Soc. Netw. Anal. Min. (2015)5:15
DOI 10.1007/s13278-015-0254-4

ORIGINAL ARTICLE

Modeling influence diffusion to uncover influence centrality and community structure in social networks

Wenjun Wang[1] · W. Nick Street[1]

Received: 22 December 2014 / Revised: 23 April 2015 / Accepted: 29 April 2015

Abstract Node centrality and vertex similarity in network graph topology are two of the most fundamental and significant notions for network analysis. Defining meaningful and quantitatively precise measures of them, however, is nontrivial but an important challenge. In this paper, we base our centrality and similarity measures on the idea of *influence* of a node and exploit the implicit knowledge of influence-based connectivity encoded in the network graph topology. We arrive at a novel influence diffusion model, which builds egocentric influence rings and generates an influence vector for each node. It captures not only the total influence but also its distribution that each node spreads through the network. A *Shared-Influence-Neighbor* (SIN) similarity defined in this influence space gives rise to a new, meaningful and refined connectivity measure for the closeness of any pair of nodes. Using this influence diffusion model, we propose a novel *influence centrality* for influence analysis and an *Influence-Guided Spherical K-means* (IGSK) algorithm for community detection. Our approach not only differentiates the influence ranking in a more detailed manner but also effectively finds communities in both undirected/directed and unweighted/weighted networks. Furthermore, it can be easily adapted to the identification of overlapping communities and individual roles in each community. We demonstrate its superior performance with extensive tests on a set of real-world networks and synthetic benchmarks.

✉ Wenjun Wang
wenjun-wang@uiowa.edu

W. Nick Street
nick-street@uiowa.edu

[1] Department of Management Sciences, University of Iowa, Iowa City, IA, USA

Keywords Social network analysis · Influence diffusion · Community detection · Influence centrality

1 Introduction

This research originates from our attempt to find communities in social networks. Community detection is an important but difficult problem in network analysis and has attracted a great deal of effort from many disciplines. Although a strikingly large number of algorithms have been presented, there are still many open issues. A general but crucial problem is the lack of a quantitatively precise definition of community. While most researchers describe a community as a group of nodes with higher internal than external connectivity, this notion of connectivity is ambiguous and results in many different objective functions and performance metrics. From hierarchical clustering, graph partitioning, and spectral methods to modularity maximization, statistical mechanics, and label propagation, most existing algorithms are rooted in degree or betweenness centrality, edge density, or random-walk-based closeness, etc. While these notions capture the intuition of network connectivity to some extent, the existing literature (Fortunato 2010; Leskovec et al. 2010; Yang et al. 2011) suggests there are still significant areas for improvement. Our goal is to find a more precise measure to decode the connectivity and proximity embedded in the network graph topology and to use this measure to extract community structure.

Our work is motivated by observations from real-world communities. Community members have individual social roles: leaders, core members, liaisons between communities, etc. Some may be even simultaneously associated with multiple communities. Some are more influential than

Published online: 26 May 2015

others, and some are more susceptible to influence. We argue that individual roles, influence, and susceptibility are implicitly embedded in the network topology. In fact, it is influence that not only differentiates individual roles but also acts as the force holding the individuals together to form the community. On the other hand, we notice a simple but meaningful *Shared-Nearest-Neighbor* (SNN) similarity (Jarvis and Patrick 1973) used in traditional clustering, which indicates that two nodes both being close to a set of neighboring nodes suggest they are close to each other. This is naturally extended to our influence-based scenario and can be rephrased as follows: two nodes both influencing a set of (direct and indirect) neighbors suggest they are close to each other. We refer to it as *Shared-Influence-Neighbor* (SIN) similarity, which captures our intuitive notion of community that two nodes both influencing a set of neighbors confirms their closeness and makes them more likely to be in the same community.

All of this makes influence a natural context for the community detection, but the question is how to define a quantitatively precise measure of influence. This leads to another hot topic, namely influence analysis. One fundamental step in influence analysis is to differentiate the relative influence significance among the nodes, where the influence is often characterized using various centralities. The four most widely used measures of centrality in network analysis are *degree* centrality, *closeness* centrality, *betweenness* centrality, and *eigenvector* centrality. Each indicates some specific strength of a node's structural role in the network. However, these measures are not fine enough to quantify a node's comprehensive strength in terms of the total amount of influence one can spread through the network as a seed node. Further, none of them is able to measure the influence-based proximity between pairs of nodes, which as indicated could be a desirable metric for community detection.

We use concepts and techniques from the fields of network modeling, artificial intelligence, and data mining to decode the influence-based connectivity and similarity in network graph topology and arrive at a simple but powerful influence diffusion model. Based upon this model, we propose a novel algorithm for influence ranking, community detection, and role detection in social networks. Our approach naturally incorporates these three important tasks into one integrated framework. Moreover, it can be applicable to not only undirected binary networks, but also directed and weighted networks. Experiments on a set of real-life and synthetic networks show the superior performance of our algorithm.

The rest of this paper is organized as follows. We start with a brief discussion of the related research in Sect. 2, and elaborate our methodology in Sect. 3. Experimental results and performance comparison are shown in Sect. 4. We conclude and point out future work in Sect. 5.

2 Related work

Community detection and influence analysis are essential tasks in network analysis and of great importance in a wide variety of applications. They have received extensive interest and effort from many disciplines. Especially in the field of community detection, a plethora of algorithms have been presented over the years. An in-depth survey can be found in Fortunato (2010), Malliaros and Vazirgiannis (2013) and Xie et al. (2013). We focus here on papers that are most relevant to our concerns and considerations.

2.1 Centralities

Centrality concepts were originally developed and well studied in social network analysis. Centrality refers to the identification of the most important or the most influential nodes in a social work. A set of noteworthy centrality measures were presented and discussed in detail in Wasserman and Faust (1994). There are four widely used centralities that measure the influence significance of a node from different perspectives. *Degree* centrality is a simple but very coarse measure. It has many ties and fails to take into account the influence significance of even the immediate neighbors. *Closeness* centrality (Sabidussi 1966) is defined as the inverse of the sum of lengths of the shortest paths of a node to all other nodes. It measures how fast a node can spread information in the network. *Betweenness* centrality (Freeman 1977) quantifies the number of times a node appears in the shortest path of two other nodes. It can be regarded as a measure of how often a node acts as a broker or gatekeeper of information flow. The closeness and betweenness centralities are both based on the shortest path. However, the spread of information does not always go along the shortest path in reality. To address this issue, researchers have proposed a number of variants, which include *information* centrality (Stephenson and Zelen 1989), *random-walk closeness* (Noh and Rieger 2004), *random-walk betweenness* (Newman 2005), *maximum-flow betweenness* (Freeman et al. 1991), and *current-flow closeness/betweenness* centralities (Brandes and Fleischer 2005).

Eigenvector centrality (Bonacich 1972) is the component of the principal eigenvector of the adjacency matrix, which captures an intuitive but important concept: connecting to a more influential node contributes more influence significance to the node of interest than connecting to a less influential node. Unfortunately, it fails to capture the fact that influence is attenuated when passing through the network. The well-known PageRank (Page et al. 1999) is a variant of eigenvector centrality. For undirected graphs, PageRank degenerates into degree centrality. *Katz* centrality (Katz 1953) is a good generalization of degree centrality and eigenvector centrality plus an attenuation factor associated with the path length. However, for all undirected graphs or directed graphs with cycles, it allows the influence to be transmitted around a loop infinitely. This is not realistic. In addition, its attenuation factor is a user-specified parameter, which makes its influence ranking nondeterministic and less meaningful.

2.2 Connectivity metrics

Due to the lack of a quantitatively precise definition of community, many different objective functions, performance metrics, and corresponding algorithms are presented. From the simplest node degree to the popular modularity (Newman 2006; Newman and Girvan 2004), many connectivity metrics are commonly used in the literature, such as *internal density*, *conductance*, *cut ratio*, *average out-degree fraction* and so on. Leskovec et al. (2010) present an empirical comparison of a range of community-detection algorithms that are based on the above and some other similar metrics. They point out that these intuitive notions of cluster quality tend to fail as one aggressively optimizes the community score and conclude that approximate optimization of community score introduces a systematic bias into the extracted clusters. Another evaluation of various objective functions is proposed by Yang et al. (2011). Their experimental results also cast doubt on the quality of those commonly used connectivity metrics and their corresponding objective functions.

