IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 28, 2019, accepted April 14, 2019, date of publication April 23, 2019, date of current version May 24, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2912628

PHG: A Three-Phase Algorithm for Influence Maximization Based on Community Structure

LIQING QIU, WEI JIA, JINFENG YU, XIN FAN, AND WENWEN GAO

Shandong Province Key Laboratory of Wisdom Mine Information Technology, College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266510, China

Corresponding author: Liqing Qiu (liqingqiu2005@126.com)

This paper was supported by the Nature Science Foundation of China under Grant 61502281 and Grant 71772107.

ABSTRACT The main purpose of influence maximization is to find a subset of key nodes that could maximize the spread of information under a certain diffusion model. In recent years, many studies have focused on the problem of influence maximization. However, these studies usually ignore the role of community structure which captures a significant effect on the process of influence propagation. To address above problem, we propose a novel hybrid algorithm PHG, which is a three-phase algorithm for influence maximization based on community structure. In our algorithm the influence propagation process is divided into three phases: 1) partition phase; 2) heuristic phase; and 3) greedy phase. Specifically, we first design an efficient algorithm CCSC that finds key nodes in each community to construct a candidate set by detecting community structure. Second, we find the most potential influence nodes from a candidate set by combing the influence weight of nodes and the community influence of nodes through the analysis of the community structure of the impact on nodes. Finally, we greedily select the nodes with maximization marginal gain from remaining a candidate set. The extensive experimental results on artificial and real-world social networks show that our algorithm obtains a better influence spread as well as an acceptable running time.

INDEX TERMS Community structure, heuristic phase, influence maximization, greedy phase, partition phase.

I. INTRODUCTION

In recent years, social networks have drawn much attention such as Facebook, Twitter and Google+, which serve as an important medium or platform for sharing their thoughts, news and any type of information to users. In social networks, some users can propagate information through some kinds of relation (friendship or co-authorship, etc.), which leads to an important application in viral marketing. For example, a company wants to market a new product to customers. It wishes to select a small number of influential users to adopt the product so as to create a large cascade of further adoptions through the word-of-mouth effect. In above example, the crucial problem is how to select influential users so that these users could maximize spread their influence on other users. The crucial problem known as influence maximization, is to find top-K influential nodes in social networks that maximizes the influence.

The associate editor coordinating the review of this manuscript and approving it for publication was Seyedali Mirjalili.

Influence maximization problem was first introduced by Domingos and Richardsom[1]. Kempt *et al.* [2] proved the influence maximization problem is NP-hard and gave two fundamental propagation models: independent cascade model and linear threshold model. Some efficient approximate algorithms have been proposed [2]–[4] based on above two models in order to maximize the influence spread of nodes. Although the greedy algorithms are efficient, they ignore the role of community structure in the process of influence propagation. In the real world, people tend to connect with other people via strong links to form a community. Information flows between these communities at high speed which results in most people within communities know the same message immediately. Therefore, some researches [5]–[7] are focusing on finding key nodes only inside communities. Despite these researches reduce time complexity of algorithms, the connectivity between communities has been ignored. Information flows among different communities, expanding the breadth of information dissemination. Hence, in order to identify influential nodes, more and more works

are devoted to finding an efficient method by considering community structure.

According to the above analysis, we propose a three-phase algorithm for influence maximization, called Partition-Heuristic-Greedy Algorithm(PHG), to model the based-community influence maximization under linear threshold model. In our algorithm, we firstly design an efficient algorithm for narrowing down the search space of the candidate seeds, called Community-based Candidate Set Construction Algorithm(CCSC), to construct a candidate set by detecting community structure. Then we find the most potential influence nodes from a candidate set by combining the influence weight of nodes and the community influence of nodes through the analysis of the community structure of the impact on nodes. And finally, we greedily select the nodes with maximization marginal gain from remaining a candidate set. Consequently, the proposed method is superior to some classical algorithms in influence spread as well as a reasonable running time.

The contributions of this paper are summarized as follows:

- In real social networks, nodes tend to cluster together by some link relations. Therefore, we propose PHG by utilizing the community structure information to solve the problem of influence maximization.
- We heuristically choose the seed nodes from a candidate set by combining the influence weight of nodes and the community influence of nodes instead of relying only on the influence weight. Moreover, it is noticed that the community influence of nodes can be measured by the degree of nodes, the number of communities that nodes connected directly and the size of the community which will be discussed in detail in section 3.
- We carry out our experiments using several artificial and real-world social networks to demonstrate that the performance of our algorithm is not only efficient but also has a more favorable influence spread compared with state-of-the-art algorithms.

The rest of this paper is organized as follows. In section 2 we introduce related work of influence maximization containing basic diffusion model, influence maximization algorithms and community detection. We describe our proposed algorithm in detail in section 3 and verify our algorithm in artificial and real-world social networks in section 4. Finally, in section 5 we give our conclusion of this paper.

II. RELATED WORK

In this section, we introduce the following three aspects: basic diffusion model, influence maximization algorithms and community detection.

A. BASIC DIFFUSION MODEL

The diffusion model is a fundamental rule that information can propagate on it in an appointed social network. Assume that each node only has two states in the network: active state or inactive state. If a node has already accepted a message, this node will be an active node. Otherwise, this node will be an inactive node. With the unceasingly conducting of the studies, IC model and LT model are popular and have been widely applied to the problem of the influence maximization. In this paper, we mainly use the LT model which is considered more superior than the IC model for modeling influences in social networks owing to taking the "influence accumulation" property into account [4], [8].

1) LINEAR THRESHOLD MODEL

In this model, each node is allocated a threshold $\theta(w) \in [0, 1]$, which represents that the difficulty of this node is affected. For each directed edge $(v, w) \in E$, there is an influence weight $b_{v,w}$ ($\sum_{v} b_{v,w} \leq 1$) which reflects the influence of an active node v on its inactive neighbor w. When a node w is being an inactive state, the node w can be activated only by the total influence weight of all active neighbors of the node w is at least θ_w (as shown in formula(1)). The diffusion process will stop when no further nodes can be activated.

$$\sum_{v \in neighbor(w)} b_{v,w} \geq \theta_w \tag{1}$$

That is to say, when an active node v attempts to activate its neighbor w unsuccessfully, the influence of an active node v on its inactive neighbor w will be accumulated instead of disappeared immediately like the IC model. This accumulated influence greatly increases the possibility of the inactive node w activated by other neighbors. Based on this "influence accumulation" property, in our algorithm, we find the most potential influence nodes in the heuristic phase. Although these nodes are not the most influential nodes, their potential influence will be accumulated which results in activating more nodes in the greedy phase.

