorithm

```
Input: Network G=(V, E), number of seed nodes k
Output: the number of cores of all nodes in the network Cv
function KS = kcore(A)
while length(DeletedNodes) ≠ n
newD = U * TempA
    [~,tmp2] = find(newD <= ks)
    nextDel = setdiff(tmp2, DeletedNodes)
    if isempty(nextDel)
            innerIndex = 0
            CurDeletedNodes = []
            Ks = ks + 1
    else
            TempA(nextDel, :) = 0
            TempA(:, nextDel) = 0
            DeletedNodes = [DeletedNodes, nextDel]
            CurDeletedNodes = [CurDeletedNodes, nextDel]
            innerIndex = innerIndex + 1
            inner(1, nextDel) = innerIndex
            KS(1, nextDel) = ks
    end if
    inner(2, CurDeletedNodes) = innerIndex
end while
end function
```

The following illustration shows the execution process of the K-core decomposition through a simple network. The algorithm first starts the 1-core decomposition which selects all nodes with their degree less than 1 in the network. Then, these nodes will be deleted. The process keeps repeating until the degree of all remaining nodes is at least 1. The coreness of all these deleted nodes is 0 and thus the remaing graph is a 1-core subgraph. When the minimal degree in the graph is 1, the algorithm will begins 2-core decomposition which selects the nodes whose degree is less than 2 and keeps deleting them. The coreness of these deleted nodes will be 1 and leaving the 2-core subgraph. By analogy, the algorithm keeps performing 3-core and 4-core decomposition on the graph in Figure 1. When all the nodes are deleted, the algorithm terminates and all the nodes have their corresponding coreness, just as shown in Figure 1.

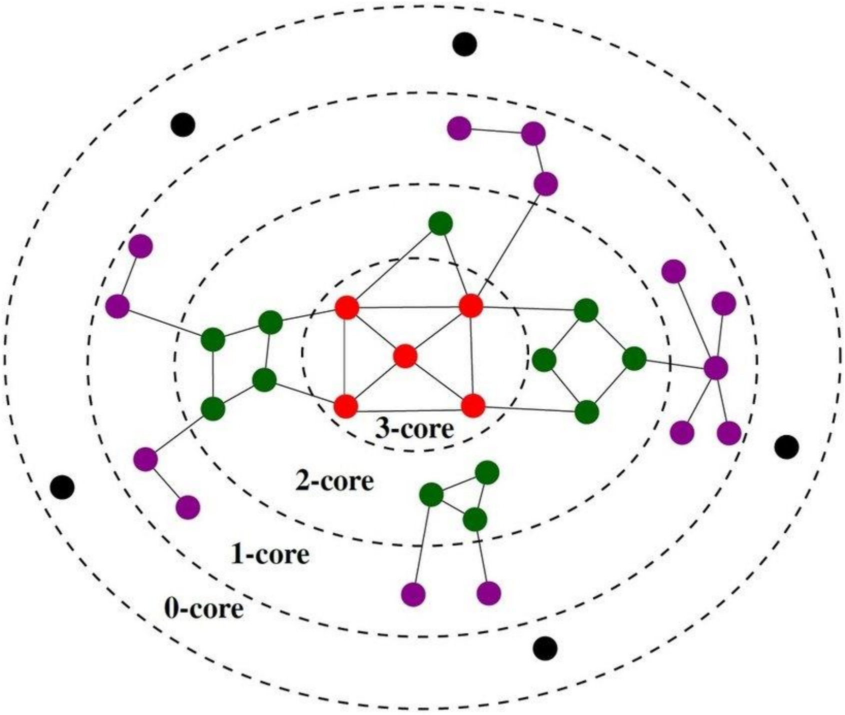


Fig. 1 Illustration of the K-core decomposition. Here, k max = 3.

B. EKCA Algorithm

After K-core decomposition of the network, appropriate parameters need to be selected as a measurement for node influence. The Effective K-core Algorithm (EKCA) is thus proposed to filter effective neighbours by comparing their coreness. The effective K-core of a node can be calculated by the following formula:

$$ekc(v) = \sum_{u \in \Gamma(v)} \frac{1}{C_v - C_u},$$

where C_v denotes the coreness of v and $\Gamma(v)$ denotes the set of v's neighbours.

The process of the EKCA algorithm is shown in the following Algorithm 2.