Modularity-based methods have crucial limits as well in spite of their popularity. Guimera et al. (2004) demonstrate random graphs may have partitions with large modularity values due to fluctuations in the edge distribution. Fortunato and Barthelemy (2007) suggest a more fundamental issue, showing that modularity optimization has a resolution limit that may prevent it from identifying well-defined communities below a certain size.

2.3 Similarity measures

Another important class of community-detection algorithms address the problem using various similarity (or closeness) measures. Intuitively, members in the same community are more similar to each other than to the rest of the network. However, defining a meaningful and quantitatively precise similarity measure in connectivity-based graph topology is not straightforward. Superficially, the shortest path between a pair of nodes seems like a direct measure of their distance. Unfortunately, it is not able to differentiate the closeness of nodes in terms of community structure in the network since a single edge can easily link a node deeply located in one community to a node densely connected in another community. An alternative is to consider all paths running between two nodes. Since information can in fact spread along non-shortest paths, Estrada and Hatano (2009) define the communicability of a pair of nodes as the weighted-sum of all the paths connecting them. Since the nodes/edges can be revisited along the walks, the total number of paths is infinite. Consequently, they use the inverse factorial of the path length as the weight to show shorter paths make larger contribution to the communicability than longer ones. They develop a community-detection algorithm using the concept of the communicability graph.

Many sophisticated random-walk-based measures of distance have been proposed. The underlying intuition of these methods is that random walks on a community-structured graph have a much higher chance to get trapped in a community than to travel between communities. Nadler et al. (2005) define a diffusion distance as well as the diffusion map based on the random walk on the graph, which provides novel insight into spectral clustering algorithms. Yen et al. (2009) propose a Euclidean-commute-time distance using the average first-passage time of random walkers on the graph and present an extension of it as a sigmoid commute-time kernel. While the above similarity measures extract the community structure to some extent, their high computational cost usually results in a time complexity of $O(n^3)$ for community detection. Pons and Latapy (2006) propose a well-formulated measure of distance between vertices using their respective probability distributions. Their algorithm (called *Walktrap*) achieves a time-complexity of $O(n^2 \log n)$ in most cases and $O(mn^2)$ in the worst case. In addition, all the above similarity measures are developed based on undirected and unweighted networks. The extension to directed and weighted networks may not be straightforward.

Another interesting approach is to project the graph topology into an n-dimensional Euclidean space so that we can use well-defined and meaningful spatial measures (like Euclidean distance and cosine similarity) and a variety of well-studied clustering methods. The spectral-based similarity proposed by Donetti and Muñoz (2004) is such an example. Each node in the network graph is mapped to a point in a D-dimensional space in which the coordinates are given by its projections on the first D nontrivial eigenvectors. Our SIN similarity (Wang and Street 2014) falls into this category as well, which we elaborate in next section with an extension to weighted networks.

3 Methodology

We draw inspiration from the PageRank algorithm in the sense that we cannot solely rely on the node degree. We have to find an intelligent way to embed influence into a node and pass it around in the network. This leads to a novel influence diffusion model. The influence defined in our diffusion model is different from the influence defined in many other diffusion models such as the popular *independent cascade* model and the *linear threshold* model (Kempe et al. 2003), in which the influence of a node is quantified by the number of inactive nodes it can activate. We assume the influence decays along the path while it is transmitted and measure a node's influence significance by the total amount of influence it spreads out in the network.

Our approach differs from prior work in many ways. From the point of view of centralities, our model extends *degree* centrality from immediate neighbors to multi-step neighborhood, includes not only the shortest paths (that the *closeness and betweeness* centralities rely on) and non-shortest paths, and takes into account both neighbors' influence significance (like *eigenvector* centrality) and influence attenuation (like *Katz* centrality) but without cycling. Our *influence centrality* gives rise to a new, more meaningful, and finer measure of a node's comprehensive strength on diffusing influence in the network as a seed node. Further, we not only find out the total influence a node spreads out, but also keep track of where and how much its influence is distributed in its neighborhood so as to construct its influence vector for community detection.

3.1 Influence diffusion model

The influence in our diffusion model can be interpreted in terms of a piece of message, an idea, an advertisement, or a rumor. Similar to the word-of-mouth communication or storytelling, the message spreads in the network through parallel replication (like a radio broadcast) rather than transfer (one does not lose the message after he forwards it to another person). In fact, those replicas might not be exactly the same as the original message, and they may slightly vary from each other. One important and distinguishing feature in our diffusion model is that anyone who spreads the message needs to sign it such that in case the message circulates back to him he knows he has previously spread that message and will not spread it repeatedly. For example, if person A spreads a rumor to person B, B passes it to person C, and if C passes it back to B and A, both B and A will ignore it after they find out they have already signed it (this mechanism avoids cycles). On the other hand, after the rumor is passed from A to B to C, later when B receives the rumor from person D, he will forward it to both A and C (if B does not see his signature on that rumor), and both A and C will receive it and keep broadcasting it (if they do not see their signatures on it either). From a graph-theoretic point of view, the rumor traverses the network via walks (both nodes and links can be revisited multiple times) but without cycling. Another distinctive feature of our diffusion model is that we put into consideration that the message may lose its effectiveness and fidelity while it is transmitted in the network, and its influence gradually fades away along the diffusion path. We map these features into three important rules in our influence diffusion model as discussed later.

It is noted that the influence in our model is realized and transmitted through *out-links* step by step. Therefore, for a specific real-life network, we need to understand what the link direction represents for in its application. For example, in a citation network, if paper i cites paper j, then the network contains a directed link from node i to node j. However, this directed link does not reflect the direction of the influence propagation since it is actually the cited paper j that influences the citing paper i. So, we need to reverse the citation network to fit into our influence diffusion model. Our model can be applied to both undirected/directed and unweighted/weighted networks. For any undirected link, influence can be transmitted through it in either direction. In other words, if the link between nodes i and j is undirected, we replace it with a directed link from i to j and another directed link from j to i. For weighted networks, we need a *normalization scheme* that fits in the influence-based scenario.

The weight on the link often describes the strength of the relationship between a pair of nodes of interest. It is closely related to the influence. However, the influence of node i with respect to its neighboring node j is not solely measured by the *absolute strength* node i exerts on j. Instead, it is determined by the *relative strength* when compared to the strength of influence node j receives from all other neighbors. In other words, it depends on how *relatively susceptible* node j is to the influence of node i, which leads to a simple but meaningful normalization scheme. Given a directed link pointing from node i to node j and its raw weight w_{ij}, and letting L denote the set of immediate neighbors that point to j (i.e. the in-link neighbors of node j), the proposed *normalized susceptibility weight* is defined as

$$\hat{w}_{ij} = \frac{w_{ij}}{\max_{k \in L} w_{kj}}.$$

As an example, we illustrate in Fig. 1a a simple directed and weighted network. It is noted that the original undirected link between nodes 4 and 5 is replaced with a pair of directed links with the same raw weight in each direction. The corresponding normalized susceptibility weights are shown in Fig. 1b. We use the normalized susceptibility weight on each link to estimate the fraction of influence transmitted from one node to another following the link direction. A desirable property is built in this normalization scheme, that is, it naturally reduces to the unweighted network when all weights are set to 1.