Algorithm 1 Greedy Algorithm

Input: Network $G = (V, E)$,k
Output: seed set S
1: $S \leftarrow \Phi$
2: for $i = 1$ to k do
3: $\quad v = \arg\max_{u \in v \backslash S}(\delta(S \cup \{u\}) - \delta(S))$
4: $S = S \cup \{v\}$
5: Return S

B. INFLUENCE MAXIMIZATION ALGORITHMS

Given a social network graph $G = (V, E)$, a seed set S contains k influential active nodes. For each edge $(v, w) \in E$, finding a seed set $S \subseteq V$ such that the expected number of nodes influenced by S, $\delta(S)$, is maximized. However, this influence maximization problem is proved to be NP-hard under the basic diffusion model. Most of the proposed algorithms try to obtain approximate solutions. As a result, an effective greedy algorithm was proposed which can guarantee the influence spread is within $(1 - 1/e - \varepsilon)$ of the optimal influence solutions[2]. As a basic influence maximization algorithm, the greedy algorithm of finding the k influence largest active nodes detailed is given in algorithm 1.

The core idea of the greedy algorithm is that selecting a node with marginal gain increment maximization (as shown in line3) to join in a seed set S at each step until the size of seed set is k. Although the greedy algorithm obtains a better influence spread, it is proved to be time consuming for applying to a large-scale network [15], [23], [24].

In order to optimize the greedy algorithm running efficiency, a lot of improved greedy algorithms were proposed such as CELF++ [9], NewGreedy [10] and MixGreedy [10]. Although these algorithms run much faster than the simple greedy algorithm, they still can not handle large-scale networks owing to time consuming. Another way to reduce the time complexity is to find the influence maximization nodes with some heuristic strategies, such as degree [2], pagerank [3] and PMIA [11]. As a result, these algorithms seriously decrease the accuracy. To solve the imbalance problem between the efficiency and effectiveness of the above algorithms, Tian *et al.* [4] presented a hybrid algorithm, called HPG, combining the advantages of both heuristic and greedy algorithms under the LT model. Specially, the proposed algorithm first heuristically chooses half of the initial seeds with the biggest potential influence and then greedily chooses the other half initial seeds with the most influence.

Above algorithms further improve the influence spread and the running time of the existing algorithms from different perspectives. However, these algorithms ignore the importance of community structure during the information diffusion which was proved by some works[12], [13]. Thus, some recent studies [14]–[16], [27]–[31] began to use community structure to solve influence maximization. These studies firstly divide communities by classical community algorithms. And then they compute the influence of nodes in each community to approximate the influence of the whole network. Moreover, these algorithms assume that different communities are isolated. However, obviously, they ignore the flow of information among different communities which reduces the influence spread of algorithm obviously.

Following the above analyses, we attempt to find suitable ways, such as combining the community structure and the advantages of the heuristic and the greedy algorithms efficiently, to solve the efficiency problem of the existing algorithms in influence maximization. In this paper, we propose a three-phase algorithm PHG for influence maximization based on community structure. Moreover, we divide our algorithm into partition, heuristic and greedy three phases which will be discussed in next section.

C. COMMUNITY DETECTION

A community is characterized as a subset of nodes which are more linked to each other closely than other nodes outside the community. In order to identify a good community structure in a network, a large number of outstanding algorithms such as FPMQA[17], MECM[18] and BiLPA[19] have been proposed in the last decade years. Among these algorithms, the modularity-based algorithms such as Newman Fast algorithm[20] and Louvain algorithm[21] are popular and widely used in community detection. Modularity (Q) as the objective function is used to measure the quality of the community partition. Moreover, the value of Q is greater, indicating that the network community detecting is better. Although using modularity to detect community achieves a good result, it turns out to be computationally expensive for the whole networks[26]. Thus, some works focus on utilizing the modularity increment function to detect community which is computed as follows:

$$\Delta Q = [\frac{\sum_{in}+k_{i,in}}{2m} - \left(\frac{\sum_{tot}+k_i}{2m}\right)^2] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right] \quad (2)$$

where $\sum_{in}$ is the sum of the weights of the links inside the community to which the node i is assigned, $\sum_{tot}$ is the sum of the weights of the links incident to nodes in the community, k_i is the sum of weights of the edges connected to the node i, $k_{i,in}$ is the sum of weights of links from node i to nodes in community, and m is the total weights in the network.

In this paper, we select Louvain algorithm with low time complexity $O(n \log n)$ as our community detection algorithm. Owing to the simplicity of this algorithm, it can be computed extremely fast even in a large-scale network. Moreover, this algorithm obtains close to the nature communities in networks. Thus, we use Louvain algorithm to detect community. The Louvain algorithm consists of two steps, namely modularity optimization and new graph construction respectively. The detail of the Louvain algorithm is summarized below by its description using Algorithm 2.

Algorithm 2 Louvain Algorithm

Input: Network $G = (V, E)$
Output: community structure C
1: Initialize each node with its own community C_i, $\Delta Q = 0$
2: while $\Delta Q \geq 0$ do//step1: modularity optimization
3: for each $v \in V$ do
4: Put the node v into its each neighbor's community C_i
5: Compute ΔQ
6: $C_i = \max(\Delta Q)$
7: $C_i = C_i \cup \{v\}$
8: for each $C_i \in C$ do // step2: new graph construction
9: each C_i as a new node input step1 to construct a new graph
10: Return C

Algorithm 2 detailedly outlines the calculation process of the Louvain algorithm. First, for each node of the network, the algorithm initially allocates it for its own community and removes it from its original community to its neighbor's community according to modularity increment(ΔQ) maximization standard(lines 1-7). After finishing the modularity optimization, the network is divided into a number of communities. And then, for obtained each new community,

the algorithm constructs a new graph by inputting it as a new node into step1(lines 8-9). This process is repeated iteratively until communities of all nodes no longer change. The Louvain algorithm performs well in community detection. Besides, it is proved to be successful when applying on many different types of large-scale networks[25].

III. PHG ALGORITHM

In this section, we firstly present the design of algorithm PHG. Then, based on community structure, we divide our algorithm into three phases which are detailed in the following section.

A. THE DESIGN OF ALGORITHM PHG

The proposed algorithm is comprised of three phases, i.e., the partition phase, the heuristic phase and the greedy phase. In the partition phase, we deign an efficient algorithm CCSC that constructs a candidate set by detecting community structure. Based on community structure, in the heuristic phase, we select seeds from a candidate set by considering influence weight of nodes and the community influence of nodes. In the greedy phase, we select seeds with maximization marginal gain from the candidate set. The framework of the proposed algorithm is shown in FIGURE 1.