Algorithm 2. EKCA

```
Input: Network G=(V, E), number of seed nodes k
Output: Seed node set S
Initialize S = Ø
for i = 1:Node
temp = find(A(i,:))
    for j = 1:Degree(i)
        if nor_core(temp(j)) <= nor_core(i)
            if nor_core(temp(j)) == nor_core(i)
                sp_Degree(i) = sp_Degree(i) + 2
            else
                sp_Degree(i) = sp_Degree(i) + 1/(nor_core(i)-nor_core(temp(j)))
            end if
        end if
    end for
end for
[rank, location] = sort(sp_Degree,'descend')
seed_spDegree = location
for i = 1:k
    m = find(sp_DegreeA == max(sp_DegreeA))
    seed_spDegreeND(i) = m(1)
    temp = find(A(seed_spDegreeN D(i),:))
```

```
        sp_DegreeA(temp) = 0
        sp_DegreeA(seed_spDegreeND(i)) = 0
    end if
    return S
```

The coreness of a node can describe its position in the network topology more accurately than the degree. The idea of the proposed EKCA is that the nodes with high influence are generally considered to have a large number of neighbours. Besides, a large influence of neighbours could also benefit their centered node. In EKCA, the influence of a node *v* is measured by making a difference on coreness between *v* and its neighbours whose coreness is small than *v*. Since a large coreness of v's neighbour benefits *v* better, the reciprocal of the differences is used as their weight. Specially, if the coreness of *v*'s neighbour is equal to that of *v*, their weight will be set to 2. Finally, the effective K-core of *v* (*ekc(v)*) is the sum of the weights of its neighbours. After the effective K-core of all nodes is calculated, the algorithm iteratively selects seed nodes based on these values. When selecting seed nodes, this paper adopts the Neighbour Deleting (ND) strategy which deleting all neighbours of a node when it is selected as a seed. As a consequence, there will be no edge between the final seed nodes, and thus the Rich-club phenomenon is avoided.

V. **EXPERIMENT AND ANALYSIS**

A. Experiment description

Five networks are used in the experiment and their information is shown in Table 1.

TABLE I
EXPERIMENTAL DATA SET

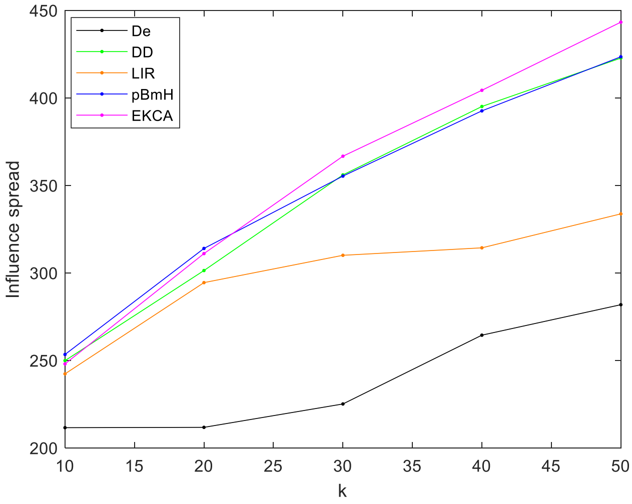
Network	**Nodes**	**Edges**	**Average degree**	**Pc**
NetScience	1461	2742	3.451	0.323
Power	4941	6594	2.669	0.437
CaGrQc	5242	14496	5.531	0.091
CaHepTh	9877	25998	5.264	0.072
PGP	10680	24316	4.554	0.048

NetScience [15] (Network Science) is a network of co-authorships in the area of network science. Power [16] contains information about the power grid of the Western States of the United States of America. CaGrQc [17] (General Relativity and Quantum Cosmology) is a collaboration network of Arxiv General Relativity. CaHepTh [17] (High Energy Physics-Theory) is a collaboration network of Arxiv High Energy Physics Theory. PGP [18] (Pretty Good Privacy) is the interaction network of users of the Pretty Good Privacy (PGP) algorithm.

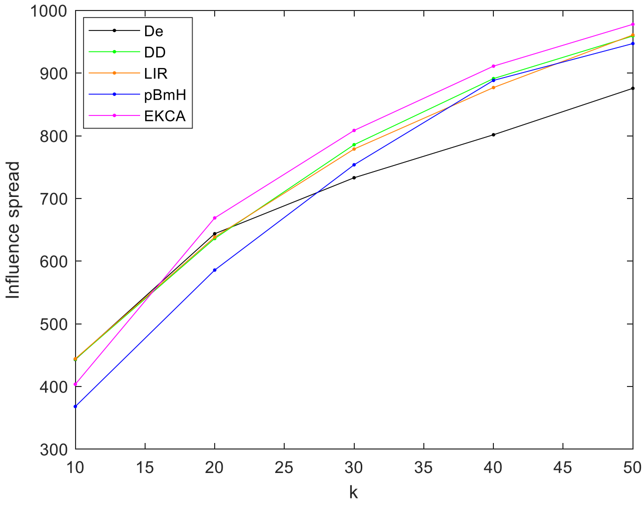
The *Pc* in Table 1 is the network spreading threshold of the Independent Cascade (IC) Model. Literature [14] shows that different networks usually have different appropriate propagation probabilities due to their different structures. If the propagation probability *p* is too small ($p<Pc$), the influence spread of nodes in the network is very limited. If the propagation probability is too large ($p>Pc$), choosing 1 node or 50 nodes would barely make any difference in the final results. Therefore, under the IC model, an appropriate propagation probability should be selected for experiments. In this experiment, the network spreading threshold is selected as the propagation probability.

