Knowledge-Based Systems 42 (2013) 74–84

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Identifying influential nodes in complex networks with community structure

Xiaohang Zhang [a,*], Ji Zhu [b], Qi Wang [a], Han Zhao [a]

[a] *School of Economics and Management, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[b] *Department of Statistics, University of Michigan, Ann Arbor, MI 48109, USA*

ARTICLE INFO

Article history:
Received 2 July 2012
Received in revised form 14 January 2013
Accepted 16 January 2013
Available online 26 January 2013

Keywords:
Influential nodes
Complex networks
Community
Bond percolation process
k-Medoid clustering

ABSTRACT

It is a fundamental issue to find a small subset of influential individuals in a complex network such that they can spread information to the largest number of nodes in the network. Though some heuristic methods, including degree centrality, betweenness centrality, closeness centrality, the *k*-shell decomposition method and a greedy algorithm, can help identify influential nodes, they have limitations for networks with community structure. This paper reveals a new measure for assessing the influence effect based on influence scope maximization, which can complement the traditional measure of the expected number of influenced nodes. A novel method for identifying influential nodes in complex networks with community structure is proposed. This method uses the information transfer probability between any pair of nodes and the *k*-medoid clustering algorithm. The experimental results show that the influential nodes identified by the *k*-medoid method can influence a larger scope in networks with obvious community structure than the greedy algorithm without reducing the expected number of influenced nodes.

1. Introduction

Complex networks are pervasive to natural and social sciences, ranging from social and information networks to technological and biological networks [1,2]. One main function of complex networks is to transfer information, trust, ideas, diseases and influences between any two nodes [3]. The information can spread beyond the dyad, occasionally reaching a large number of nodes. Spreading is a ubiquitous process, which describes many important network activities [4,5]. The knowledge of the roles that network nodes play in the spreading process is crucial for developing efficient methods to either hinder or accelerate spreading, given diseases or information, respectively.

It is a fundamental issue to find a small subset of influential individuals in a complex network such that they can spread information to the largest number of nodes in the network [6–8]. The solution to this issue has many applications [6,9,10]. Kempe et al. give a vivid example [6]. A company initially targets a small number of "influential" individuals in the network by giving them free product samples and hopes that the initially selected users will recommend the product to their friends, their friends will influence their friends' friends and so on; many individuals will thus ultimately adopt the new product through the powerful word-of-mouth effect (also called viral marketing). The problem is formally called influence maximization: for a parameter k, find a *k*-node set to maximize the influence effect where influence is propagated in the network according to a stochastic cascade model [6]. The influence effect is usually measured by the expected number of influenced nodes at the end of the diffusion process.

Many networks of interest in the sciences, including social, computer, metabolic and regulatory networks, are found to divide naturally into communities or modules [11]. In this study, we analyze the effects of some popular methods on identifying influential nodes in complex networks with community structure, including degree centrality, betweenness centrality, closeness centrality, the *k*-shell decomposition method and a greedy algorithm. We argue that sometimes these methods have limitations when applied to complex networks with community structure as they do not explicitly account for networks' community structure and do not identify the influential nodes from communities in a relatively balanced way. For example, the nodes with high degree that are often believed to be influential players may all lie in the same community with larger size such that they can only influence nodes in the same community. In this study, we propose a novel method for identifying influential nodes in complex networks with community structure. We also propose a new measure for assessing the influence effect based on influence scope maximization, which can complement the traditional measure of the expected number of influenced nodes. Our method is compared with the greedy algorithm [6] based on some real and computer-simulated networks. To study the spreading process, we apply a widely used information diffusion model, the independent cascade (IC) model [12,6,8].

[*] Corresponding author. Tel.: +86 13910390890.
E-mail addresses: zhangxiaohang@bupt.edu.cn (X. Zhang), jizhu@umich.edu (J. Zhu), buptwangqi@bupt.edu.cn (Q. Wang), h_zhao@bupt.edu.cn (H. Zhao).

0950-7051/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.knosys.2013.01.017

2. Definition

The influence maximization problem is examined on a directed and weighted network $G = (V, E, W)$ for the IC model. Here, V and E are the sets of all nodes and links in the network, respectively. Let N and L be the numbers of elements of V and E, respectively. W is the corresponding weight set of E, i.e., each link $(u, v) \in E$ from node u to v has a corresponding weight $w_{uv} \in W$.

2.1. Independent cascade model

In the IC model, each link $(u, v) \in E$ is assigned a real value $\beta_{uv} \in [0,1]$ that is referred to as probability of information transfer through link (u, v). For weighted networks, we assume that weight w_{uv} denote connection strength through link (u, v) and $w_{uv} \geqslant 0$ for any $w_{uv} \in W$. In most real networks, weights can be transformed to denote connection strength through some simple operations. β_{uv} can be defined thus:

$$\beta_{uv} = 1 - (1 - \beta)^{w_{uv}}, \tag{1}$$

where $\beta \in [0,1]$ is a designated propagation probability and w_{uv} is the weight of link (u, v). In this study, for unweighted networks, we set $w_{uv} = 1$ for any link $(u, v) \in E$. Thus, propagation probability β_{uv} of each link (u, v) can be specified directly or be computed based on the weight w_{uv} and β. In this study, we adopt the latter to determine the propagation probability.

In the IC model, some assumptions are made: (1) The state of a node is either *active* or *inactive*; a node is in state of active if it has adopted the information. (2) Nodes can switch from being inactive to being active, but cannot switch from being active to being inactive. (3) The diffusion process of active nodes unfolds in discrete time-steps $t \geqslant 0$; at $t = 0$ the nodes in an initial set A first become active and all the other nodes are in state of inactive. The diffusion process of IC model is presented formally in Algorithm 1.

Algorithm 1. Independent cascade model

```
Input: $G = (V,E,W),\beta$ and $A$ /*$G$ is the network; $\beta$ is a
    designated propagation probability; $A$ is an initial active set
    with k nodes.*/
Output: $AS$ /*$AS$ is the set of active nodes at the end of the
    spreading process*/
$AS := A$;
$CA := A$; /* $CA$ is the set of active nodes in current time-step */
$NA := \emptyset$; /* $NA$ is the set of active nodes in next time-step */
do {
  for (each node $u$ in $CA$)
    for (each node $v$ in $\mathcal{N}(u)$) /* $\mathcal{N}(u) = \{x|(u,x) \in E, x \notin AS\}$ */
      if ($v$ adopt information from $u$ with probability $\beta_{uv}$)
        $NA := NA \cup \{v\}$;
  $AS := AS \cup NA$;
  $CA := NA$;
  $NA := \emptyset$;
} while ($CA \neq \emptyset$);
output $AS$;
End Algorithm
```

2.2. Measures of information propagation effects

In this study, three measures of information propagation effects are defined. The first is the average number of active nodes at the end of the spreading process $\sigma(A)$ that is often used in many previous studies [6–8]. $\sigma(A)$ is computed thus:

$$\sigma(A) = \frac{1}{n}\sum_{i=1}^{n}|AS_i(A)|, \tag{2}$$

where A is an initial set of active nodes; $AS_i(A)$ is the output set of active nodes at the end of ith spreading simulation process of the IC model; $|AS_i(A)|$ is the number of nodes in $AS_i(A)$; and n is the number of independent spreading simulations of the IC model.

The second measure is the average scope of active nodes at the end of multiple spreading processes $\varphi(A)$, computed thus:

$$\varphi(A) = \frac{1}{H}\sum_{h=1}^{H}\sum_{v \in V}\omega(v, S_h(m, A)), \tag{3}$$

where H is the number of experimental groups; each group consists of m independent spreading process simulations of the IC model; V is the set of nodes in network; $S_h(m,A)$ is the union of the output sets of active nodes in the m independent spreading simulations of the hth experimental group; $\omega(v,S_h(m,A)) = 1$ if $v \in S_h(m,A)$, 0 otherwise. With multiple spreading processes, often pervasive in reality, improving $\varphi(A)$ can guarantee that the information is more likely to spread in a larger scope.