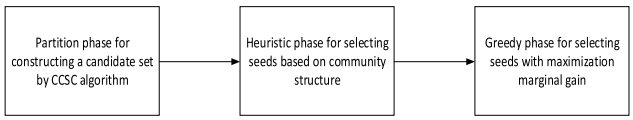


FIGURE 1. Framework of the proposed algorith.

In the partition phase of our proposed algorithm, we find key nodes to construct a candidate set by detecting community structure. The key nodes include the core nodes and boundary nodes which can spread information quickly in its own community and easily spread information among different communities respectively. In the real world, there is a same phenomenon with our proposed algorithm: a part of people spread information often rely on their circle of friends. In other words, the more friends a person has in his circle of friends, the faster the information spreads. While there are still some other people who are not in the same circle of friends with their friends due to the difference in interests, regions and so on. This part of people spread information from a circle to another circle by their friends which expands the breadth of the information transmission. For example, in Figure2, it shows two different-sized communities, community1 and community2, which contains $\{1, 2, 3, 4, 5, 6\}$ and $\{7, 8, 9\}$ respectively. The color node means we select it as a member of the candidate set. As in Figure2(a), if we only choose node1 to join in the candidate set, the community1 will obtain considerable amount gains while the community2 will gain a little. However, if we choose node1 and node3 to influence other nodes, as in Figure2(b), the probability of information spread to the community2 will be increased considerably. The reason for selecting node1 and

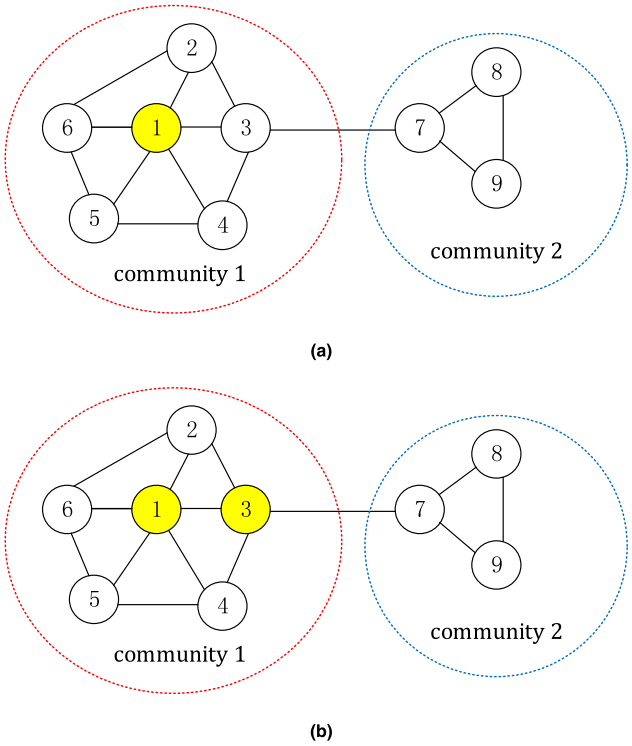


FIGURE 2. Examples of selection of seeds. (a): Select node1 as a see. (b): Select node1 and node3 as seeds.

node3 instead of selecting node1 to influence other nodes is that the node1 as a core node can only spread the information in its own community, while the node3 as a boundary node in community1 can spread information to the community2 by its neighbor node7.Thus, we select core nodes and boundary nodes to construct a candidate set, in order to further expand the influence spread of nodes.

B. PARTITION PHASE

We have introduced the importance of key nodes during the process of influence propagation in the previous part. Notice that as social networks in real world are extremely large, the search space for selecting seeds is a difficult task. Therefore, it is necessary to construct a candidate set by detecting community structure to reduce the scope of seeds search. In order to describe CCSC algorithm, we will first give the following definitions.

Definition 1 (Community Label): For a social network $G = (V, E)$ contains M communities $C = (C_1, C_2, \ldots C_M)$, the community label of node v is defined as

$$C_j(v), j = 1, 2, \ldots M \tag{3}$$

Obviously, the community label describes the approximate location of nodes in the network. Moreover, it is an important indication to determine nodes' attributes, especially the boundary nodes. As we have discussed in above, the boundary nodes connect two different communities which expand the breadth of the information transmission. Thus, based on community label, we define boundary label as follows:

Definition 2 (Boundary Label): For each node $v \in V$ with its neighbor w, the boundary label of node v is defined as

$$\begin{cases} B(v) = 1, & C(v) \neq C(w) \\ B(v) = 0, & C(v) = C(w) \end{cases} \tag{4}$$

where $C(v)$ represents the community label of node v. If the node v has different boundary label from its neighbor w, $B(v) = 1$, otherwise, $B(v) = 0$.

In terms of the community label and the boundary label, we have identified the boundary nodes. Based on the above analysis, we know that the candidate node set is composed of the boundary nodes and the core nodes. What criteria do we rely on to identify core nodes? Degree centrality is the most direct metric for characterizing node centrality in network analysis. Moreover, the greater the degree of a node, the more important the location of a node in the network. Therefore, we consider the degree centrality to identify the core nodes which is defined as follows:

Definition 3 (Degree Centrality): For a social network $G = (V, E)$ with n nodes, the degree centrality of node is defined as

$$C_D(v) = \sum_{w=1}^{n} e_{vw}(v \neq w) \tag{5}$$

where $C_D(v)$ denotes as the degree centrality of node v, $\sum_{w=1}^{n} e_{vw}$ is used to compute the number of direct connection between the node v and the other $n-1$ nodes w.

Depending on Definition 2 and Definition 3, we can determine the boundary nodes and the core nodes in each community. Moreover, the main purpose of the partition phase is to construct a candidate set by finding the boundary nodes and the core nodes. In order to describe the construction of the candidate set, we give the following definition.

Definition 4 (CS): Given a network $G = (V, E)$ with M communities $C = (C_1, C_2, \ldots C_M)$, where S_{core} is the sum of core node set and $S_{boundary}$ is the sum of boundary node set. The candidate set is defined as

$$CS = (S_{core} \cup S_{boundary}) \tag{6}$$

In formula (6), the sum of core node set is defined as

$$S_{core} = \bigcup_{k=1}^{M} C_{C_k} \tag{7}$$

C_{C_k} denotes the core node set in each community.

The sum of boundary node set is defined as

$$S_{boundary} = \bigcup_{k=1}^{M} B_{C_k} \tag{8}$$

B_{C_k} denotes the boundary node set in each community.

Based on above definitions, the candidate set can be constructed by CCSC algorithm. Given a directed graph $G = (V, E)$ and parameters P denoted as the size of the core node set in each community. Noticed that $P = 10\%$ of community size is enough for selecting good nodes in most cases. The CCSC algorithm with finding a candidate set can be described as follows.