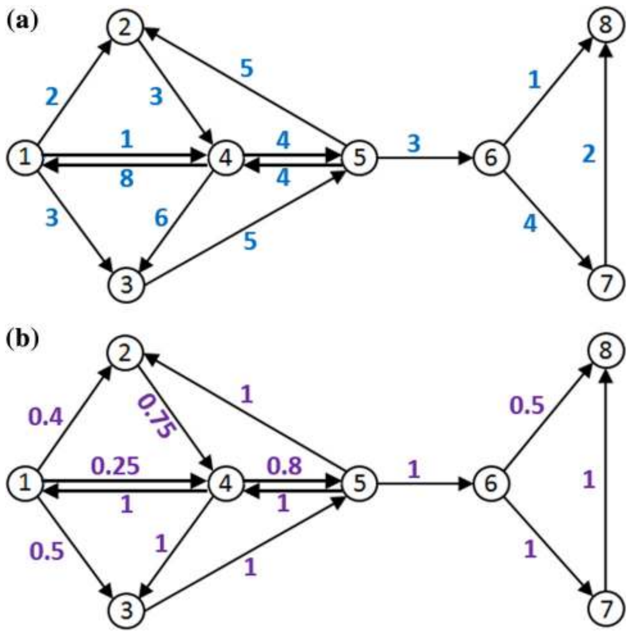

Fig. 1 Example of a directed and weighted network. **a** Raw weights and **b** normalized susceptibility weights

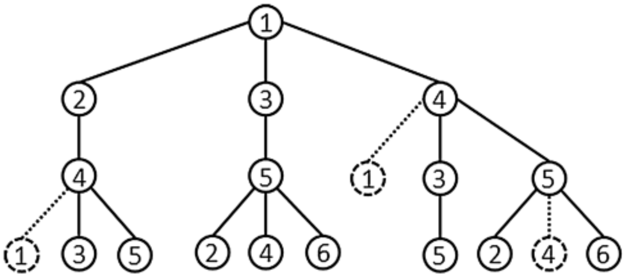

Fig. 2 Egocentric influence rings of node 1

Our influence diffusion model can be regarded as a simple branching process, in which influence originates from a root node and propagates step by step to its offsprings following out-links. We refer to the resultant generation-branching tree as the egocentric influence rings of the root node. There are three important rules implemented in this model:

1. *Cycling is prohibited*. It makes sense since no one should repeatedly exerts influence in cycles in the same round of an influence diffusion process. This distinguishes our model from *Katz* centrality and most random-walk-based algorithms.
2. *Revisits along different routes are allowed and independent*. This is a realistic imitation in the sense that the influence originating from the root node may be delivered to the same person via many different routes independently. This distinguishes our centrality from the closeness and betweenness centralities which only focus on the shortest path.
3. *The farther away from the root, the less influence on arrival*. This is a reasonable assumption that captures the influence locality such as the well-known *3-degree-of-influence* phenomenon (Christakis and Fowler 2007).

Specifically, while the influence propagates along a path, it is attenuated in two independent ways. One is the *weight-associated attenuation*. As discussed above, the normalized susceptibility weight on each link reflects fraction of influence transmitted through the link. Therefore, the weight-associated attenuation of influence from a source node to a destination node is the *product* of the normalized susceptibility weights of the corresponding links that constitute the path. The other is the *depth-associated attenuation*. We draw inspiration from the small-world phenomenon and the concentric scales of resolution around a particular node depicted in Easley and Kleinberg (2010). It is claimed that the probability of a center node linking to a node at a fixed distance d of the ring is proportional to d^{-2}, which fits well in our influence scenario. We, therefore, define the depth-associated attenuation as the *inverse square of the depth* from the root node. The compound attenuation is the product of the weight-associated attenuation and the depth-associated attenuation.

We take the simple network shown in Fig. 1b as an example and illustrate the egocentric influence rings of node 1 in Fig. 2. The first two rules described above are implemented in the construction of the egocentric influence rings. For example, when the influence goes along nodes $1 \rightarrow 4$ and gets back to node 1, this branch flow stops there since no cycling is allowed. In fact, the loop is not even closed, as indicated by the dashed lines in the figure. Similarly, when the influence goes along nodes $1 \rightarrow 2 \rightarrow 4$, the branch flow going back to node 1 stops propagating before getting back to node 1. On the other hand, node 5 is visited 4 times along nodes $1 \rightarrow 3 \rightarrow 5$, nodes $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$, and so on. In addition, following a diffusion path from nodes $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$, the actual influence nodes 2, 4, and 5 acquire from node 1 is 1×0.4, $\frac{1}{2^2} \times 0.4 \times 0.75$, and $\frac{1}{3^2} \times 0.4 \times 0.75 \times 0.8$, respectively.

3.2 Influence matrix

We employ a modified depth-limited search algorithm to explore the egocentric influence rings of each node and generate an influence vector for each node. Let *NSW* denote the normalized susceptibility weight and *IV* denote the influence vector. The pseudocode in Fig. 3 shows how we sweep over all the nodes to build the influence matrix that consists of the influence vectors of all the nodes in the network.

```
Input : Directed/weighted graph $G(V,E)$ with $n = |V|$
        Maximum depth $depthLimit$
Output : Influence matrix $IM$
 1: Calculate the $NSW$ of each edge
 2: Set the influence significance of each node to 1
 3: for node $i$ = 1 to $n$ do
 4:     Empty open/close list
 5:     Set all nodes to be unexplored
 6:     OpenList-PushStack ($Node(i)$)
 7:     Set $Node(i).depth = 0$
 8:     while OpenList is not empty do
 9:         $cNode$ = OpenList-PopStack ( )
10:         $Node(i).IV(cNode.nIndex)$ +=
            $(cNode.depth)^{-2} \times cNode.weight$
11:         Pop all nodes in CloseList with $depth \geq cNode.depth$
12:         Set those nodes to be unexplored
13:         if $cNode.depth < depthLimit$ then
14:             for each out-link neighbor $j$ of
                $Node(cNode.nIndex)$ do
15:                 if $Node(j)$ is unexplored then
16:                     Create a new OpenList node $nNode$
17:                     Set $nNode.nIndex = j$
18:                     Set $nNode.weight = cNode.weight \times$
                        $Node(cNode.nIndex).NSW(j)$
19:                     Set $nNode.depth = cNode.depth + 1$
20:                     OpenList-PushStack($nNode$)
21:                 end if
22:             end for
23:             Set $Node(cNode.nIndex)$ is explored
24:             CloseList-PushStack ($Node(cNode.nIndex)$)
25:         end if
26:     end while
27:     $IM(i) = Node(i).IV$
28: end for
29: Return $IM$
```

Fig. 3 Pseudocode of InfluenceMatrix-Builder

Without loss of generality, our algorithm takes a directed and weighted network and a depth limit as input. It maintains an open list of to-be-explored nodes and a close list of already-explored nodes, both implemented as a stack (Last-In-First-Out). Each node in the open/close list contains an integer variable *nIndex* that denotes its node index in the network and another integer variable *depth* that indicates its depth in the influence rings of the root node. The node in the open list also contains a variable *weight* that stores the product of the normalized susceptibility weights along the influence diffusion path. Each node of the network contains a Boolean variable that indicates whether it has been explored so as to avoid cycling.

The algorithm starts with the calculation of the normalized susceptibility weight of each link using the normalization scheme described above and assigns an initial influence significance of 1 to each node (Lines 1–2). For each iteration, after emptying the open/close lists and setting all nodes to be unexplored, a (root) node is pushed into the open list (Lines 4–7), and then the depth-limited search is explored until the open list is empty. Whenever a node (*cNode*) is popped from the open list (Line 9), we calculate the attenuated influence and accumulate it in the root node's influence vector accordingly (Line 10). Then we pop from the close list all the nodes whose depths are greater than or equal to the depth of node *cNode* and set all of those nodes to be unexplored (Lines 11–12). This is the mechanism that allows revisits from different routes. Then we check whether the depth of node *cNode* is less than the depth limit. If it is, we continue the exploration by pushing to the open list all of the unexplored *out-link neighbors* of *Node(cNode.nIndex)*. For each of them, we create a new open-list node *nNode* that records the node index, current depth, and the chain product of the normalized susceptibility weights (Lines 14–22). Finally, we mark *Node(cNode.nIndex)* as explored and push it into the close list (Lines 23–24). Each iteration of the *For* loop generates an influence vector of a specific node, which contains all the nodes it influences associated with the corresponding influence value. After sweeping over all the nodes, the algorithm creates an influence matrix for the network as a whole.