The third measure is the probability of node $v \in V$ adopting the information, computed thus:

$$IP_v(A) = \frac{1}{n}\sum_{i=1}^{n}\omega(v, AS_i(A)), \tag{4}$$

where A is an initial set of active nodes; $AS_i(A)$ is the output set of active nodes at the end of the ith spreading simulation process of the IC model; n is the number of independent spreading simulations; $\omega(v,AS_i(A)) = 1$ if $v \in AS_i(A)$, 0 otherwise.

2.3. LFR model

In this study, we use some artificial networks that are generated by the LFR model [13,14] to evaluate the influential nodes identification methods. In the LFR model, both the degree and the community size distributions of the artificial networks are assumed to follow power laws, with exponents γ and λ, respectively. The number of nodes is N and the average degree is $\langle k \rangle$. The construction of a realization of LFR model proceeds through the following steps.

(1) Each node is given a degree taken from a power law distribution with exponent γ. The extremes of the distribution k_{min} and k_{max} are chosen such that the average degree is $\langle k \rangle$. The configuration model [15] is used to connect the nodes so to keep their degree sequence.
(2) Each node shares a fraction $1 - \mu$ of its links with the other nodes of its community and a fraction μ with the other nodes of the network; μ is the mixing parameter.
(3) The sizes of the communities are taken from a power law distribution with exponent λ, such that the sum of all sizes equals the number N of nodes of the graph. The minimal and maximal community sizes s_{min} and s_{max} are chosen so to respect the constraints imposed by the definition of community: $s_{min} > k_{min}$ and $s_{max} > k_{max}$. This ensures that a node of any degree can be included in at least a community.
(4) At the beginning, all nodes are homeless, i.e., they are not assigned to any community. In the first iteration, a node is assigned to a randomly chosen community; if the community size exceeds the internal degree of the node (i.e., the number of its neighbors inside the community), the node enters the community, otherwise it remains homeless. In successive iterations we place a homeless node to a randomly chosen community: if the latter is complete, we kick out a randomly selected node of the community, which becomes homeless. The procedure stops when there are no more homeless nodes.

(5) To enforce the condition on the fraction of internal neighbors expressed by the mixing parameter μ, several rewiring steps are performed, such that the degrees of all nodes stay the same and only the split between internal and external degree is affected, when needed. In this way the ratio between external and internal degree of each node in its community can be set to the desired share μ with good approximation.

(6) In order to build a weighted network, an unweighted network with a given topological mixing parameter μ_t is first generated and then a positive real number is assigned to each link. To do this two parameters, α and μ_w, are specified. The parameter α is used to assign a strength s_u to each node $u, s_u = k_u^{\alpha}$; such power-law relation between the strength and the degree of a node is frequently observed in real weighted networks. The parameter μ_w is used to assign the internal strength $s_u^{(in)} = (1 - \mu_w)s_u$, which is defined as the sum of the weights of the links between node u and all its neighbors having at least one membership in common with u.

(7) In order to build a network with overlapping communities, two parameters, O_n and O_m, need to be specified. The parameter O_n is the number of overlapping nodes in the network. The parameter O_m is the number of memberships of the overlapping nodes.

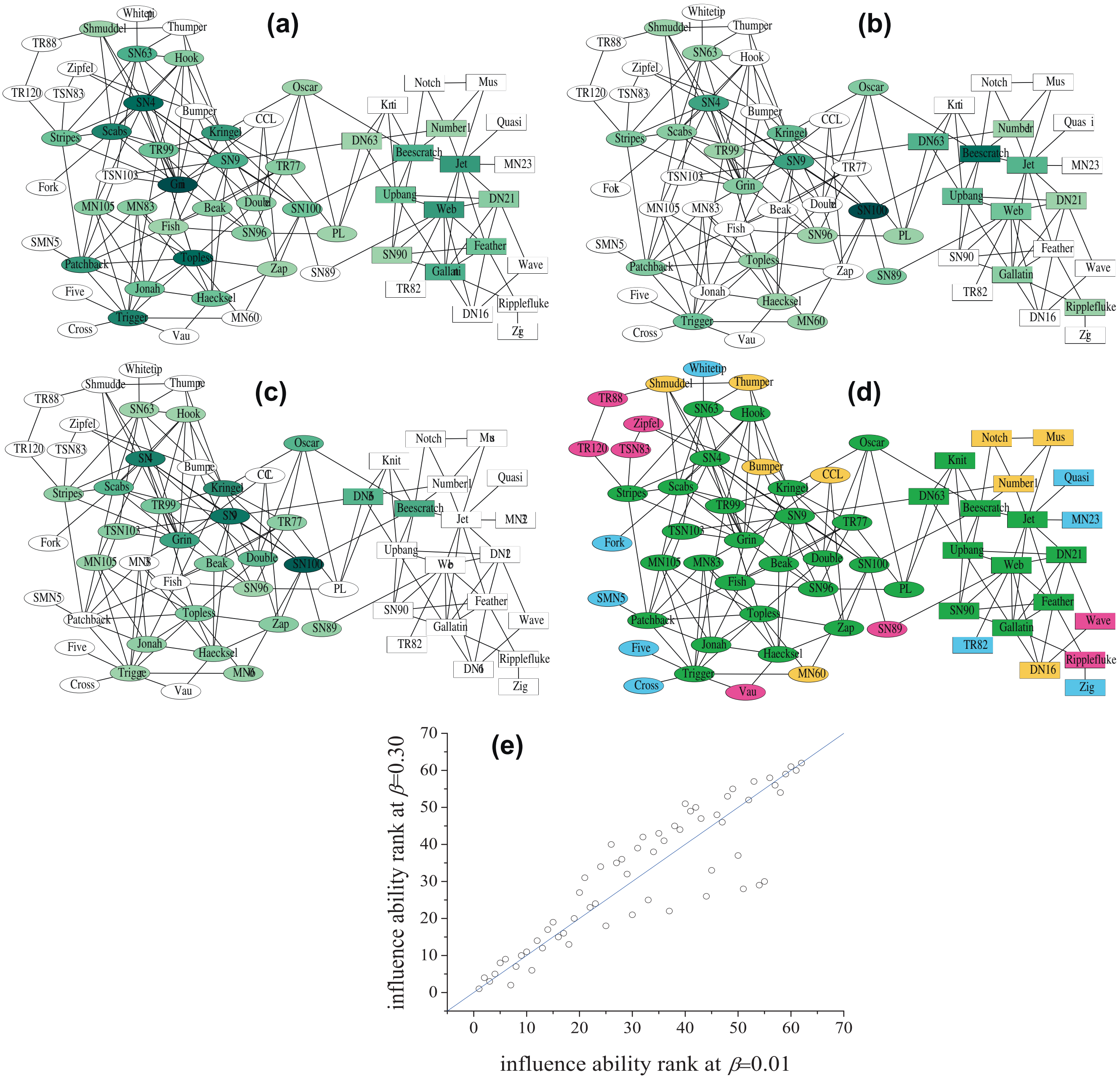

Fig. 1. Lusseau's network of bottlenose dolphins. The partition by shapes matches the biological classification of the dolphins proposed by Lusseau et al. [25]. (a) The colors (from dark to light green) indicate the degree centrality level (from high to low) of the nodes. (b) The colors (from dark to light green) indicate the betweenness centrality level (from high to low) of the nodes. (c) The colors (from dark to light green) indicate the closeness centrality level (from high to low) of the nodes. (d) The colors indicate the shells of k-shell decomposition, and green nodes are the core of the network. (e) The ranks of the influence ability $\sigma(\{u\})$ (Eq. (2)) for each node $u \in V$ at diffusion probability $\beta = 0.01$ and $\beta = 0.30$ in the dolphin network. Each point in the plot corresponds to a node in the network. (For interpretation of the references to color this figure legend, the reader is referred to the web version of this article.)

Lancichinetti et al. provide the corresponding software package of the LFR models [16]. In this study, we use this package to generate artificial networks.