Algorithm 3 CCSC Algorithm

Input: Network $G = (V, E)$, parameters P
Output: a candidate set CS
1: $CS \leftarrow \Phi$
2: $C = (C_1, C_2, \ldots C_M) \leftarrow$ *Louvain* algorithm
3: for each C_k do
4: for each $v \in C_k$ do
5: if $(B(v) == 1)$ then
6: $B_{C_k} = B_{C_k} \cup \{v\}$
7: else
8: while $P \left| C_{C_k} \right| < P$ do
9: $v_0 \leftarrow \max(C_D(v))$
10: $C_{C_k} = C_{C_k} \cup \{v_0\}$
11: $S_{boundary} \leftarrow \bigcup_{k=1}^{M} B_{C_k}, S_{core} \leftarrow \bigcup_{k=1}^{M} C_{C_k}$
12: $CS \leftarrow (S_{core} \cup S_{boundary})$
13: return CS

The core idea of the CCSC algorithm is to divide into M communities in which finding key nodes to construct a candidate set. Firstly, we conduct the Louvain algorithm used to divide communities, denoted as $C = (C_1, C_2, \ldots C_M)$, which was described in Algorithm 2, (lines 1-2). In order to find key nodes which is consist of core nodes and boundary nodes in each community. Secondly, we need to determine each node's boundary attribute for each community, if the node is a boundary node, we will add the node to the boundary node set, that is B_{C_k}, $1 \leq k \leq M$ (lines 3-6). Otherwise, we will select the top-P degree nodes to join in the core node set $C_{C_k} 1 \leq k \leq M$ (lines 7-10). Finally, we integrate each boundary node set B_{C_k} and each core node set C_{C_k} in each community to form S_{core} and $S_{boundary}$ respectively, and then add the two sets to the candidate set CS (lines 11-13).

C. HEURISTIC PHASE

In light of the constructed the candidate set, the heuristic phase aims to find a part of the most potential influence nodes from a candidate set based on community structure. Notice that the candidate set is composed of core nodes and boundary nodes which can provide more information about the community structure. Especially, the boundary nodes are the bridge connecting the different communities which are very useful for influence maximization. Therefore, it needs to consider community structure information to measure a node's influence. Specifically, we estimate the community influence of nodes by combining the degree of nodes, the number of communities that nodes connected directly, and the size of the community. Community influence not only considers the degree of nodes but also combines the location and connectivity of nodes in the network, which can better evaluate the importance of nodes to influence transmission. Based on

above attributes, the community influence can be measured as follows.

Definition 5 (Community Influence): For a node v, the community influence is defined as

$$CI(v) = \begin{cases} C_D(v) + C_N(v) + AvgN_S(v)/3, & v \in S_{boundary} \\ C_D(v) + C_S(v)/2, & v \in S_{core} \end{cases} \tag{9}$$

In formula(9), $C_D(v)$ is the degree of the nodes, $C_N(v)$ denotes the number of communities which the node v connected directly, $AvgN_S(v)$ denotes the average size of the community which the node v's neighbors belong to, and C_S represents the size of the community which the node v belongs to. The $C_N(v)$ and $AvgN_S(v)$ are defined as below,

$$C_N(v) = \sum_{w \notin C_j(v), w \in neighbor(v)} e_{vw} \tag{10}$$

where e_{vw} denotes the number of neighbors of the node v.

$$AvgN_S(v) = \frac{\sum_{i \neq j, w \in neughbor(v), w \notin C_j(v)} |C_i(w)|}{C_N(v)} \tag{11}$$

where $|C_i(w)|$ represents the size of community which the node w belongs to.

In order to ensure that the contribution of each attribute in formula(9) to the final result is same. We adopt Min-Max normalization to handle original data that make the value of each attribute is mapped to [0-1] space. The formula can be listed as follows:

$$y = \frac{(x - \min value)}{(\max value - \min value)} \tag{12}$$

where x and y represent the value before and after conversion respectively.

Community influence describes the topology and importance of nodes in the whole network. However, we still can not ignore the influence weight which provides an important impact on inactive nodes[8]. Thus, we take the community influence and the influence weight into account to measure the potential influence. The potential influence can be measured as follows.

Definition 6 (Potential Influence): For a node v, the potential influence is defined as

$$P(v) = \sum_{w \in neighbor(v), w \notin A(v)} b_{vw} \cdot CI(v) \tag{13}$$

where b_{vw} represents the influence weight of node v on its neighbor w, $A(v)$ denotes the set of active neighbors of node v.

With the purpose of finding the most potential nodes in the heuristic phase, our method is to evaluate the key node's (i.e., core nodes and boundary nodes) potential influence by considering the community influence and the influence weight. Notice that this potential influence will be accumulated by activated nodes and will be maximized in the greedy phase, which results in activating more nodes than classical algorithms which will be discussed in section 4.

D. GREEDY PHASE

After finishing the most potential nodes selection, we apply the greedy algorithm(Algorithm 1) for finding the most influential nodes to achieve higher influence spread than the heuristic phase. Notice that the task of seed selection in the greedy algorithm is very time consuming. Therefore, we run the CCSC algorithm(Algorithm 3) for pruning the insignificant nodes to construct a candidate set. Different from the classical greedy algorithm, we select seeds from the candidate set instead of the entire network which effectively reduces the time that the node computes influence. Moreover, we obtain a better influence spread than other classical algorithms which performs in the later experiments.

E. PHG ALGORITHM

Based on the preceding descriptions for each phase, the algorithm for the PHG is shown in Algorithm 4.

Algorithm 4 PHG Algorithm

Input: Network $G = (V, E)$, k, c, a candidate set CS
Output: a seed set S
1: $S \leftarrow \Phi$, $CS \leftarrow \Phi$, $k_1 = \lceil ck \rceil$, $k_2 = k - \lceil ck \rceil$
Partition phase:
2: $CS \leftarrow CCSC$ algorithm
Heuristic phase:
3: for $i = 1$ to k_1 do
4: $\quad v = \arg\max_{u \in CS \setminus S}(P(u))$
5: $S_i = S_{i-1} \cup \{v\}$
Greedy phase:
6: for $i = 1$ to k_2 do
7: $\quad v = \arg\max_{u \in CS \setminus S}(\delta(S \cup \{u\}) - \delta(S))$
8: $S_{i+k1} = S_{i-1+k1} \cup \{v\}$

Given a directed graph $G = (V, E)$, a parameter k and a parameter c denoted as the size of seed set and heuristic factor respectively. The PHG algorithm can be described as follows.