We also develop a closed form for the influence matrix (up to a depth limit of 3) when the network is unweighted. Let A denote the adjacency matrix of the network (without self-loops). The matrix A^n (i.e., the matrix product of n copies of A) has an interesting interpretation: the entry in row i and column j gives the number of paths of length n from node i to node j. Let D_n denote the diagonal matrix of A^n. The entry d_{ii} is the number of paths of length n for node i to walk to itself. Then the influence matrix M with different depth limit is given as

$$\begin{aligned}
M_0 &= I \\
M_1 &= M_0 + A \\
M_2 &= M_1 + \frac{1}{4}(A^2 - D_2) \\
M_3 &= M_2 + \frac{1}{9}\left[(A^2 - D_2)A - D_3 - AD_2 + A \otimes A^T\right].
\end{aligned}$$

M_0 is the initial assignment of influence significance of 1 to each node at depth 0, where I is the identity matrix. M_1 is simply the first-step influence propagation. In M_2, we avoid two-step cycling by subtracting D_2 from A^2 and then multiply it by 2^{-2}, which is the two-step influence attenuation coefficient. In M_3, we first let all the nodes on the second depth propagate to depth 3, which is represented by $(A^2 - D_2)A$, then subtract the three-step cycling D_3 of the root node and the two-step cycling of all the first-step nodes AD_2. For all those first-step nodes that link to the root node with an undirected link, we remove them twice, one in D_2A and one in AD_2. And so we get one back by adding $A \otimes A^T$, which is the component-wise multiplication of matrix A and its transpose matrix A^T. Finally, we multiply by 3^{-2} to reflect the three-step influence attenuation.

In practice, we do not need to create an $n \times n$ influence matrix. Instead, we store each influence vector in a compact dynamic array that only stores the node index and the corresponding influence value of those nodes influenced by the respective root node. This implementation significantly reduces the space complexity and speeds up the calculation of total influence of each node and the computation of the *Shared-Influence-Neighbor* similarity.

3.3 Influence centrality

As described above, the influence significance of a node is quantified by the total influence it spreads throughout the network. Once the influence matrix is built, it is straightforward to measure the influence significance of each node, which is simply the summation of all the elements in the influence vector (a row vector in the influence matrix) corresponding to each individual node. Let $R(i)$ denote the influence significance of node i and N denote the total number of nodes in the network. Then we can write it as

$$R(i) = \sum_{j=1\ (j \neq i)}^{N} M_{ij}.$$

Since the root node never distributes its own influence to itself, each diagonal element of the influence matrix has a value of 1. It is not included in the calculation of the influence significance even though it does not change the influence ranking. We refer to our influence ranking as *influence centrality*. The pre-specified depth limit is a nice gauge to measure the influence at different scales. When the depth limit is set to 1, our influence centrality reduces to degree centrality.

Further, an important characteristic is hidden in the influence matrix. Let $Row(i)$ and $Column(i)$ denote the influence matrix's ith row vector and ith column vector, respectively. Then $Row(i)$ is the influence vector of node i that describes where and how much influence node i distributes in the network. Interestingly, the column vector $Column(i)$ is exactly a representation of where and how much influence node i acquires from the network. In other words, $Row(i)$ consists of the set of nodes that are influenced by node i, and $Column(i)$ represents the set of nodes that influence node i. The summation of all the elements in $Column(i)$ is the total influence node i receives from other nodes, which could be a good indicator of susceptibility ranking among all the nodes in the network. All of these deserve further analysis in depth as future work.

3.4 Influence-guided spherical K-means (IGSK)

From a geometric perspective, our algorithm projects the graph into an n-dimensional influence space, in which each node defines one dimension. The position of each node in this space is determined by its influence vector. Once the influence matrix is generated, we measure the closeness of any pair of nodes with the cosine similarity of their respective influence vectors. This is a *soft* definition of the *Shared-Influence-Neighbor* (SIN) similarity. More precisely, for a pair of nodes i and j, let V_i and V_j denote their normalized influence vectors, respectively. The *strict* definition of SIN similarity is

$$S_{ij} = V_i(j) \times V_j(i) \ + \sum_{k=1\ (k \neq i,\, k \neq j)}^{N} V_i(k) \times V_j(k).$$

The difference is that in the *strict* definition, the diagonal elements of the influence matrix [i.e. $V_i(i)$ and $V_j(j)$] are not included in the calculation of the SIN similarity (or the normalization of the respective influence vectors). Instead, they are replaced with $V_i(j) \times V_j(i)$, which is actually the mutual influence between nodes i and j. The *strict* definition is more accurate than the *soft* one even though the discrepancy might be negligible in most cases. Now that we have the closeness measure for any pair of nodes in the network, a variety of well-studied clustering algorithms can be applied to find communities. In this paper, we use spherical K-means clustering (Dhillon and Modha 2001). Hence our algorithm is termed as *influence-guided spherical K-means* (IGSK). Since the centroid is basically a virtual node in the cluster, the *strict* SIN similarity is not applicable in this case. Hence, we use the *soft* SIN similarity in IGSK.

We take advantage of the influence ranking in a heuristic way for initializing the cluster centroids. Intuitively, the most influential member of a community has a higher probability to be located in the center area of the community. We first choose the node of the highest influence ranking as the centroid of cluster 1. For the next $(K-1)$ centroids, we choose the remaining node of highest influence ranking and assign it as a centroid of a cluster if its SIN similarity with each of the already-selected centroids is less than a *closeness threshold*. This mechanism significantly improves both the accuracy and the efficiency.

The remaining parameter is the depth limit. Remember that the influence diminishes inversely proportional to the square of the depth. We find that setting the depth limit to 3 is sufficient for community detection. Moreover, for small-size networks or small communities, or when the community structure is fuzzy, setting the depth limit to 2 may have advantages over setting it to 3. In practice, we run IGSK twice by setting the depth limit to 2 and 3, respectively. Between the two resultant partitions, we finalize the cluster assignment with the one of higher modularity (Newman 2004).

3.5 Overlapping community and role detection

Most complex networks exhibit overlapping community structures in which some nodes are characterized with multiple community memberships. In fact, the overlap is a significant feature of various social networks. However, finding overlapping communities or identifying the overlapping nodes is another prominent challenge in the community-detection field. Interestingly, our approach can be easily adapted to the identification of overlapping communities. While IGSK produces a *crisp* assignment (in which each node belongs to one and only one community), it can be naturally turned into a *fuzzy* assignment (in which each node is assigned to each community associated with a *belonging factor*). The belonging factor is a measure of strength of the association of a node to a community.

Let N denote the total number of nodes in the network, K denote the total number of communities, V_i denote the influence vector of node i (without normalization), and C_j denote the set of nodes in community j (based on the crisp assignment of IGSK). Then we define the belonging factor of node i associated with community j as

$$a_{ij} = \frac{\sum_{n \in C_j} V_i(n)}{\sum_{m=1\ (m \neq i)}^{N} V_i(m)}, \quad \forall i \in N, \quad \forall j \in K.$$

This definition has a clear and natural interpretation: the belonging factor a_{ij} represents the ratio of the total influence node i transmits to community j to the total influence it spreads out in the network. Further, with a tunable *belonging threshold*, we can exploit the overlapping community structure at different scales and easily find the overlapping nodes.

In addition, for each community of IGSK *crisp* assignments, we can rank all of its community members by their *internal* influence, *external* influence, and *comprehensive* influence. The *internal* influence is the total influence a node spreads inside its community. In contrast, the *external* influence is the total influence it sends out to other communities. The *comprehensive* influence is the sum of the internal and external influence. These three influence rankings enable us to identify the roles of individual members in each community, such as leaders, core members, and inter-community liaisons.

4 Experiments

To get the preliminary insights and verify the validity of our approach, we perform the centrality analysis using two small real-life social networks and a large citation network. For community detection, we focus on networks with known communities. Since the reliable ground truth for large-scale real-world networks is rarely available, we test our algorithm on several small real-life networks and a large set of synthetic LFR benchmarks (Lancichinetti et al. 2008) and evaluate the performance by comparing with the ground truth and a set of state-of-the-art algorithms.

4.1 Network description

There are six real-life networks: karate club (Zachary 1977), sawmill communication (Michael and Massey 1997), Mexican political power (Gil-Mendieta and Schmidt 1996), dolphin social network (Lusseau et al. 2003), American college football (Girvan and Newman 2002), and an arXiv HEP-TH citation network (Gehrke et al. 2003). All of them are widely used benchmarks for algorithm evaluation.