3. Related work

3.1. Centrality-based method

In complex network literature, researchers define many centrality measures on nodes and links [17,18]. In networks with a broad degree distribution, it is believed that the most connected people (hubs) are the key players, being responsible for the most extensive spreading process [19,20]. Degree centrality, however, is a local measure. The global network structure often plays a more important role such that there are plausible circumstances under which the highly connected nodes have little effect on the range of a given spreading process [7]. In Fig. 1a, the hub of node "trigger" exists at the periphery of the network, so it will have a minimal impact on the spreading process. Furthermore, if a network with community structure has unbalanced community sizes, which is pervasive in reality [21], the high degree nodes may lie in the same community with larger size. In Fig. 1a, the top 5 highest degree nodes all lie in the same community such that they can only influence nodes in the same community when the propagation probability β is low.

In the social network theory context, the importance of a node for spreading is often associated with betweenness centrality, a measure of how many shortest paths cross through this node, which is believed to determine who has more interpersonal influence on others [22,23]. The betweenness centrality of node u is defined as $b_u = \sum_{i,j}\sigma(i,u,j)/\sigma(i,j)$, where $\sigma(i,u,j)$ is the number of shortest paths between nodes i and j that pass through node u, and $\sigma(i,j)$ is the total number of shortest paths between i and j. In networks with community structure, nodes with high betweenness often serve as intermediaries in spreading information between communities such that they tend to lie in the junction of communities. Furthermore, in networks with unbalanced community sizes, nodes with high betweenness are more likely to locate in the community with larger size. In Fig. 1b, the two nodes with highest betweenness lie in the junction of the two communities, and most nodes with high betweenness lie in the community with larger size. Therefore, the nodes with high betweenness are pivotal to information transfer between communities, rather than as initial information spreaders. Similarly, closeness centrality, which can be regarded as a measure of how long it will take to spread information from a node to all other nodes sequentially [24], may highlight the nodes at the junction between communities or in the community with larger size (Fig. 1c).

Kitsak et al. [7] propose identifying the core and periphery of a network using the k-shell decomposition method (Appendix A). They show that in many real-world complex networks, the best spreaders do not correspond to the nodes with a high degree and betweenness; the most efficient spreaders are instead those located within the network core, as identified by k-shell decomposition. The k-shell decomposition method, however, does not always work well. In Fig. 1d, the nodes in the core occupy a large proportion of the network such that the influential nodes cannot be identified. In networks with unbalanced community density, the core identified by k-shell decomposition may lie only in the community with high community density. Moreover, the k-shell decomposition method is only suitable for unweighted and undirected networks.

For the methods of degree, betweenness, closeness centrality and the k-shell decomposition, there are two common limitations. First, they may adapt to identify the original influential spreaders only when the spreading originates in a single active node. For a spreading process originating in many active nodes simultaneously, spreading origins located at a particular distance from one another must be considered; otherwise, the nodes influenced by the origins may greatly overlap. These classic methods may identify influential nodes that do not locate far enough. We therefore cannot select the k-node set directly using the centrality measures or the k-shell index. Second, these methods solely focus on network topology structures. On the one hand, they do not account for spreading mechanism (e.g., the IC model) that can affect the spreading effects as spreading process depends on spreading mechanism. This may be the reason why the centrality measures perform worse than the greedy algorithm (Section 3.2) [6] that accounts for the spreading mechanism directly in identifying influential nodes. On the other hand, the propagation probability β used in the spreading process is not considered. We analyze the effects of β on influence ability ranks of nodes, which is measured by ranking $\sigma(\{v\})$, for all $v \in V$ (Eq. (2)) in a ascending order. The results show that the influence ranks of nodes change greatly at different propagation probabilities under the IC model (Fig. 1e).

3.2. Greedy algorithm

Kempe et al. [6] propose a simple greedy algorithm to find k influential nodes set A_k, that approximates the maximization of the expected number of active nodes $\sigma(A_k)$ (Eq. (2)) at the end of the spreading process of the IC model. This method uses a hill climbing strategy and chooses the nodes with maximal marginal gain, presented in Algorithm 2.

Algorithm 2. Greedy Algorithm

```
Input: $G = (V,E,W),\beta,k$ /* $G$ is the network; $\beta$ is a designated
   propagation probability; $k$ is the number of target
   influential nodes*/
Output: $A_k$ /*$A_k$ is the identified $k$ influential nodes set*/
$A_k := \phi$;
for ($i$:=1 to $k$){
    let $v$ be a node that maximizes the marginal gain $\sigma(A_k \cup \{v\}) - \sigma(A_k)$;
    $A_k := A_k \cup \{v\}$;
}
output $A_k$;
End Algorithm
```

Kempe et al. [6] demonstrate that the simple greedy algorithm is a $(1 - 1/e - \epsilon)$ of the optimal result in the IC model; here e is the base of the natural logarithm and ϵ is any positive real number. Thus, this is a performance guarantee slightly better than 63%. In addition to their provable guarantees, the approximation algorithms significantly outperform node selection heuristics based on degree and distance centrality. In this study, we compare the greedy algorithm with our proposed method.

4. *k*-medoid method

In this study, we propose identifying k influential nodes using the k-medoid method. First, we construct an $N \times N$ information transfer probability matrix M on network $G = (V,E,W)$, where N is the number of nodes in V. Element m_{uv} of M denotes the information transfer probability through all paths from node u to v. Second, k medoids are identified as k influential nodes by applying the

k-medoid clustering algorithm [26] on M that can be regarded as a similarity matrix.

4.1. Information transfer probability matrix

The information transfer probability matrix M is computed based on a bond percolation process. A representative question of bond percolation process is about liquid percolation [27]. Assume that some liquid is poured on top of some porous material. Will the liquid be able to make its way from hole to hole and reach the bottom? This physical question is modeled mathematically as a three-dimensional network of vertices in which the links (bonds) between each two neighbors may be open (allowing the liquid through) with probability p, or closed with probability $1-p$, and they are assumed to be independent. Therefore, for a given p, what is the probability that an open path exists from the top to the bottom? This problem, called now bond-percolation, was introduced in the mathematics literature by Broadbent and Hammersley [28], and has been studied intensively by mathematicians and physicists since.

The IC model on network $G = (V,E,W)$ can be exactly mapped on to a bond percolation process in the same network [6,8,29,30]. Each link (u,v) of E is randomly designated either “open” with probability β_{uv} (Eq. (1)) or “closed” with probability $1-\beta_{uv}$ independently. For information diffusion on the network, open links represent the ones through which information propagates, and closed links represent the ones through which information does not propagate. We use the notations proposed by Kimura et al. [8] to describe a bond percolation process on network. A bond percolation process corresponds to an L-dimensional vector $r \in R_G$,

$$R_G = \{r = (r_{uv})_{(u,v)\in E} \in (0,1)^L\}, \tag{5}$$

where L is the number of links in G; $r_{uv} = 1$ if $(u,v) \in E$ is designated “open”, or $r_{uv} = 0$ if it is designated “ closed”.The probability distribution $q(r)$ of the corresponding bond percolation mode is given thus:

$$q(r) = \prod_{(u,v)\in E} \{\beta_{uv}^{r_{uv}}(1-\beta_{uv})^{1-r_{uv}}\}, r \in R_G, \tag{6}$$

where β_{uv} is the propagation probability through link (u,v) in the IC model.

Let E_r denote the set of all the open links for $r \in R_G$, and let G_r denote the graph (V,E_r). For each $r \in R_G$, we can consider the deterministic diffusion model on G_r such that information can transfer from node u to v if v is reachable from u on G_r. We propose that the information transfer probability m_{uv} from node u to v is defined thus:

$$m_{uv} = \frac{1}{n}\sum_{i=1}^{n} \omega(u,v;G_{r_i}), r_i \in R_G, \tag{7}$$

where r_i denotes the ith independent sampling from $q(r)$, and $\omega(u,v;G_{r_i}) = 1$ if v is reachable from u on G_{r_i}, 0 otherwise, and n is the number of sampling.

The process of computing information transfer probability matrix is described by Algorithm 3. The main task of computing M is to find reachable pairs of nodes in G_r. For an undirected network, finding reachable pairs can be based on the connected components in G_r that can be searched through the breadth-first search (BFS) algorithm [31]. The nodes in each connected component are reachable each other; and the nodes between connected components are unreachable. For a directed network, finding reachable pairs can be executed in two steps. First, the strongly connected components in G_r can be searched through Tarjan’s algorithm [32]. Inside each strongly connected components, the nodes are reachable each other. Second, the connectivity between the strongly connected components are judged. If strongly connected component C_i can reach C_j, all nodes in C_i can reach the nodes in C_j.