Algorithm 4 shows the pseudo code of our solution. Firstly, the information of the community structure and a candidate set can be discovered by the CCSC algorithm. We prune some insignificant nodes by detecting community structure in order to reduce the running time for the next two phases (lines 1-2). Then we select the most influential nodes with a number of $\lceil ck \rceil$ based on community structure information as initial seeds in the heuristic phase (lines 3-5). At last, we conduct the greedy algorithm to select the largest marginal gain nodes as seeds in the greedy phase (lines 6-8). Notice that we select the initial seeds from the candidate set which contains a large amount of community structure information. The initial seeds may have a high degree or have a good connectively in each step through utilizing this community structure information which performs a good result that shows in experiments part.

1) TIME COMPLEXITY

The total calculation of our PHG algorithm mainly consists of three parts-partition phase, heuristic phase, and greedy phase.

During partition phase, it spends $O(n \log n)$ time to detect community, where n represents the number of nodes. Then, the computational complexity of $O(n)$. And finally, in the greedy phase, we select k influential seeds from candidate set by greedy algorithm, thus the time complexity is $O(kn'm')$, where n' denotes the number of candidate nodes and m' represents the number of the candidate edges. Consequently, the total time complexity of our algorithm is $O(n \log n + n + kn'm')$.

IV. EXPERIMENTS AND RESULTS

In order to compare our algorithm with other algorithms, we conduct our experiments on artificial and real-world social networks. First of all, we introduce the experiments setup in this experiment. Second, we analyze the performance of our algorithms.

A. EXPERIMENTS SETUP

Our experiments setup consists of: (a)the datasets, (b) the diffusion model, and (c) the algorithms to compare three aspects.

1) DATASETS

We use five datasets which contains all kinds of sizes and types data. The first one is called Amazon which was collected on March 02,2003. There contains a directed edge from i to j if a product i is frequently co-purchased with product j. Epinions and Brightkite are two-medium sized datasets with a who-trust-whom online social network and a location-based social network respectively. In Epinions website, all users may choose whether or not trust reviews which are posted by other users and the Brightkite network was collected using their public API. While the smaller dataset is HepTh which is from the e-print arXiv and contains the collaboration relations between authors in the High Energy Physics Theory. Finally, we use the pajek software to generate a Synthetic network at random.

All real-world datasets are available from SNAP library on the Stanford University website and the statistical properties of all datasets are summaries in TABLE 1.

TABLE 1. Summary of the datasets.

Datasets	Amazon	Epinions	Brightkite	HepTh	Synthetic
#Node	262111	75879	58228	9877	53036
#Edge	1234877	508837	214078	25998	130738
Max. degree	366	3027	1134	1383	355
Avg. degree	9.009	13.412	7.353	10.020	4.93
#Connected Components	1	11	547	428	829
Large component size	262111	75877	56739	8638	51241
Average component size	262111	6898	106	23	64

2) DIFFUSION MODEL

We use the Linear Threshold model to evaluate our algorithm. The influence weight $b_{v,w}$ is usually defined as $1/C_D(v)$, where $C_D(v)$ is the normal degree of v. It means that for an inactive node v, all its neighbors have the same contribution to the node v. However, this is not in accord with the real world. Thus, we give a new definition of the influence weight $b_{v,w}$, we not only consider the number of neighbors of v but also consider how these neighbors connect to each other.

$$b_{v,w} = \frac{C_D(v)}{\sum_{w \in N(v)} C_D(w)} \tag{14}$$

where $N(v)$ represents the set of neighbors of node v.

3) ALGORITHMS TO COMPARE

We compare the performance of our algorithm with five algorithms which contain a hybrid algorithm, a greedy algorithm and three heuristic algorithms. The following is a list of algorithms we evaluate in our experiments.

- PHG: The algorithm presented in this paper. The section of heuristic factor c is discussed in section 4.2.
- HPG: A hybrid algorithm[4] finds the influence of each node as a combination of the heuristic algorithm and the greedy algorithm to track the effect of each node in the entire network. In this algorithm, heuristic factor c sets as same as our algorithm.
- Greedy: A highly effective in influence spread discussed in Algorithm 1 which uses Monte Carlo simulations to compute the influence of each node. This algorithm chooses the largest marginal gain node to add it to the seed set in each step.
- PageRank: It was first proposed to rank webpages. We use this algorithm finds the most influential nodes as seeds. The algorithm stops when the score vectors from two consecutive iterations differ by at most 10-4 as every L1 norm.
- Degree: The algorithm chooses the nodes with maximization degree as seeds which is a standard method compared to other algorithms for social networks.
- Random: This algorithm chooses seeds at randomly.

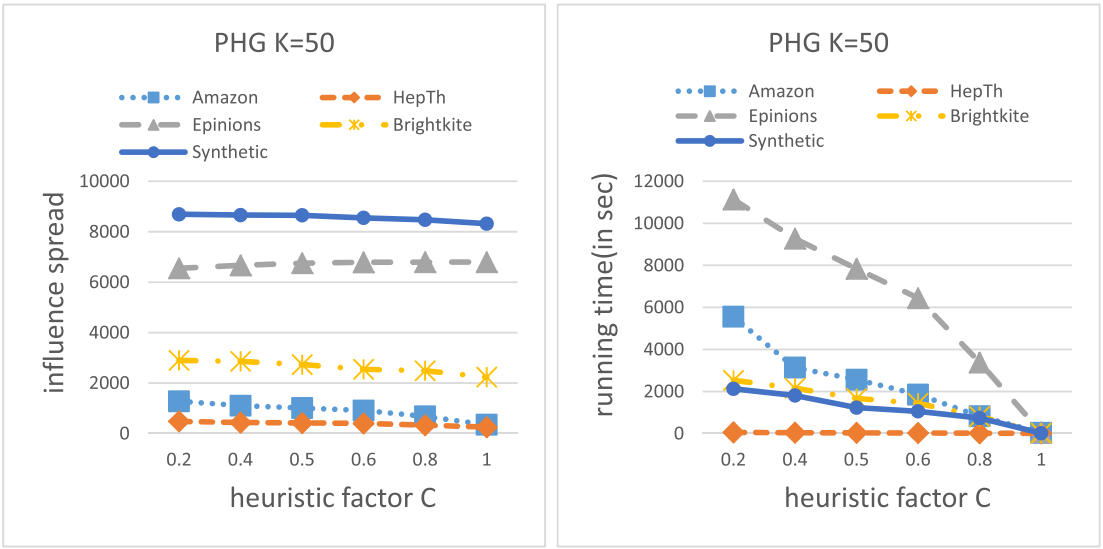
The performances of these algorithms runs are discussed in the next section. It is noted that HPG algorithm uses the hybrid idea which is similar to our algorithm. The greedy algorithm is a classical method to solve the influence maximization with a good influence spread. Other heuristic algorithms PageRank, Degree and Random are basic algorithms compared with most of the other works and have a good running time.

B. EXPERIMENTAL RESULTS

Our proposed algorithm is evaluated in four domains: (a) the community detection; (b) the tuning of the heuristic factor c;

TABLE 2. The results of the community detection of the datasetsd.