B. Experimental results and analysis

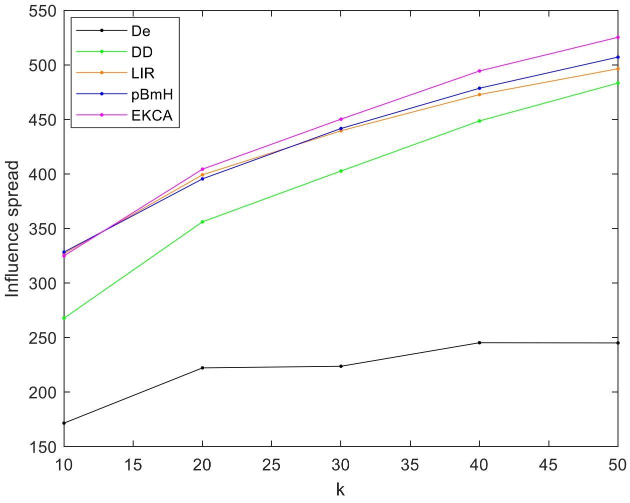
This section compares the proposed EKCA algorithm with 4 baseline algorithms, including Degree, DegreeDiscount, LIR, and pBmH (Probability-based Multi-hop). Figure 2 shows the influence spread of 5 algorithms with the increasing size of seed nodes.



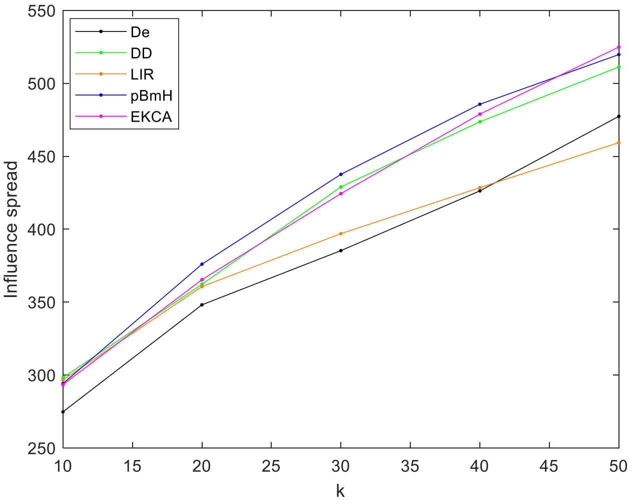
(a)NetScience



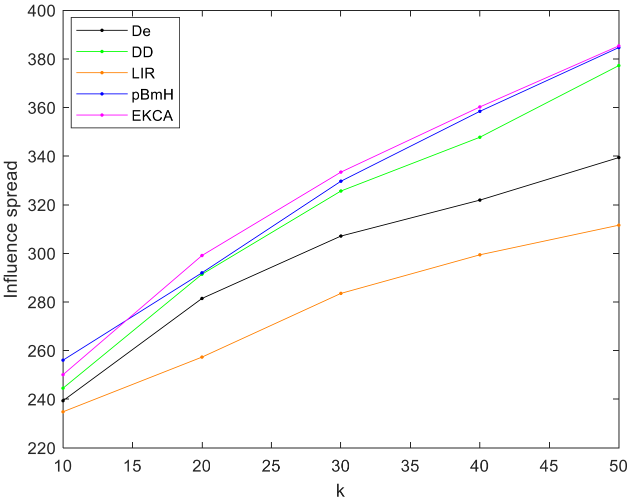
(b)Power



(c)CaGrQc



(d)CaHepTh



(e) PGP

Fig. 2 The results of different algorithms on different networks

Figure 2 (a) shows the results on the NetScience network. EKCA has the largest spread of influence. The influence spread of LIR and Degree is smaller and the influence spread of pBmH and DegreeDiscount is relatively close to that of EKCA. When the size of the seed set is small, the influence spread of EKCA is slightly lower than pBmH and DegreeDiscount. However, with the increasing of the seed size, the influence spread of EKCA gradually widens the gap with pBmH and DegreeDiscount. Figure 2(b) shows the result on the Power network. The influence spread of EKCA is also better than other algorithms. As the seed size increases, the influence spread of Degree becomes worse. Figure 2(c) shows the result on the CaGrQc network. The influence spread of EKCA is obviously better than pBmH, LIR, and DegreeDiscount. The Degree algorithm is rather poor on this network. Figure 2(d) shows the result on the CaHepTh network. The influence spread of EKCA is slightly lower than pBmH, close to DegreeDiscount, and higher than LIR and Degree. Figure 2(e) shows the results on the PGP network. EKCA gives the largest influence spread, with pBmH and DegreeDiscount tightly follows. The influence spread of LIR and Degree is comparatively lower.