Algorithm 3. Computing transfer probability matrix

```
Input: $G, \beta$ and $n$ /*$G$ is the network; $\beta$ is a designated
  propagation probability; $n$ is the number of sampling*/
Output: $M$ /*$M$ is the information transfer probability matrix*/
Initialize: set each element in $M$ to 0;
for ($i$:=1 to $n$)
{
  select $r \in R_G$ randomly according to $q(r)$;
  for (each pair of nodes $u$ and $v$, $v$ can be reachable from $u$ in
  $G_r$)
    $m[u,v]$:=$m[u,v]$ + 1;
}
for (each element $m[u,v]$ in $M$)
  $m[u,v]$:=$m[u,v]/n$;
Output $M$;
End Algorithm
```

4.2. k-medoid clustering algorithm

The k-medoid algorithm is a classical partitioning clustering technique that clusters a data set of n objects into k clusters known a priori. A medoid can be defined as the object of a cluster, whose average similarity to all objects in the cluster is maximal, i.e., it is a most centrally located point in the cluster. We use the most common realization of k-medoid clustering algorithm, Partitioning Around Medoids (PAM) [26], to find k influential nodes (Algorithm 4).

Algorithm 4. k-Medoid clustering algorithm

```
Input: $M$ and k /*$M$ is the information transfer probability
  matrix; $k$ is the number of target influential nodes*/
Output: $A_k$ /*$A_k$ is the identified k influential nodes set*/
Initialize: randomly select k of the $N$ nodes as initial medoids;
repeat
{
  associate each remaining node to the closest medoid; /*
  “closest” here is measured by information transfer
  probability in $M$ */
  for (each medoid $m$)
    for (each non-medoid node $o$)
      swap $m$ and $o$ and compute the total benefits of the
  configuration; /* “benefits” here is defined as the sum of
  information transfer probability from the medoid nodes to
  their non-medoid nodes */
  select the configuration with the highest benefits to form
  new set of medoids;
} until no change;
Output $A_k$;
End Algorithm
```

4.3. Computation complexity

The k-medoid method is composed of two parts, computing transfer probability matrix and k-medoid clustering. The main task of computing transfer probability matrix M is to find reachable pairs of nodes in G_r based on connected components or strongly connected components. For an undirected network, the time complexity of finding connected components through the BFS algorithm is $O(N_{G_r} + L_{G_r})$, where N_{G_r} and L_{G_r} denote the number of nodes and links in G_r separately. For a directed network, the time complexity of finding strongly connected components through Tarjan's algorithm is also $O(N_{G_r}+L_{G_r})$. The space complexity of computing transfer probability matrix is $O(N^2)$ in the worst case when all elements of information transfer matrix M are computed. When large-scale networks are sparse and the propagation probability β is small, however, the links in E_r occupy a small proportion of the links in E. Only the pairs of nodes that can reach in E_r must compute the information transfer probability. We analyze the space complexity of the information transfer probability matrix by computing the ratio of nonzero elements in M for different size networks generated by the LFR model (Fig. 2). The propagation probability β affects the ratio significantly. For the different size networks, the ratio approaches 0 as $\beta = 0.005$ and the ratio equals 1 as $\beta = 0.05$. The average degree $\langle k\rangle$ also has positive effects on the ratio. For the different size networks, however, the ratios are greatly different as $\langle k\rangle$ grows bigger. The ratio of the network with 10,000 nodes, for example, is about 10 times larger than the ratio of the network with 100,000 nodes. The mixing parameter μ also has positive effects on the ratio. But as $\mu = 0.7$, the ratio for the network of $N = 100{,}000$ is still very low (approximately 0.0048).

The complexity of each iteration in PAM is $O(k(N-k)^2)$, where k is the number of medoids and N is the number nodes in network. For large values of N and k, such computation becomes very costly. To deal with larger data sets, a sampling-based method, called CLARA [33], can be used. The idea behind CLARA is as follows: Instead of taking the whole set of data into consideration, a small portion of the actual data is chosen as a representative of the data. Medoids are then chosen from this sample using PAM. If the sample is selected in a fairly random manner, it should closely represent the original data set. The representative objects (medoids) chosen will likely be similar to those that would have been chosen from the whole data set. CLARA draws multiple samples of the data set, applies PAM on each sample, and returns its best clustering as the output. As expected, CLARA can deal with larger data sets than PAM. The complexity of each iteration now becomes $O(ks^2+k(N-k))$, where s is the size of the sample, k is the number of medoids, and N is the total number of nodes in the network.

5. Experiments

In this study, we compare the k-medoid method with the greedy algorithm on some real networks, including the bottlenose dolphin network [25], the network of Zachary's karate club [34] and an e-mail network [35]. Moreover, we use some artificial networks generated by the LFR model [13,14]. In this study, when the average number of active node $\sigma(A)$ (Eq. (2)) and the adopting probability $IP(A)$ (Eq. (4)) are computed, the number of spreading simulation is 100,000 for all experiments. When the average scope of active nodes $\varphi(A)$ (Eq. (3)) is computed, 100 groups of experiments are executed and each group contains 1,000 independent spreading simulations.

5.1. Real networks

In real networks with community structure, the greedy algorithm may choose influential nodes in the community with larger size or higher density such that some nodes in other communities can adopt the information from the influential nodes with a probability IP of approximate zero. Although in theory the expected IP should be greater than zero, the experimental results of IP can be zero as the number of spreading simulations is finite. In Fig. 3a, the two influential nodes identified by the greedy algorithm are all located in the community with larger size such that most nodes

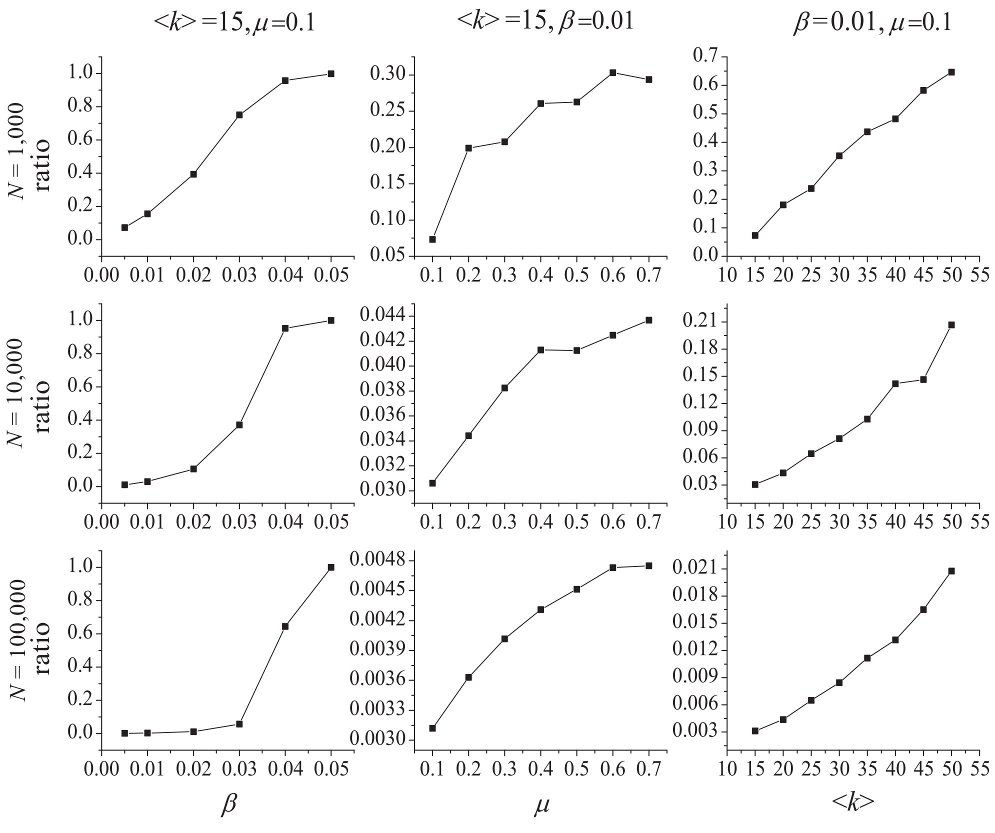

Fig. 2. Space complexity of computing the transfer probability matrix M. The number of spreading simulations is 10,000 when M is computed.

in another community cannot adopt information in the propagation process of IC model with the two influential nodes as initial active ones. In the same experiment, except for the propagation probability $\beta = 0.5$, the two influential nodes identified by the greedy algorithm are " Grin" and "Ripplefluke" . Node "Ripplefluke" lies in the periphery of the network and has only two links to the rest of the network; it is thus not robust for this node to spread information to other nodes. Similarly, in Fig. 3c, the four influential nodes identified by the greedy algorithm are located only in two of four communities.