Datasets	Amazon	Epinions	Brightkite	HepTh	Synthetic
#Communities	218	1616	992	483	1029
Max. Com. Size	71532	10312	7653	791	1021
Min. Com. Size	4	1	2	1	1
Avg. Com. Size	1202.34	46.95	58.70	20.45	51.54
Modularity (Q)	0.915	0.897	0.874	0.768	0.815
Running time(s)	389	356	176	5	78
Parameter u	0.0006	0.00010	0.00026	0.00139	0.00098



(a) (b)

FIGURE 3. The effect of factor c on influence spread and running tim. (a): Spread of influence. (b): Running tim.

(c) the influence spread compared to other algorithms; and (d) the running time compared to other algorithms.

1) COMMUNITY DETECTION

We first evaluate the effect of community detection. We conduct the Louvain algorithm(Algorithm 2) to detect community which was proved that has a good performance [22]. In order to describe effectively the performance of the Louvain algorithm on each dataset, we do our experiments with a parameter u (the minimum/maximum size of the communities ($S_{\min}/S_{\max}$)), denoted as the average proportion of each node which does not belong to the same community with its neighbors in the network. The results of the community detection for different datasets are shown in TABLE 2.

TABLE 2 clearly shows the results of the community detection. As the above mentioned, the parameter u directly determines different types of the community structure, and smaller u means that the community structure is stronger. Moreover, the modularity Q is a metric to evaluate the quality of a community structure. Combing the performance of the modularity and parameter u in each dataset, we conclude that the higher the value of modularity, the stronger of the community structure. Moreover, the value of modularity of five datasets range from 0.76 to 0.91 indicates the results of the community detection are accurate. Because of the datasets on influence maximization are large-scale networks, we must consider the running time when detecting community. From Table 2 we know that all datasets have a fast running, especially in HepTh.

2) TUNING OF HEURISTIC FACTOR c

We evaluate the effect of the changing of the factor c on influence spread and running time with five datasets and results are shown in Figure3. It is clear that with the growth of c, the influence spread on all datasets decreases gradually expect the Epinions. Similarly, with the growth of c, the running time on all datasets decreases quickly. This is accordance with the characteristics of heuristic algorithms with low influence

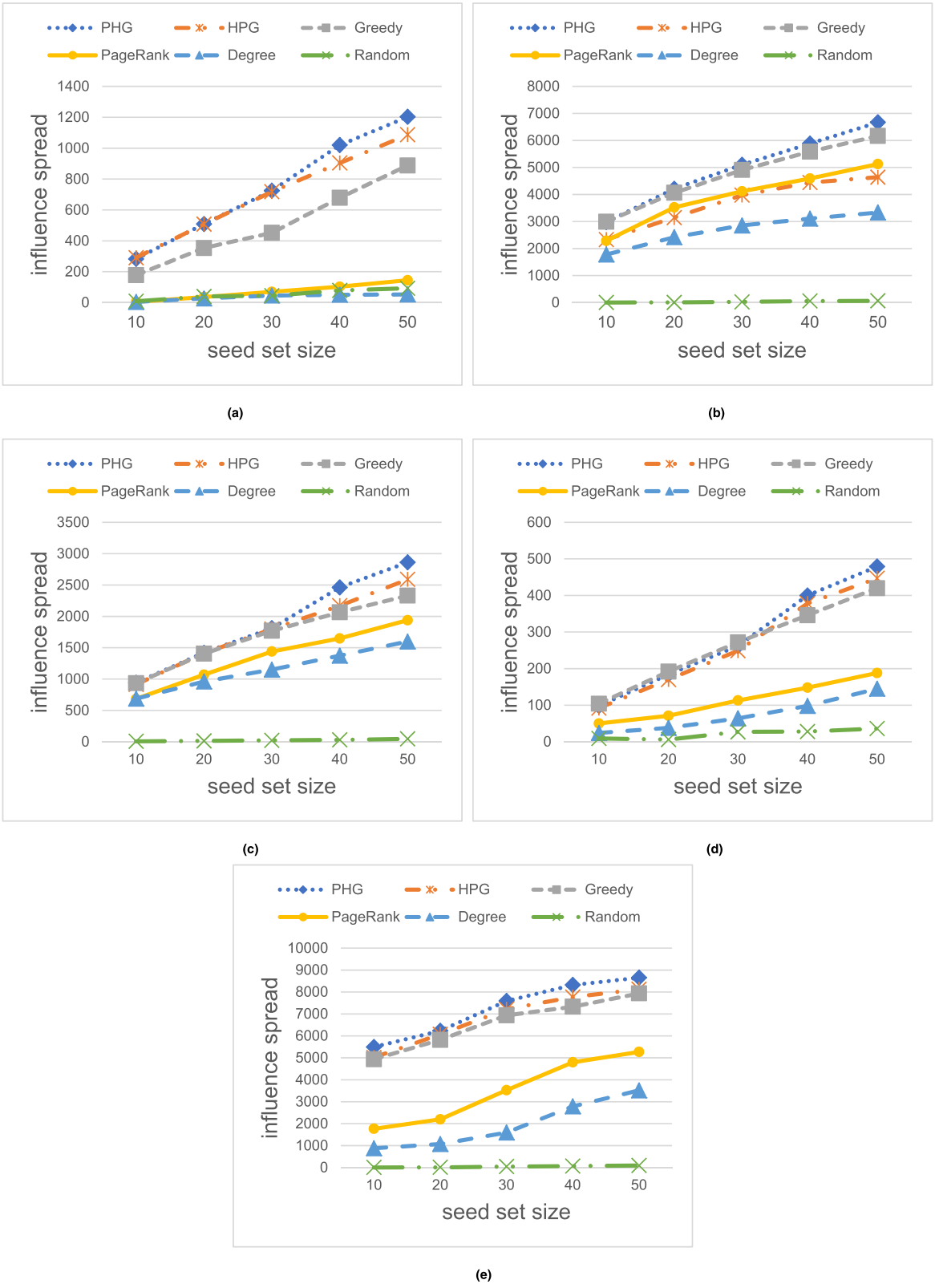


FIGURE 4. The influence spread on different datasets. (a): Amazon (b): Epinions. (c): Brightkite. (d): HepTh. (e): Synthetic.

spread and running time and greedy algorithms with high influence spread and running time. Therefore, in order to obtain a suitable influence spread and running time, we set the value of factor c on Amazon, Brightkite and Synthetic are 0.4, 0.4 and 0.5 respectively. Observed the Figure3, we discover that the greedy element has little effect on the Epinions.