In order to compare with the set of representative algorithms examined in Lancichinetti and Fortunato (2009a), we generate a set of LFR benchmark graphs using the same parameter settings: *average degree* $= 20$, *maximum degree* $= 50$, *degree distribution exponent* $= -2$, *community-size distribution exponent* $= -1$. There are two different network sizes (1000 and 5000 nodes) and two different ranges for community sizes (S and B). “S” stands for “small”, which means *min/max community size* $= 10/50$. In contrast, “B” stands for “big”, which means *min/max community size* $= 20/100$. In each of the 8 *unweighted* benchmark sets (4 undirected sets and 4 directed sets), we vary the *topological mixing parameter* μ_t from 0.1 to 0.8. Similarly, we generate 4 sets of LFR *weighted* networks using the same parameters described above plus another 2 parameters: *weight-strength distribution exponent* $\beta = 1.5$ and *weight mixing parameter* μ_w. While we vary the weight mixing parameter from 0.1 to 0.8, the topological mixing parameter is set to 0.5 and 0.8, respectively. We generate 5 realizations for each value of the topological/weight mixing parameter, which results in a total of 480 LFR benchmarks.

4.2 Centrality analysis

Figure 4 presents a graphical illustration of the well-known Zachary’s karate club network. The 34 club members split into two groups due to the disagreement between the club instructor (node 1) and the club president (node 34). The orange squares represent members associated with the instructor, and the white circles represent members in the president’s group. Figure 5 shows its global influence ranking in terms of our *influence centrality* scores (*depthLimit* $= 2$). We use a rating scale of 0 to 10, with 10 meaning “most influential”.

As we can see, our *influence centrality* gives the two leaders (nodes 1 and 34) the highest scores and finds a set

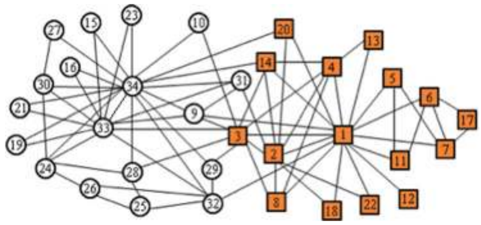

Fig. 4 Zachary's karate club

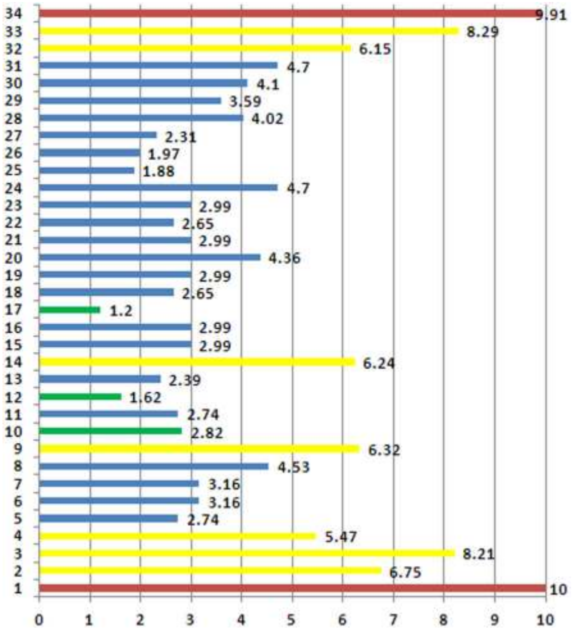

Fig. 5 Influence ranking of Zachary's karate club

of core members (nodes 2, 3, 4, 9, 14, 32, and 33) in the club. One may argue it is expected that they get higher scores simply because of their higher degrees. In fact, it is not that straightforward. For example, although node 4 has a higher degree than node 14, its score is actually lower than that of node 14. One may also notice that nodes 10 and 17 both have a degree of 2, but node 10 has a much higher score than node 17. It makes sense since node 10 connects to nodes 3 and 34, which are much more influential than nodes 6 and 7 to which node 17 connects. Moreover, even though node 12 only has a degree of 1, it also gets a score greater than node 17 because node 12 has a direct connection to the group leader (node 1). It follows our intuition that connecting to a more influential person contributes more influence to the person of interest than connecting to a less influential one. Our *influence centrality* unveils the connectivity-based influence significance in a more detailed manner.

We do more centrality analysis using sawmill communication network. Figure 6 illustrates its network connectivity and ground-truth communities as indicated with different node colors/shapes. We list in Table 1 our *influence* centrality (*depthLimit* = 3, referred to as "Influence" in the table) and compare it against a set of conventional centralities, where "CFC" and "CFB" stand for *current-flow-closeness* and *current-flow-betweenness* centralities (Brandes and Fleischer 2005), respectively. As we can see, the *PageRank* simply degenerates into the *degree* centrality as expected. The *closeness* and *betweenness* centralities are not quantitatively fine or comprehensive enough to differentiate the overall influence ranking. The *closeness* centrality scores of 10 vs. 4.12 do not show the expected large difference in influence significance of HM-1 vs. HP-1 as compared to our *influence* centrality scores of 10 vs. 0.62. The *betweenness* centrality fails to measure the influence significance of 9 employees by giving them a score of zero. These include nodes 15 and 22, who are actually the immediate neighbors of the most influential employee node 12. The drawbacks of the *closeness/betweenness* centralities are inherent in their definitions since they only focus on the shortest path and fail to incorporate the neighboring nodes' influence significance or the attenuation of influence along the diffusion path. The *current-flow closeness/betweenness* centralities do not show much improvement from the conventional ones. Only *eignvector* centrality exhibits the similar ranking pattern as our *influence* centrality.

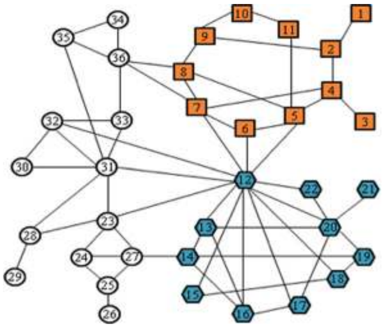

Fig. 6 Sawmill communication network

It is worth pointing out this comparison is not to prove the failure of other centralities. As discussed in Sect. 2.1, we understand each centrality has a different perspective to measure some specific strength of a node's structural role in the network. What we want to show is that our *influence centrality* provides a novel centrality measure that differentiates the nodes' comprehensive significance on influence diffusion in a more meaningful and more precise manner.

We examine its validity in directed networks using the arXiv HEP-TH citation network, which consists of 27,771 papers and 352,807 citations among them. Those papers are in the field of high-energy physics and were added to the e-print arXiv between 1992 and 2003. The first 4 digits