The k-medoid method tends to choose a representative influential node from each community in these real networks. In Fig. 3b, the two influential nodes identified by the k-medoid method are located in the two communities separately such that all nodes in the network can be infected with $IP > 0$. Even when the influence probability β is set to 0.5, the identified influential nodes do not change. In Fig. 3c, the four influential nodes identified by the k-medoid method lie in the four communities separately.

In this study, we also use two other measures, $\sigma(A)$ (Eq. (2)) and $\varphi(A)$ (Eq. (3)), to compare the greedy algorithm and the k-medoid method. The first line of Fig. 4 compares the greedy algorithm and the k-medoid method on the dolphins network at different propagation probability β. For different β values, the average number of active nodes for the greedy algorithm and k-medoid method are close. The average active scopes of the k-medoid method are higher than those of the greedy algorithm. These results illustrate that in this case the k-medoid method can cover a larger active scope than the greedy algorithm on the premise of not reducing the average active nodes. The k-medoid method can also do a good job in the e-mail network without an obvious community structure (the second line of Fig. 4).

5.2. Artificial networks

In this study, we generate four artificial networks through the LFR models to compare the greedy algorithm and the k-medoid method. (1) LFR-SU: an unweighted and undirected network (the number of nodes is 200; the average degree is 6; the maximum degree is 30; the degree distribution exponent is -2; the community size distribution exponent is -1.2; the community size is between 20 and 60; the mixing parameter is 0.05). (2) LFR-LU: an unweighted and undirected network (the number of nodes is 1000; the average degree is 15; the maximum degree is 50; the degree distribution exponent is -2; the community size distribution exponent is -1.2; the community size is between 20 and 100; the mixing parameter is 0.1, 0.2 or 0.3). (3) LFR-WD: a weighted and directed network (the number of nodes is 1000; the average degree is 15; the maximum degree is 50; the degree distribution exponent is -2; the community size distribution exponent is -1.2; the community size is between 20 and 100; the mixing parameter is 0.1; the mixing parameter for the link weights is 0.1, 0.2 or 0.3). (4) LFR-WO: a weighted and directed network with overlapping communities (the number of nodes is 1000; the average degree is 15; the maximum degree is 50; the degree distribution exponent is -2; the community size distribution exponent is -1.2; the community size is between 20 and 100; the mixing parameter is 0.1; the mixing parameter for the link weights is 0.1; the number of overlapping nodes is 100, 300, 500; the number of memberships of the overlapping nodes is 2).

For the LFR-SU network, the the greedy algorithm chooses influential nodes only from four of six communities such that some nodes in the communities having no influential nodes can only adopt the information with a probability of approximate zero (Fig. 5a). The k-medoid algorithm, however, identify an influential node from each community (Fig. 5b).

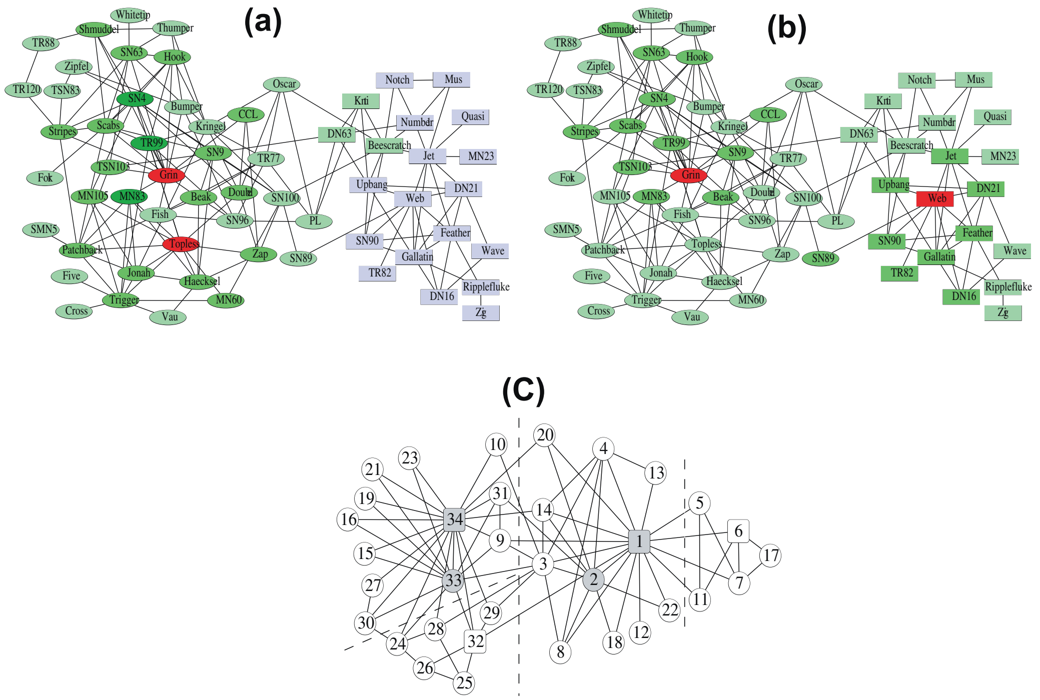

Fig. 3. Comparison of the greedy algorithm and the k-medoid method. (a) and (b) the spread of influence for the greedy algorithm and the k-medoid method separately in the bottlenose dolphin network. The red color denotes identified influential nodes, and the green colors (from dark to light green) indicate the adoption probability level, IP (Eq. (4)), (from high to low) of the nodes, and blue denotes that $IP = 0$. (c) The network of Zachary' karate club [34]. The partitions separated by dash lines correspond to the best partition found by optimizing the modularity from Newman and Girvan [36]. Grey denotes the influential nodes identified by the greedy algorithm, and the square shape denotes the influential nodes identified by the k-medoid method. For (a–c), the diffusion probability $\beta = 0.01$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