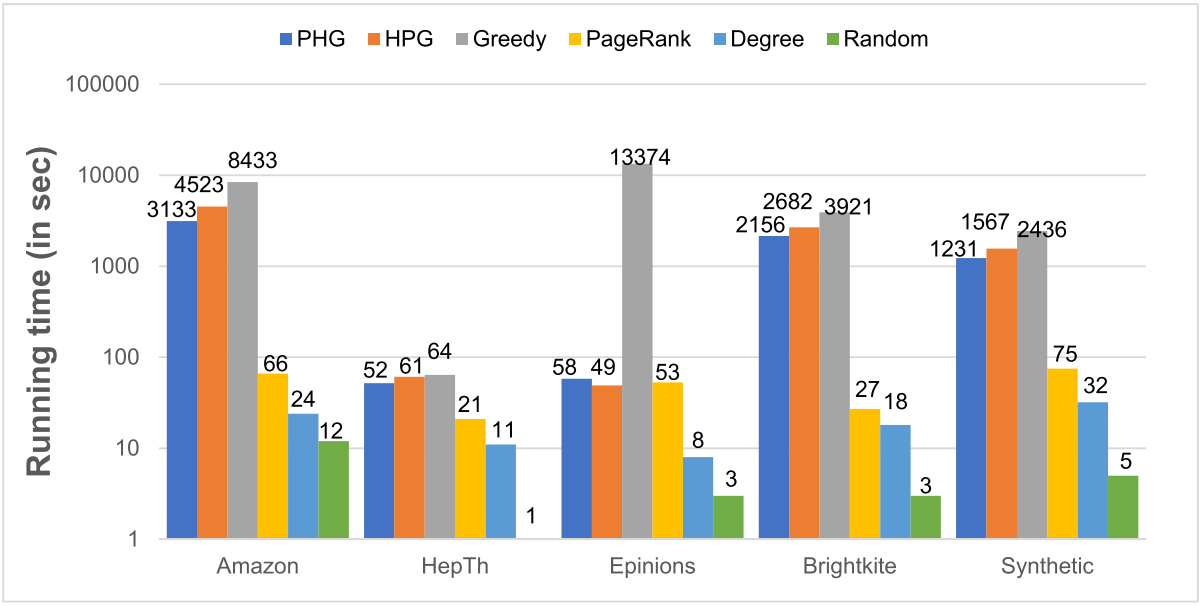


FIGURE 5. Running time of different algorithms in four datasets.

Thus, we set the value of factor c on Epinions is 1. Owing to the small size of the HepTh, the running time is not a main factor limiting the performance of this dataset. According to the influence spread, we set the value of factor c on HepTh is 0.2.

3) INFLUENCE SPREAD

We compare our algorithm with other algorithms on five datasets, as shown in FIGURE 4. We can discover that our algorithm performs best among other algorithms according to the influence spread. As our expectation, the Random algorithm as the baseline performs worst on all datasets. This primarily is because the Random algorithm does not consider any feature of the network, while other algorithms all effectively utilize the attributes of the network more or less. The simple heuristic algorithms including PageRank and Degree are better than Random, however, they are still significantly worse than other algorithms including Greedy, HPG and PHG on most datasets except for the Epinions dataset. As high influence algorithms, HPG and Greedy have a similar influence spread with our algorithm PHG when seed set size is smaller. However, the performance of our algorithm is becoming better and better as the seed set size is increasing. For example, our algorithm is 10.50% and 22.78% better than HPG and Greedy on Brightkite dataset when the seed set size is 50. The result clearly shows that our algorithm can effectively find top influential seeds by taking advantage of community structure information which provides the degree, the location and the connectivity of a node in a network.

4) RUNNING TIME

The FIGURE 5 shows the running time of different algorithms on several datasets. Here the running time is the time of selecting $k = 50$ seeds.

From the results we see that the running time of PageRank, Degree and Random are considerably low on all datasets. The reason is probably that these algorithms only consider the single element to heuristically choose seeds which can't provide any performance guarantee, greatly shortening the running time in the information transmission process. Except PageRank, Degree and Random, the PHG has the best running time among other algorithms. For example, the PHG is 62.85% and 30.73 % lower than Greedy and HPG on Amazon dataset. The reason is that, the PHG algorithm prunes insignificant nodes with little community structure information which leads to the reduction of the range of seed selection. It is noticed that the PHG algorithm and the HPG algorithm have a less running time on Epinions dataset when comparing with other datasets. This difference is mainly due to different values of the heuristic factor c. And based on above analyze on the heuristic factor c, we know that the larger the value of the heuristic factor c, the less the greedy algorithm's contribution to the PHG algorithm and the HPG algorithm. Therefore, the greedy algorithm does not any impact both on the PHG algorithm and the HPG algorithm when $c = 1$ on Epinions dataset, which reduces the running time drastically.

V. CONCLUSIONS

In this paper, we propose a novel hybrid PHG, which is a three-phase algorithm for influence maximization based on community structure. Previous studies usually ignore the role of community structure which captures a significant effect on the process of influence propagation. Thus, we take the community structure information into account and divide the influence propagation process into three phases: (i): partition phase; (ii): heuristic phase; (iii): greedy phase. Firstly, we design an efficient algorithm CCSC that finds key nodes

in each community to construct a candidate set by detecting community structure. And then we find the most potential influence nodes from a candidate set by combing the influence weight of nodes and the community influence of nodes through the analysis of the community structure of the impact on nodes. At last, we greedily select the nodes with maximization marginal gain from remaining a candidate set. We evaluate the performance of our proposed algorithm on the real datasets and synthetic dataset which achieves excellent stability in influence spread as well as an acceptable running time.

We believe our study in influence maximization problem will provide more different directions in the future. First, we use community detection method is non-overlapping, while in the real world the community structure is overlap. Therefore, we should take into account how apply our algorithm to the overlapping community. Second, we only consider our algorithm on the linear threshold model without other diffusion models such as the Independent Cascade Model and the Weighted Cascade Model. Finally, we will analyze the framework of our algorithm to future improve influence spread and running time.

REFERENCES

[1] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2001, pp. 57–66.

[2] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2003, pp. 137–146.

[3] Z. L. Luo *et al.*, "A PageRank-based heuristic algorithm for influence maximization in the social network," in *Recent Progress in Data Engineering and Internet Technology*, 2012, pp. 485–490.

[4] T. J. Tian, Y. T. Wang, and X. J. Feng, "A new hybrid algorithm for influence maximization in social networks," *Chin. J. Comput.*, vol. 35, no. 10, pp. 1956–1965, Oct. 2011.

[5] C. Kim *et al.*, "Influence maximization algorithm using Markov clustering," *Database Syst. Adv. Appl.*, vol. 7827, no. 2013, pp. 512–515, Apr. 2013.