Table 1 Comparison of different centralities on sawmill communication network

Employee	Node	Influence	Eigenvector	Degree	PageRank	Closeness	CFC	Betweenness	CFB
HP-1	1	0.62	0.09	0.77	0.79	4.12	3.25	0.00	0.00
HP-2	2	1.69	0.47	2.31	2.37	5.19	5.27	1.05	1.80
HP-3	3	0.87	0.34	0.77	0.79	4.93	3.74	0.00	0.00
HP-4	4	3.35	1.68	3.08	3.13	6.54	6.70	2.47	2.72
HP-5	5	5.36	3.63	3.85	3.90	8.19	8.01	3.37	3.73
HP-6	6	4.95	3.52	2.31	2.33	7.56	7.32	0.01	1.65
HP-7	7	5.76	3.90	3.85	3.89	8.29	8.28	2.77	3.57
HP-8	8	3.92	2.05	3.08	3.13	6.67	7.47	1.22	2.71
HP-9	9	1.90	0.56	2.31	2.36	5.35	5.83	0.45	2.07
HP-10	10	1.14	0.27	1.54	1.58	5.00	4.74	0.06	0.89
HP-11	11	1.73	0.78	1.54	1.57	5.96	5.12	0.68	1.01
HM-1 (Juan)	12	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
HM-2	13	5.41	4.32	3.08	3.06	7.31	7.53	0.37	1.48
HM-3	14	3.73	2.45	3.08	3.06	6.13	7.33	0.53	2.54
HM-4	15	3.38	2.60	1.54	1.53	6.80	5.93	0.00	0.49
HM-5	16	5.29	4.14	3.08	3.06	7.23	7.47	0.34	1.41
HM-6	17	4.78	3.82	2.31	2.30	7.01	7.00	0.02	0.96
HM-7	18	3.93	2.95	2.31	2.30	6.94	6.69	0.34	1.18
HM-8	19	3.01	2.07	2.31	2.29	5.91	6.71	0.18	1.45
HM-9	20	5.99	4.86	4.62	4.60	7.23	7.90	1.38	2.73
HM-10	21	1.36	0.98	0.77	0.76	5.31	4.09	0.00	0.00
HM-11	22	3.78	2.99	1.54	1.53	6.87	6.16	0.00	0.59
EM-1	23	5.77	4.02	3.85	3.84	8.10	7.96	3.59	4.11
EM-2	24	2.63	1.27	2.31	2.30	6.13	5.88	0.74	1.44
EM-3	25	1.76	0.62	2.31	2.30	4.93	4.99	0.96	1.30
EM-4	26	0.62	0.12	0.77	0.77	3.95	3.14	0.00	0.00
EM-5	27	3.19	1.68	3.08	3.07	6.36	6.68	1.29	2.71
Y-1	28	3.18	1.91	2.31	2.30	6.24	6.13	0.96	1.80
Y-2	29	0.79	0.39	0.77	0.77	4.76	3.56	0.00	0.00
Forester	30	2.74	1.80	1.54	1.54	5.86	5.80	0.00	0.38
Mill manager	31	6.68	5.11	5.38	5.39	8.19	8.60	2.84	4.79
Owner	32	5.16	3.85	3.08	3.08	7.39	7.47	0.51	1.81
Kiln operator	33	3.61	2.23	2.31	2.32	6.60	6.94	0.22	1.62
EP-1	34	1.70	0.75	1.54	1.55	5.48	5.37	0.00	0.57
EP-2	35	2.95	1.61	2.31	2.32	6.60	6.47	0.57	1.64
EP-3	36	3.91	2.12	3.85	3.88	6.80	7.68	1.15	3.51

of each paper ID represent the year and the month when the paper was published online. For instance, paper *9510017* indicates it was published in October of 1995. Table 2 lists the top 10 papers identified by our *influence* centrality (*depthLimit* = 3), *in-degree* centrality, and *PageRank*, respectively. The in-degree centrality is based on the number of citations a paper receives, which is listed in parenthesis in the *In-degree* column. The number listed in parenthesis in the *Influence* column is the in-degree ranking of the corresponding paper. The number listed in parenthesis in the *PageRank* column is the number of citations of the corresponding paper.

Like most centrality measures, our influence centrality is correlated with degree centrality. All the top-10 *influence-centrality* papers have very high in-degrees, which we can tell by their respective in-degree ranking. It includes 7 of the top-10 *in-degree centrality* papers but ranks them in different order. It is hard to rigorously prove our *influence centrality* gives the exact ranking of those papers. But we believe it differentiates the influence ranking in a more

Table 2 Comparison of different centralities on the arXiv HEP-TH citation network

Rank	Influence	Indegree	PageRank
1	9510017 (6)	9711200 (2414)	9402044 (257)
2	9503124 (8)	9802150 (1775)	9205068 (167)
3	9711200 (1)	9802109 (1641)	9205027 (191)
4	9410167 (15)	9407087 (1299)	9207053 (102)
5	9510135 (14)	9610043 (1199)	208020 (205)
6	9802150 (2)	9510017 (1155)	9204102 (71)
7	9802109 (3)	9908142 (1144)	9301042 (344)
8	9610043 (5)	9503124 (1114)	9201019 (16)
9	9407087 (4)	9906064 (1032)	9205081 (77)
10	9601029 (17)	9408099 (1006)	9209016 (76)

meaningful and more precise manner than *in-degree* centrality. As discussed above, the degree centrality is actually a special case of our influence centrality, i.e., setting $depthLimit = 1$. Then it simply counts the number of immediate in-link neighbors of the citation network. When we set $depthLimit = 3$ as we do by default, we explore the whole 3-step neighborhood, in which their neighbors' influence significance is incorporated in the influence diffusion process with the consideration of influence attenuation. It is observed that *PageRank* fails to rank the influence significance in this case. All the top-10 *PageRank* papers have very low in-degrees (citations) compared to the top-10 *influence centrality* and *in-degree centrality* papers. They receive such high rankings simply because they are old papers that do not have any out-links (since the papers they cited are not included in the dataset).

4.3 Community detection

As discussed in Sect. 3.4, when initializing the centroids in our IGSK algorithm, we use a *closeness threshold* φ to avoid selecting centroids that are too close to each other. In practice, one may tune the threshold to find the best community partition in terms of (highest) modularity. However, to evaluate the effectiveness of our IGSK algorithm and to make a fair comparison with other algorithms, we cannot tune it for each network individually so as to present the best results. Instead, we empirically set φ to 0.6 when *depthLimit* is 2, and 0.8 when *depthLimit* is 3 for all the networks used in the experiment. The only exception is that for those LFR benchmarks with really fuzzy community structures (e.g. the topological mixing parameter $\mu_t = 0.8$), we do the random initialization of centroids.

The results of applying IGSK to the 5 real-life networks are listed in Table 3. We use *RandIndex* for comparison with the results of 6 algorithms given in Yang et al. (2011). These 6 algorithms are *RankClus* (Sun et al. 2009), *Walktrap* (Pons and Latapy 2006), *K-means* (Dhillon et al. 2005), *LinkCommunity* (Ahn et al. 2010), *SPICi* (Jiang and Singh 2010), and *Betweenness* (Girvan and Newman 2002). We also use *Normalized Mutual Information* (NMI) for comparison with the results of another set of 6 algorithms given in Hajibagheri et al. (2013), which include *GPSODM* (Hajibagheri et al. 2013), *GGADM* (Hajibagheri et al. 2012), *HA* (Leung et al. 2009), *MMC* (van Dongen 2000), *LPA* (Raghavan et al. 2007), and *Infomap* (Rosvall and Bergstrom 2008). It is shown that IGSK is only inferior to *GPSODM* and *LPA* a little bit on the American college football network, but overall, IGSK achieves the best performance.

We illustrate our results of the tests on the 4 sets of undirected and unweighted LFR benchmarks (1000-S/B and 5000-S/B) in Fig. 7a, in which each curve shows the variation of the averaged NMI score with respect to the topological mixing parameter μ_t. Our IGSK algorithm demonstrates excellent performance. Even when μ_t is set to 0.5 (the threshold of defining strong communities), IGSK achieves NMI scores of 0.999, 0.992, 0.968, and 0.99 for 1000-S, 1000-B, 5000-S, and 5000-B datasets, respectively. Further, IGSK is generally not sensitive to the community size or the network size.

We illustrate in Fig. 7b, c the performance of the 8 state-of-the-art algorithms examined in Lancichinetti and Fortunato (2009a). They are referred to as *Blondel et al.* (Blondel et al. 2008), *MCL* (van Dongen 2000), *Infomod*

Table 3 Comparison on real-life networks using RandIndex and NMI

Algorithm	RandIndex			Algorithm	NMI		
	Karate	Mexican	Sawmill		Football	Dolphin	Karate
RankClus	1.000	0.489	0.530	GPSODM	1.000	0.723	1.000
Walktrap	0.745	0.536	0.560	GGADM	0.910	0.736	1.000
K-means	0.941	0.536	0.527	HA	0.907	0.707	0.754
LinkCommunity	0.743	0.536	0.560	MMC	0.885	0.579	1.000
SPICi	0.586	0.553	0.629	LPA	0.927	0.710	0.751
Betweenness	0.913	0.605	0.570	Infomap	0.899	0.695	0.643
IGSK	1.000	0.716	0.870	IGSK	0.924	0.814	1.000