In summary, The proposed EKCA in this paper has a good performance on networks with different sizes and structures. The influence spread of EKCA is almost the largest in all testing networks. The performance of LIR algorithm is unstable in different networks. The influence spread EKCA is better than two degree-based benchmarks, i.e. pBmH and DegreeDiscount, which shows the advantage of the K-core. As the number of seed nodes increases, the performance of EKCA becomes more prominent. The larger the seed size, the better the performance.

VI. CONCLUSION

To solve the influence maximization problem in social networks, this paper proposes an EKCA algorithm that shows a better spread of influence with high efficiency. In this paper, the proposed effective K-core is used as the centrality measurement for selecting seed nodes. In the effective K-core, only effective neighbours are involved to calculate the weights for their centered node. The sum of the reciprocal weights is thus used to evaluate a node's influence. The effective K-core indicates that effective neighbours with a higher coreness could bring more influence to their centered node, thus it could reflect the node influence more accurately than degree-based centralities. Moreover, a neighbour deleting strategy is adopted to select seed nodes while avoiding the Rich-club phenomenon. Experiments show that EKCA has a larger spread of influence with high efficiency on networks of different sizes. Especially, as the number of seed nodes increases, the superiority of EKCA becomes more and more prominent. The larger the seed size, the better the performance. The results show that the EKCA is both stable and robust, outperforming the existing degree-based benchmarks.

REFERENCES

[1] LIU Quan, ZHANG Ming. Domain based influence maximization and viral marketing [J].Journal of Chinese Information Processing,2017,31 (3):118-124.

[2] ZHENG Zhiyun, FU Yuan, LI Lun, et al. Influence algorithm based on center weighted link in social networks [J].Computer Engineering and Design,2017,38 (1):110-115.

[3] HU Min, SUN Xinran, HUANG Hongcheng. Edge-cover algorithm for influence maximization in social network [J].Journal of Frontiers of Computer Science and Technology,2017,11(5):720-731.

[4] Arora A, Galhotra S, Ranu S. Debunking the myths of influence maximization: An in-depth benchmarking study [C]//Proceedings of the ACM International Conference on Management of Data,2017:651-666.

[5] Kemple D, Kleinberg J, Tardos E. Maximizing the spread of influence through a social network [J]. Theory of Computing,2015,11 (4):105-147.

[6] Tong G, Wu W, Tang S, et al. Adaptive influence maximization in dynamic social networks [J]. IEEE/ ACM Transactions on Networking,2017,25 (1):112-125.

[7] Cheng S, Shen H, Huang J, et al. StaticGreedy: Solving the scalability-accuracy dilemma in influence maximization [C]//Proceedings of the 22nd ACM International Conference on Information & Knowledge Management,2013:509-518.

[8] Rabade R, Mishra N, Sharma S. Survey of influential user identification techniques in online social networks [M]. Cham: Springer International Publishing,2014:359-370.

[9] ZHANG Yingqing, LUO Ming, LI Xing. A review on the node influence and influence maximization of complex networks [J]. Journal of Modern Information,2017,37 (1):160-164.

[10] Liu D, Jing Y, Zhao J, et al. A fast and efficient algorithm for mining top-k nodes in complex networks [J]. SciRep,2017,7:43330.

[11] Nguyen DL, Nguyen TH, Do TH, et al. Probability-based multi-hop diffusion method for influence maximization in social networks [J]. Wireless Personal Communications, 2017, 93(4): 903-916.

[12] YANG Shuxin,LIU Chenghui,LU Jihua.Two stages of heuristic based algorithm for influence maximization in social network [J]. Journal of Chinese Computer Systems, 2017, 38(10): 2268-2274.

[13] Seidman S B. Network structure and minimum degree Social Networks,1983,5(3):269-287.

[14] Radicchi F, Castellano C. Fundamental difference between superblockers and superspreaders in networks [J]. Phys Rev E, 2017, 95(1-1): 012318.

[15] Mark E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E, 74(3), 2006.

[16] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. Nature, 393(1):440–442, 1998.

[17] Leskovec J, Kleinberg J and Faloutsos C. Graph Evolution: Densification and Shrinking Diameters. ACM Transactions on Knowledge Discovery from Data (ACM TKDD), 1(1), 2007.

[18] Marián Boguñá, Romualdo Pastor-Satorras, Albert Díaz-Guilera, and Alex Arenas. Models of social networks based on social distance attachment. Phys. Rev. E, 70(5):056122, 2004.