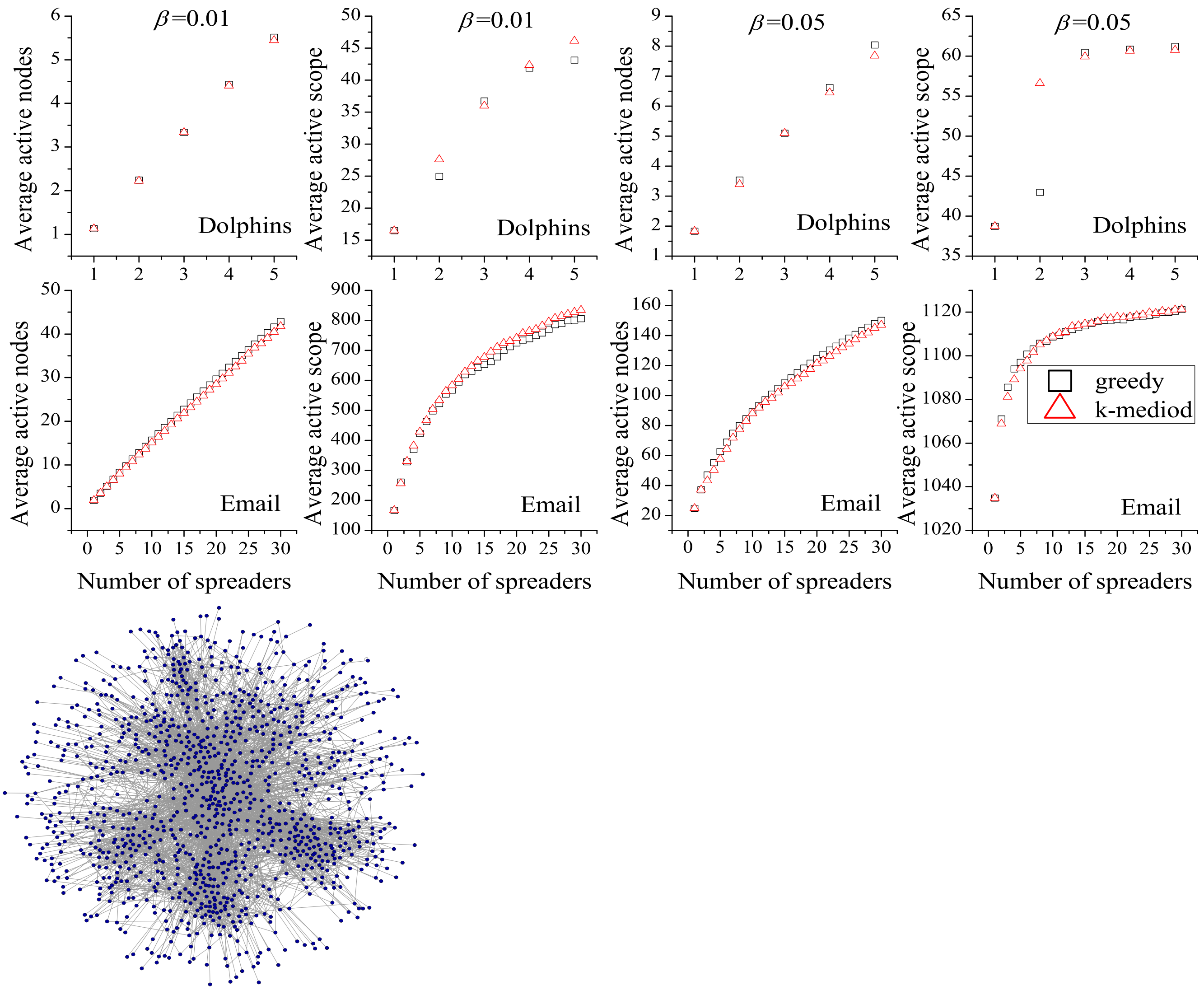

Fig. 4. Comparison of the greedy algorithm and the *k*-medoid method on two real networks, the bottlenose dolphin network and the e-mail network [35] that has no obvious community structure. The figure in the third line is the skeleton of the e-mail network.

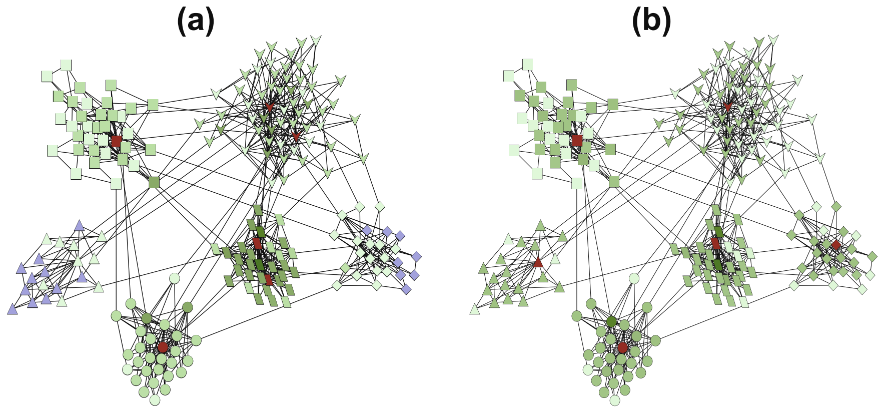

Fig. 5. Comparison of the greedy algorithm and *k*-medoid method on the LFR-SU network generated by an LFR model [13]. The built-in partitions are represented by shapes. The red color denotes identified influential nodes, and the green colors (from dark to light green) indicate the adoption probability level, *IP* (Eq. (4)), (from high to low) of the nodes, and blue denotes that *IP* = 0. (a) The spreading effects of the greedy algorithm. (b) The spreading effects of the *k*-medoid method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 6 compares the spreading effects of the greedy algorithm and *k*-medoid method on the LFR-LU network through the average active nodes $\sigma(A)$ (Eq. (2)) and the average active scope $\varphi(A)$ (Eq. (3)) at different propagation probability β (Eq. (1)) and the mixing parameter μ (see Section 2.3). The smaller value of μ means that there are more links inside communities and less links between

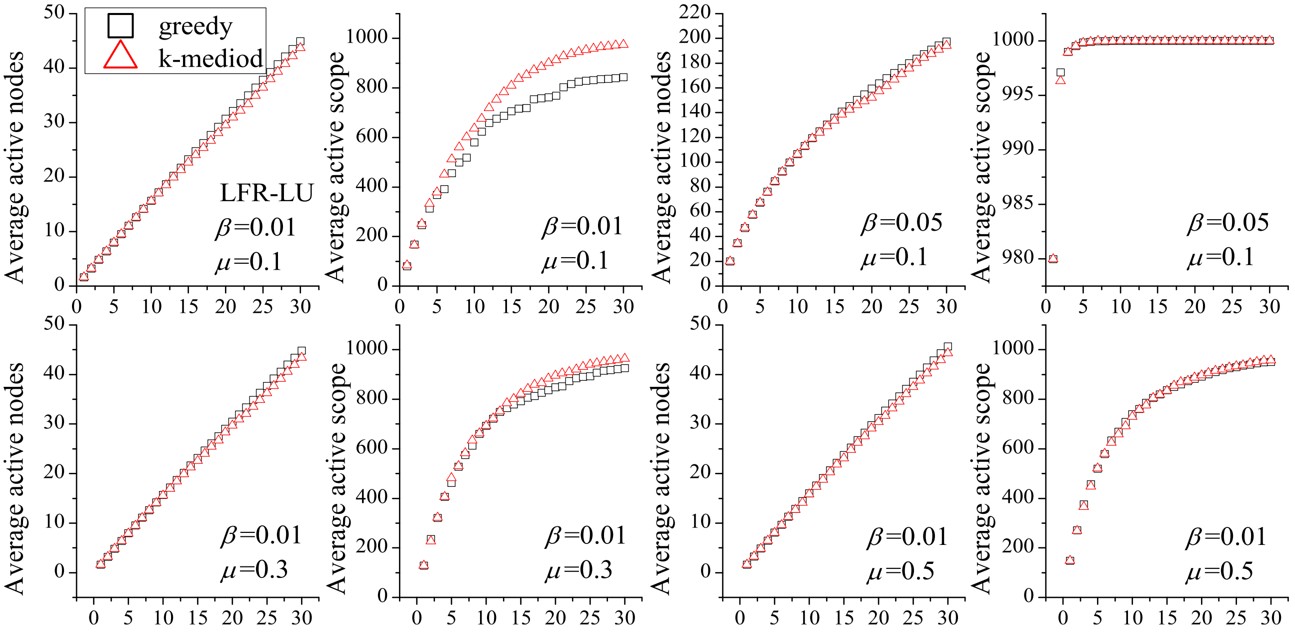

Fig. 6. Comparison of the greedy algorithm and the *k*-medoid method on the LFR-LU network.