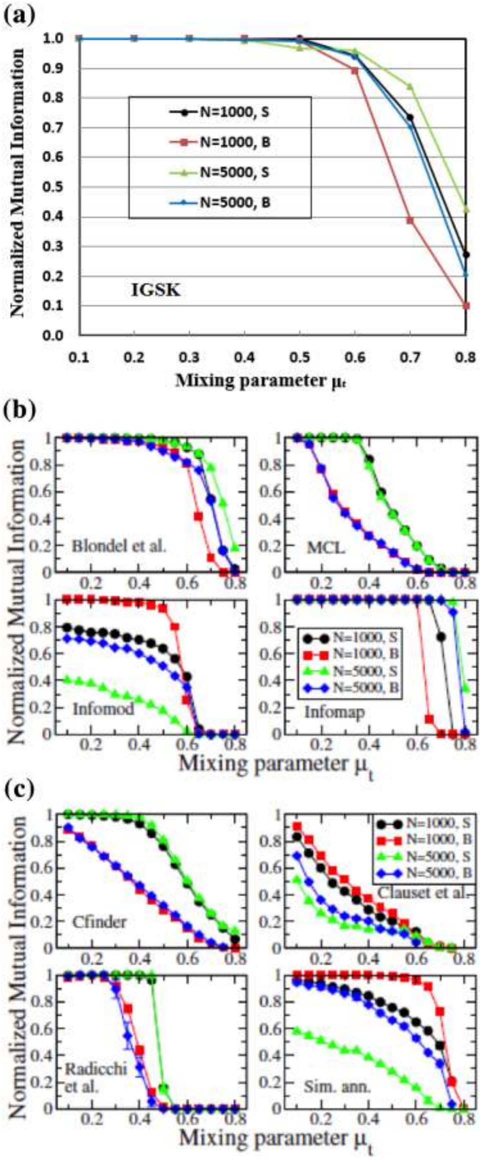

Fig. 7 Performance comparison on undirected and unweighted LFR benchmark graphs. **a** *IGSK*; **b** *Blondel et al.*, *MCL*, *Infomod*, and *Infomap*; **c** *Cfinder*, *Clauset et al.*, *Radicchi et al.*, and *Sim. ann.* [Lancichinetti and Fortunato (2009a), Copyright by The American Physics Society]

(Rosvall and Bergstrom 2007), *Infomap*, *Cfinder* (Palla et al. 2005), *Clauset et al.* (Clauset et al. 2004), *Radicchi et al.* (Radicchi et al. 2004), and *Sim. ann.* (Guimera and Amaral 2005). Apparently, IGSK outperforms 7 of them. Only *Infomap* looks a little better than IGSK. But as shown in Table 3, IGSK obviously outperforms *Infomap* on all the 3 real-life networks.

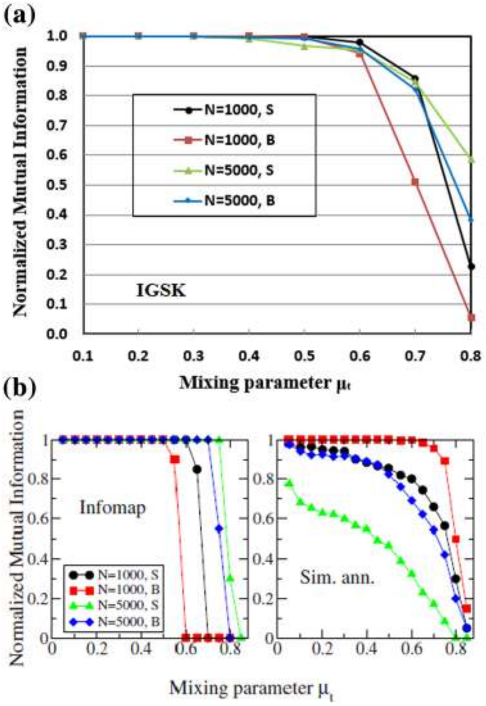

Fig. 8 Performance comparison on directed and unweighted LFR benchmark graphs. **a** *IGSK*; **b** *Infomap* and *Sim. ann.* [Lancichinetti and Fortunato (2009a), Copyright by The American Physics Society]

Finding communities in directed networks is even more challenging. Most existing algorithms are not able to deal with directed networks. We generate 4 sets of directed and unweighted networks using the same parameters as Lancichinetti and Fortunato (2009a), in which both the *degree-distribution exponent* and the *topological mixing parameter* refer to the in-degree of the nodes while the out-degree is kept constant for all nodes. This setting makes the resulting networks similar to the citation networks in terms of in-degree/out-degree distributions. Therefore, as we did with the arXiv HEP citation network, we reverse these LFR directed graphs to reflect the influence flow in our influence diffusion model and then run IGSK to find the communities. We illustrate our results in Fig. 8a and compare the performance of IGSK against the 2 algorithms investigated in Lancichinetti and Fortunato (2009a) as seen in Fig. 8b. Once again, IGSK shows remarkable performance in directed networks as well (even better than its performance in undirected networks). It is better than *Infomap* on 1000-B/S datasets, and clearly outperforms *Sim. ann* on 1000-S and 5000-S/B datasets.

To verify our weight-normalization scheme, we apply IGSK to the 4 sets of undirected and weighted networks. As done in Lancichinetti and Fortunato (2009a), we fix the topological mixing parameter μ_t to 0.5 and 0.8, respectively, and examine its performance when the weight mixing parameter μ_w varies from 0.1 to 0.8. Let us first take a look at the distribution of the weights as described in Lancichinetti and Fortunato (2009b). For a node of degree k_i, its expected *internal* weight and *external* weight can be expressed as

$$w_i^{(int)} = \frac{1-\mu_w}{1-\mu_t} k_i^{\beta-1}$$
$$w_i^{(ext)} = \frac{\mu_w}{\mu_t} k_i^{\beta-1}.$$

Then the ratio of *internal* weight to *external* weight (referred to as *int/ext ratio*) is related to the mixing parameters in a simple way:

$$int/ext\ ratio = \frac{w_i^{(int)}}{w_i^{(ext)}} = \frac{\mu_t(1-\mu_w)}{\mu_w(1-\mu_t)}.$$

To better understand how the weight plays its role in shaping the community structure with respect to the network topology, we run IGSK on each weighted network twice: one ignores the weights (denoted as *IGSK-ignore*), and the other considers the weights (denotes as *IGSK-consider*). The results are illustrated in the first 2 plots in Fig. 9. As we can see, for 5000-S-0.8 and 5000-B-0.8 datasets, their topological mixing parameter μ_t is 0.8, which indicates their community structure is fuzzy. While the weight mixing parameter μ_w varies from 0.1 to 0.7, the *int/ext ratio* decreases from 36 to 1.7 but is always > 1, which means the weight distribution always reinforces the community structure in these cases. *IGSK-ignore* gives low NMI scores as expected since it completely ignores the useful weight information. In contrast, *IGSK-consider* takes advantage of the weight information and greatly improve the performance. Further, it reflects a sensible pattern: the higher *int/ext ratio* of the weight, the stronger reinforcement of the community structure, the greater performance improvement *IGSK-consider* achieves.

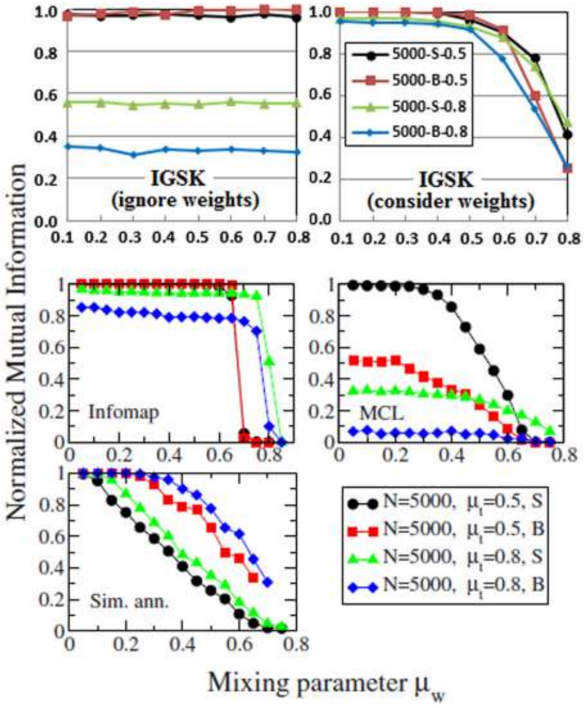

Fig. 9 Performance comparison on undirected and weighted LFR benchmarks. Plots for *Infomap*, *MCL*, and *Sim. ann* given in Lancichinetti and Fortunato (2009a), Copyright by The American Physics Society

For 5000-S-0.5 and 5000-B-0.5 datasets, μ_t is 0.5, which indicates the community structure is relatively clear topologically. In this case, when μ_w falls in the range from 0.1 to 0.4, the weight distribution confirms the community structure since the corresponding *int/ext ratio* is greater than 1. However, when $\mu_w > 0.5$, the *int/ext ratio* becomes smaller than 1, which implies the weight distribution turns into undermining the community structure. Our experimental results provide strong evidence of the above argument. On the one hand, *IGSK-ignore* shows excellent performance consistently from $\mu_w = 0.1$ to 0.8, which is expected as IGSK does on unweighted networks when $\mu_t = 0.5$ (Fig. 7a). On the other hand, *IGSK-consider* gives perfect NMI scores when $\mu_w < 0.5$, which outperforms *IGSK-ignore* by taking into account the weight information that reinforces the community structure. However, its performance worsens dramatically due to the misleading weight information when $\mu_w > 0.5$.