[6] G. Zhang *et al.*, "Research on user competition of communication networks based on community structure and linear threshold model," in *Proc. 7th Int. Conf. Intell. Hum.-Mach. Syst. Cybern.*, Aug. 2015, pp. 378–381.

[7] Y. Wang *et al.*, "Community-based greedy algorithm for mining top-k influential nodes in mobile social networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2010, pp. 1039–1048.

[8] Y. Wang and X. Feng, "A potential-based node selection strategy for influence maximization in a social network," *Lect. Notes Comput. Sci.*, vol. 5678, pp. 350–361, Aug. 2009.

[9] A. Goyal, W. Lu, and L. V. S. Lakshmanan, "CELF++: Optimizing the greedy algorithm for influence maximization in social networks," in *Proc. Int. Conf. Companion World Wide Web*, Jan. 2011, pp. 47–48.

[10] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Sep. 2009, pp. 199–208.

[11] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Jul. 2010, pp. 1029–1038.

[12] J. Shang, L. Liu, F. Xie, and C. Wu, "How overlapping community structure affects epidemic spreading in complex networks," in *Proc. 38th Int. Comput. Softw. Appl. Conf. Workshops*, Jul. 2014, pp. 240–245.

[13] J. Shang *et al.*, "Epidemic spreading on complex networks with overlapping and non-overlapping community structure," *Phys. A, Stat. Mech. Its Appl.*, vol. 419, pp. 171–182, Feb. 2015.

[14] X. Zhang *et al.*, "Identifying influential nodes in complex networks with community structure," *Knowl.-Based Syst.*, vol. 42, no. 2, pp. 74–84, Apr. 2013.

[15] Y. C. Chen *et al.*, "CIM: Community-based influence maximization in social networks," *ACM Trans. Intell. Syst. Technol.*, vol. .5, no. 25, p. 25, Apr. 2014.

[16] H. Li, S. S. Bhowmick, A. Sun, and J. Cui, "Conformity-aware influence maximization in online social networks," *VLDB J.*, vol. 24, no. 1, pp. 117–141, 2015.

[17] Z. Bu, C. Zhang, Z. Xia, and J. Wang, "A fast parallel modularity optimization algorithm (FPMQA) for community detection in online social network," *Knowl.-Based Syst.*, vol. 50, pp. 246–259, Sep. 2013.

[18] K. Zhou *et al.*, "Median evidential c-means algorithm and its application to community detection," *Knowl.-Based Syst.*, vol. 74, no. 1, pp. 69–88, Jan. 2015.

[19] Z. Li *et al.*, "Quantitative function and algorithm for community detection in bipartite networks," *Inf. Sci.*, vols. 367–368, pp. 874–889, Nov. 2016.

[20] M. E. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E, Stat. Nonlinear Soft Matter Phys.*, vol. 69, no. 2, Jul. 2003, Art. no. 066133.

[21] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech., Theory Exp.*, vol. 10, pp. 155–168, 2008.

[22] J. Shang, L. Liu, X. Li, F. Xie, and C. Wu, "Targeted revision: A learning-based approach for incremental community detection in dynamic networks," *Phys. A, Stat. Mech. Its Appl.*, vol. 443, pp. 70–85, Feb. 2016.

[23] Y. C. Chen, W. C. Peng, and S. Y. Lee, "Efficient algorithms for influence maximization in social networks," *Knowl. Inf. Syst.*, vol. 33, no. 3, pp. 577–601, Dec. 2012.

[24] H. Shi *et al.*, "A high-influence greedy maximization algorithm based on community structure," *J. Comput. Inf. Syst.*, vol. 11, no. 2, pp. 449–456, Jan. 2015.

[25] I. Perisic and I. Perisic, "Mapping search relevance to social networks," in *Proc. Workshop Social Netw. Mining Analysis*, New York, NY, USA, Jun. 2009, pp. 1–7.

[26] C. Y. Cheong *et al.*, "Hierarchical parallel algorithm for modularity-based community detection using GPUs," in *Proc. Int. Conf. Parallel Process.*, 2013, pp. 775–787.

[27] J. Shang *et al.*, "CoFIM: A community-based framework for influence maximization on large-scale networks," *Knowl.-Based Syst.*, vol. 117, pp. 88–100, Feb. 2016.

[28] M. Purohit *et al.*, "Fast influence-based coarsening for large networks," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 1296–1305.

[29] Y. Wang, "Community-based greedy algorithm for mining top-K influential nodes in mobile social networks," in *Proc. ACM Sigkdd Int. Conf. Knowl. Discovery Data Mining*, Aug. 2010, pp. 1039–1048.

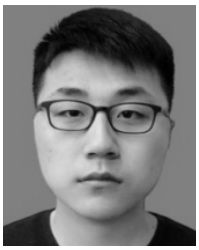
[30] X. Li *et al.*, "Community-based seeds selection algorithm for location aware influence maximization," *Neurocomputing*, vol. 275, pp. 88–100, Oct. 2018.

[31] J. Shang, H. Wu, S. Zhou, J. Zhong, Y. Feng, and B. Qiang, "IMPC: Influence maximization based on multi-neighbor potential in community networks," *Phys. A, Stat. Mech. Appl.*, vol. 512, pp. 1085–1103, Dec. 2018.

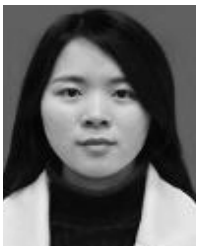
LIQING QIU was born in 1978. She received the Ph.D. degree in computer software and theory from Beihang University, Beijing, China. She is currently a Lecturer with the Shandong University of Science and Technology.

WEI JIA was born in 1994. She received the B.S. degree from Qingdao Agricultural University, Qingdao, China, in 2017. She is currently pursuing the M.S. degree with the School of Computer Science and Engineering, Shandong University of Science and Technology. Her current research interest includes social networks.

JINFENG YU was born in 1993. She received the B.S. degree from the School of Electronic Information Engineering, University of Shanghai Dianji, Shanghai, China, in 2016. She is currently pursuing the M.S. degree with the School of Computer Science and Engineering, Shandong University of Science and Technology. Her current research interests include data mining and social networks.



XIN FAN was born in 1994. He received the B.S. degree from Linyi University, Linyi, China, in 2017. He is currently pursuing the M.S. degree with the School of Computer Science and Engineering, Shandong University of Science and Technology. His current research interest includes social networks.



WENWEN GAO was born in 1992. She received the B.S. degree from the School of Information Science and Electrical Engineering, Shandong Jiaotong University, Jinan, China, in 2016. She is currently pursuing the M.S. degree with the School of Computer Science and Engineering, Shandong University of Science and Technology. Her current research interests include data mining and social networks.