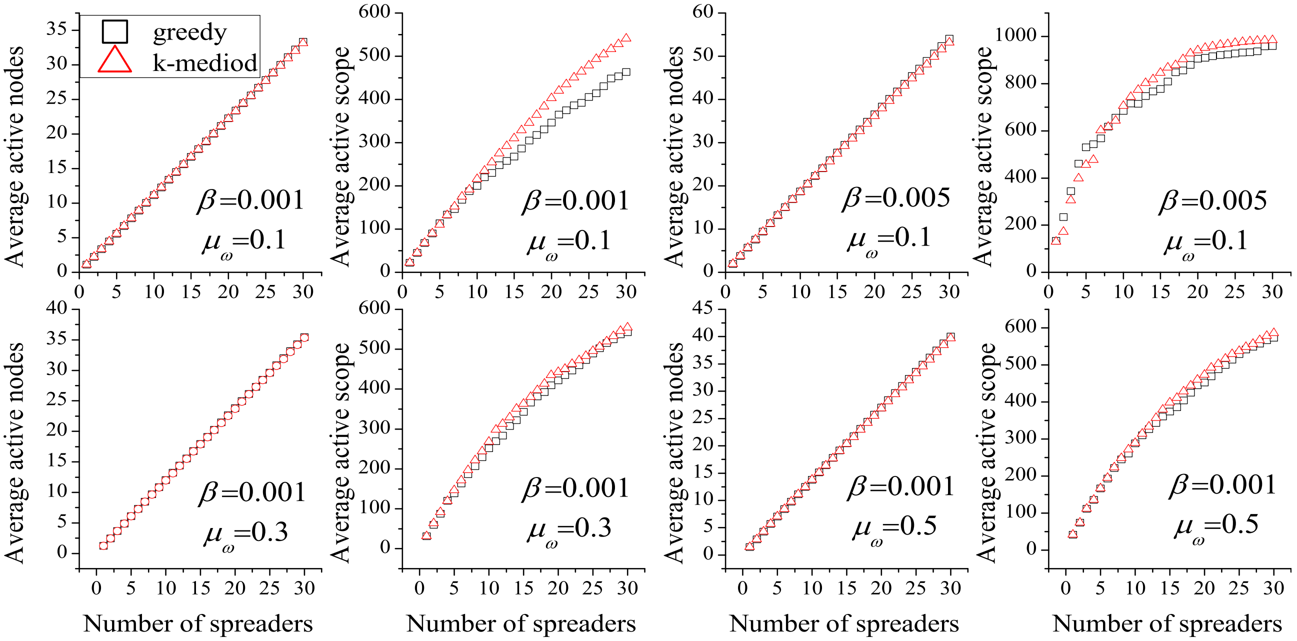

Fig. 7. Comparison of the greedy algorithm and the *k*-medoid method on the LFR-WD network.

communities, i.e., the network has more obvious community structure. For all different β values and μ values, the number of average active nodes for the greedy algorithm and the *k*-medoid method are close. The average active scopes of the *k*-medoid method are higher than those of the greedy algorithm, but as the β and μ values increase, the differences become small. These results illustrate that in the unweighted and undirected network with obvious community structure and small spreading probability, the *k*-medoid method can cover a larger scope than the greedy algorithm on the premise of not reducing the number of average active nodes.

Fig. 7 compares the spreading effects of the greedy algorithm and the *k*-medoid method on the LFR-WD network at different propagation probability β (Eq. (1)) and the internal strength parameter μ_w (see Section 2.3). The smaller value of μ_w means that the internal strength inside communities are larger and the external strength between communities are smaller, vice versa. Thus, smaller μ_w value represents more obvious community structure. For different β values and μ_w values, the number of average active nodes of the greedy algorithm are close to ones of the *k*-medoid method, and the average active scopes of the *k*-medoid method are higher than those of the greedy algorithm. With the increase of β and μ_w values, however, the differences of the average active scopes between these two methods become small. These results illustrate that in the weighted and directed network with obvious community structure and low spreading probability, the *k*-medoid method can cover a larger scope than the greedy algorithm.

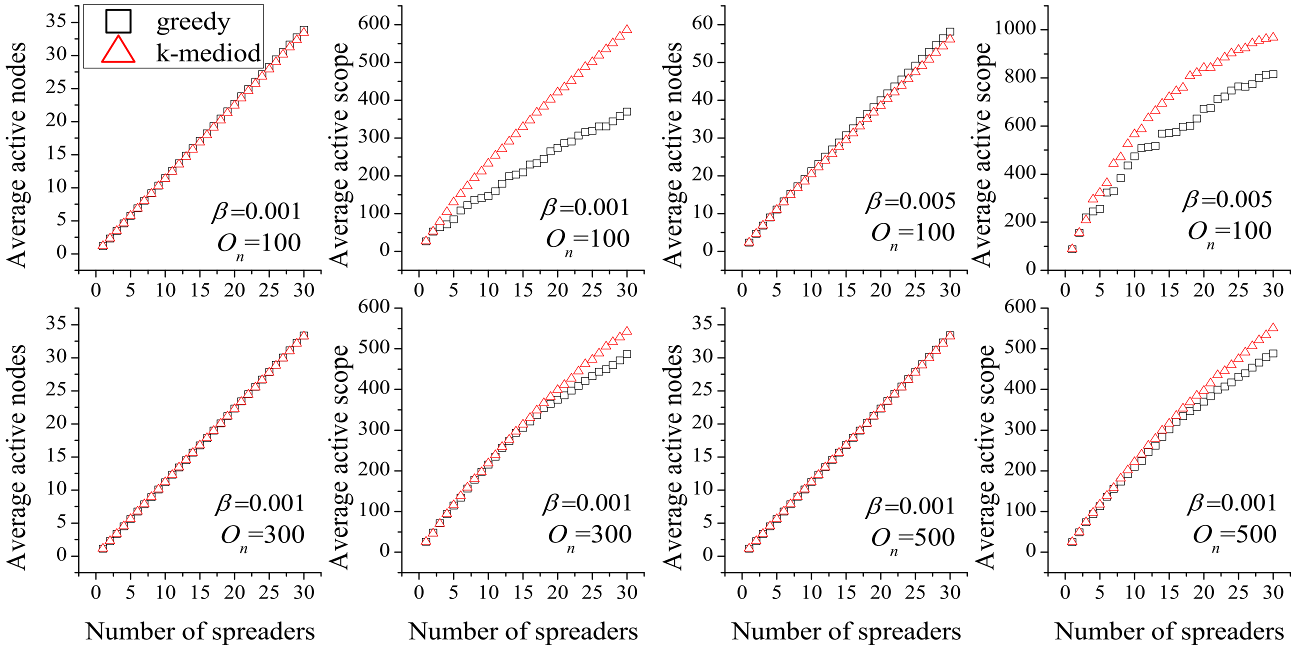

Fig. 8. Comparison of the greedy algorithm and the *k*-medoid method on the LFR-WO network.

Fig. 8 compares the spreading effects of the greedy algorithm and the *k*-medoid method on the LFR-WO network at different propagation probability β (Eq. (1)) and overlapping parameter O_n (see Section 2.3). The smaller value of O_n means that there are less overlapping nodes between communities , i.e., the network has more obvious community structure. For all different β values and O_n values, the number of average active nodes of the greedy algorithm and *k*-medoid method are close. The average active scopes of the *k*-medoid method are higher than those of the greedy algorithm, but with the increase of β and O_n values, the differences become small. These results illustrate that in the weighted and directed network with obvious community structure and low spreading probability, the *k*-medoid method can cover a larger scope than the greedy algorithm on the premise of not reducing the number of average active nodes.

6. Discussion

Finding a small subset of influential nodes that can spread maximal influence to others in a complex network is an important research problem. It can highlight many applications. Though some heuristic methods, including the degree [19,20], betweenness [22,23], closeness [24] and *k*-shell decomposition methods [7], account for local or global topological network structures and have clear definitions and low computation costs (the computation cost of betweenness centrality is high), they have limitations. First, these methods may identify influential nodes in the communities with large size as the nodes in large communities are sometimes more likely to have high centrality level, such that the identified influential nodes can only influence nodes in the same community when the propagation probability is small. Second, these methods often cannot adapt to identify multiple influential nodes because the influential nodes identified by these methods may locate not far enough and the nodes they influence overlap greatly. Third, these methods do not account for spreading mechanism and propagation probability that can affect the spreading effects as spreading process depends on the two factors.

The greedy algorithm [6] depends on the direct solution of the maximization influence problem to find the influential nodes through a spreading model. Though the greedy algorithm has high computation cost, it achieves higher performance of average active nodes than the conventional methods do. For networks with community structure, however, the greedy algorithm may tend to select influential nodes from the community with a larger size or higher density and does not account for the balance between the communities such that some nodes have an influence probability of approximate zero.

The *k*-medoid method has some advantages. On the one hand, the *k*-medoid method employs an information transfer probability, which is computed for any pair of nodes using a spreading model, to measure the similarity between nodes; thus it accounts for both the local and global topological network structures. On the other hand, the *k*-medoid method is essentially a clustering algorithm that is in accordance with community detection. The *k*-medoid method can select influential nodes in communities in a balanced way. Thus, in this study the influential nodes identified by the *k*-medoid method can influence a larger scope than the greedy algorithm on the premise of not reducing the number of average active nodes. The *k*-medoid method, however, differs from the process in which communities are first detected from a

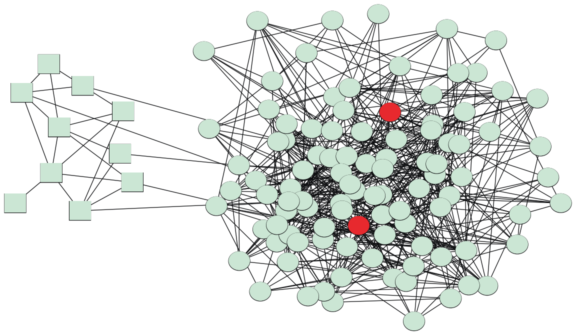

Fig. 9. A computer simulated network with two extremely unbalanced communities. The two red nodes are influential ones identified by the *k*-medoid method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

network and the influential nodes are selected from the detected communities. Community detection is often based on the criterion that there must be more links inside the community than edges linking nodes of the community with the rest of the network [37]. In a network with two extremely unbalanced communities, the two influential nodes identified by the *k*-medoid method may be located in the same community with larger size and higher density (Fig. 9). In this case, selecting an influential node from each community can decrease the spreading efficiency due to the extremely unbalanced size of the two communities. The *k*-medoid method can thus balance community size and influence maximization during the process of identifying influential nodes. The computation cost of the *k*-medoid method mainly depends on the number of nodes in the network, the propagation probability β and the average degree $\langle k \rangle$ of nodes. When β or $\langle k \rangle$ is large, the *k*-medoid method cannot adapt to large-scale networks. Designing an algorithm with low computation costs becomes our future research.

Acknowledgments

This work was supported by the Project of National Science Foundation for Distinguished Young Scholars (70901009), the Program for New Century Excellent Talents in University, the National Basic Research Program of China (2012CB315805), and the Youth Research and Innovation Program in Beijing University of Posts and Telecommunications (2012RC1006).

Appendix A. *k*-Shell decomposition

The *k*-shell decomposition method is often used to identify the core and periphery of networks [7,38]. The *k*-shell decomposition process starts by removing all nodes with only one link, until no such nodes remain, and assigning them to the 1-shell. In the same manner, it recursively removes all nodes with degree 2 (or less), creating the 2-shell. The process continues, increasing *k* until all nodes in the network have been assigned to a shell. The shells with higher indices lie in the network core.

References

[1] S.H. Strogatz, Exploring complex networks, Nature 410 (2001) 268–276.

[2] M.E.J. Newman, The structure and function of complex networks, SIAM Rev. 45 (2003) 167–256.

[3] Y. Kim, H.S. Song, Strategies for predicting local trust based on trust propagation in social networks, Knowl. Based Syst. 24 (2011) 1360–1371.

[4] E.M. Rogers, Diffusion of Innovations, Free Press, New York, 2003.

[5] M.J. Keeling, P. Rohani, Modeling Infectious Diseases in Humans and Animals, Princeton University Press, Princeton, 2008.

[6] D. Kempe, J. Kleinberg, E. Tardos, Maximizing the spread of influence through a social network, in: Proc. Conf. on Knowl. Disc. and Data Min, ACM, New York, Washington, DC, 2003, pp. 137–146.

[7] M. Kitsak, L.K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H.E. Stanley, H.A. Makse, Identification of influential spreaders in complex networks, Nat. Phys. 6 (2010) 888–893.

[8] M. Kimura, K. Saito, R. Nakano, H. Motoda, Extracting influential nodes on a social network for information diffusion, Data Min. Knowl. Disc. 20 (2010) 70–97.

[9] C. Kaiser, S. Schlick, F. Bodendorf, Warning system for online market research – identifying critical situations in online opinion formation, Knowl. Based Syst. 24 (2011) 824–836.

[10] M.P. OMahony, B. Smyth, A classification-based review recommender, Knowl. Based Syst. 23 (2010) 323–329.

[11] M.E.J. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. USA 103 (2006) 8577–8582.

[12] J. Goldenberg, B. Libai, E. Muller, Talk of the network: a complex systems look at the underlying process of word-of-mouth, Mark. Lett. 12 (2001) 211–223.

[13] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Phys. Rev. E 78 (2008) 046110.

[14] A. Lancichinetti, S. Fortunato, Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities, Phys. Rev. E 80 (2009) 016118.

[15] M. Molloy, B. Reed, The size of the giant component of a random graph with a given degree sequence, Combinatorics Probab. Comput. 7 (1998) 295–305.

[16] A Software Package to Generate the Benchmark Graphs. <http://santo.fortunato.googlepages.com/benchmark.tgz>.

[17] T. Opsahl, F. Agneessens, J. Skvoretz, Node centrality in weighted networks: generalizing degree and shortest paths, Soc. Netw. 32 (2010) 245–251.

[18] P. De Meo, E. Ferrara, G. Fiumara, A. Ricciardello, A novel measure of edge centrality in social networks, Knowl. Based Syst. 30 (2012) 136–150.

[19] R. Cohen, K. Erez, D. ben-Avraham, S. Havlin, Breakdown of the Internet under Intentional Attack, Phys. Rev. Lett. 86 (2001) 3682–3685.

[20] R. Pastor-satorras, A. Vespignani, Epidemic spreading in scale-free networks, Phys. Rev. Lett. 86 (2001) 3200–3203.

[21] H. Zardi, L.B. Romdhane, An $O(n^2)$ algorithm for detecting communities of unbalanced sizes in large scale social networks, Knowl. Based Syst. (2012), http://dx.doi.org/10.1016/j.knosys.2012.05.021.

[22] L.C. Freeman, Centrality in social networks: conceptual clarification, Soc. Netw. 1 (1979) 215–239.

[23] N.E. Friedkin, Theoretical foundations for centrality measures, Am. J. Sociol. 96 (1991) 1478–1504.

[24] G. Sabidussi, The centrality index of a graph, Psychometrika 31 (1966) 581–603.

[25] D. Lusseau, K. Schneider, O.J. Boisseau, P. Hasse, E. Slooten, S.M. Dawson, The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations, Behav. Ecol. Sociobiol. 54 (2003) 396–405.

[26] S. Theodoridis, K. Koutroumbas, Pattern Recognition, Elsevier, San Diego, 2009.

[27] http://en.wikipedia.org.

[28] S.R. Broadbent, J.M. Hammersley, Percolation processes, Math. Proc. Cambridge Philos. Soc. 53 (1957) 629–641.

[29] P. Grassberger, On the critical behavior of the general epidemic process and dynamical percolation, Math. Biosci. 63 (1983) 157–172.

[30] M.E.J. Newman, Spread of epidemic disease on networks, Phys. Rev. E 66 (2002) 016128.

[31] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, MIT Press, Boston, 2001.

[32] R.E. Tarjan, Depth-first search and linear graph algorithms, SIAM J. Comput. 1 (1972) 146–160.

[33] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley & Sons, New York, 1990.

[34] W.W. Zachary, An information flow model for conflict and fission in small groups, J. Anthropol. Res. 33 (1977) 452–473.

[35] R. Ruimera, L. Danon, A. Diaz-Guilera, F. Giralt, A. Arenas, Self-similar community structure in a network of human interactions, Phys. Rev. E 68 (2003) 065103.

[36] L. Donetti, M.A. Munoz, Detecting network communities: a new systematic and efficient algorithm, J. Stat. Mech. 2004 (2004) P10012.

[37] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (2010) 75–174.

[38] S. Carmi, S. Havlin, S. Kirkpatrick, Y. Shavitt, E. Shir, A model of Internet topology using *k*-shell decomposition, Proc. Natl. Acad. Sci. USA 104 (2007) 11150–11154.