The tests on the 4 sets of weighted networks demonstrates that our IGSK algorithm effectively captures both the network connectivity and the weight information, even though it is not able to judge whether the weight information strengthens or undermines the community structure. Practically, it can be easily fixed by running both *IGSK-consider* and *IGSK-ignore* and taking the output of the one with higher modularity. In addition, the experiment brings forth an important point that the community structure is *primarily* determined by the network topology; the weight information is a *secondary* factor that may reinforce the community structure or make it fuzzy. Finally, as shown in Fig. 9, IGSK achieves better performance than *Infomap* on the 5000-B-0.8 dataset and clearly outperforms *MCL* and *Sim. ann.* on all the 4 datasets.

4.4 Overlapping community and role detection

IGSK can be easily adapted to the detection of overlapping community and individual roles in each community. The influence-based belonging factor defined in Sect. 3.5 is a good fit for quantifying the strength of association between all pairs of nodes and communities, and the three community-level influence rankings provide us a new sensible perspective and tool to deal with the role detection. We take Zarchary's karate-club network as an example and list in Table 4 its *comprehensive-influence* ranking, *internal-influence* ranking, *external-influence* ranking, belonging factors, and IGSK community assignment of each node. It is noted that the partition by IGSK matches the ground

Table 4 Influence rankings and belonging factors of Zachary's karate club. *Com-Rank* denotes comprehensive-influence ranking, *Int-Rank* denotes internal-influence ranking, *Ext-Rank* denotes external-influence ranking, *BF-1* denotes belonging factor to community 1, *BF-2* denotes belonging factor to community 2, and *CA* denotes community assignment of IGSK ($depthLimit = 2$)

Node	Com-Rank	Int-Rank	Ext-Rank	BF-1	BF-2	CA
1	1	1	3	0.795	0.205	1
3	2	4	1	0.542	0.458	1
2	3	2	5	0.797	0.203	1
14	4	5	2	0.630	0.370	1
4	5	3	6	0.859	0.141	1
8	6	6	7	0.849	0.151	1
20	7	11	4	0.569	0.431	1
6	8	7	10	0.946	0.054	1
7	8	7	10	0.946	0.054	1
5	10	9	10	0.938	0.063	1
11	10	9	10	0.938	0.063	1
18	12	12	8	0.903	0.097	1
22	12	12	8	0.903	0.097	1
13	14	14	10	0.929	0.071	1
12	15	15	10	0.895	0.105	1
17	16	16	16	1.000	0.000	1
34	1	1	2	0.190	0.810	2
33	2	2	5	0.155	0.845	2
9	3	6	1	0.419	0.581	2
32	4	4	2	0.306	0.694	2
24	5	3	9	0.073	0.927	2
31	5	7	4	0.309	0.691	2
30	7	5	10	0.063	0.938	2
28	8	8	7	0.234	0.766	2
29	9	14	6	0.286	0.714	2
15	10	9	10	0.086	0.914	2
16	10	9	10	0.086	0.914	2
19	10	9	10	0.086	0.914	2
21	10	9	10	0.086	0.914	2
23	10	9	10	0.086	0.914	2
10	15	16	7	0.333	0.667	2
27	16	15	16	0.074	0.926	2
26	17	16	18	0.043	0.957	2
25	18	18	16	0.091	0.909	2

truth perfectly. We group the nodes by their community assignment and sort the nodes in each community by their comprehensive-influence ranking.

BF-1 and *BF-2* list the belonging factors of each node associated with communities 1 and 2, respectively, which converts the original *crisp* assignment into a *fuzzy* one for overlapping community analysis. It is interesting to observe that each node has a higher belonging factor to its own community than to the other one, which follows our intuition. Moreover, we refer to a node with multiple membership as an overlapping node. The belonging factors enable us to identify overlapping nodes with respect to a *belonging threshold*. If a node's belonging factor to a community is greater than the belonging threshold, the node is considered as a member of that community. In this case, if we set the threshold to 0.3, we find the set of overlapping nodes are nodes 3, 14, 20, 9, 32, 31, and 10. If it is set to 0.4, then only nodes 3, 20, and 9 are regarded as overlapping nodes. The *belonging threshold* is a user-defined application-dependent parameter in practice.

Further, it is straightforward to use the 3 influence rankings to uncover the roles of individual members in each community. As shown in Table 4, nodes 1 (the instructor) and 34 (the president) are of the top ranking on both comprehensive influence and internal influence in communities 1 and 2, respectively. It is reasonable to identify them as *leaders* of their respective communities, which matches the ground truth. Moreover, if we refer to a node as a *core member* if its ranking on both comprehensive influence and internal influence is among top 5 (this number can be adjusted according to the size of the community in practice), nodes 3, 2, 14, and 4 can be regarded as the core members in community 1, and nodes 33, 32, and 24 are core members in community 2. Similarly, if we refer to a node as an inter-community *liaison* if its external-influence ranking is among top 3 (not including the leaders), we find that node 3, 14, and 20 are liaisons of community 1, and nodes 9, 32, and 31 are liaisons of community 2. As we can see, our approach digs out rich connectivity information that allows us to probe the structural importance of a node in the network in a much meaningful and detailed manner.

4.5 Space and time complexity analysis

Let n denote the total number of nodes in the network, b denote the average node out-degree, d denote the depth limit, K denote the number of communities, I denote the number of iterations to converge, and L denote the average length of the influence vectors.

The space complexity depends on the influence vectors. Using an array of length n for each influence vector definitely wastes a lot of space since no nodes can spread influence to all other nodes in the whole network in general. To improve the space complexity, we store the influence vector of each node in a compact array that keeps only the nodes it influences. Then the space complexity is $O(nL)$. L is directly affected by the depth limit d, as well as the average node degree b and the network size n. It is also related to the community structure. For example, for the LFR 5000-S networks, when the topological mixing parameter is 0.1 (clear community structure), L is about 100

($d = 2$) and 550 ($d = 3$), respectively. But when the mixing parameter is 0.8 (fuzzy community structure), L jumps to 370 and 3,600, respectively.

It is hard to rigorously estimate the time complexity since it is closely related to the community structure. Generating the influence matrix is really fast with a time complexity of $O(nL)$ roughly. Once the influence-centrality value of each node is obtained, influence ranking is $O(n \log n)$ using *Heapsort*. IGSK algorithm has a time complexity of *O(nKLI)*. It is demonstrated in our experiments that IGSK converges fast when the community structure is clear. For example, for almost all our experimented LFR benchmarks, it converges in 2 iterations when the topological mixing parameter is 0.3 or less. When the community structure is fuzzy, however, it may take 10 iterations or more. We force it to stop if it does not converge after 8 iterations.

5 Conclusions

In this paper, we provide a new perspective on the influence-based connectivity of network graph topology and propose a novel influence diffusion model that is applicable to both undirected/directed and unweighted/wegithed networks.

Using this model, we define a new *influence centrality* and *Shared-Influence-Neighbor* (SIN) similarity. The *influence centrality* differentiates the node's comprehensive influence significance in a more detailed and precise manner, and the SIN similarity is well-suited as a refined vertex-pair proximity metric. We present an *influence-guided spherical K-means* (IGSK) algorithm for community detection and extensively test it on both real-life and synthetic networks. Experimental results demonstrate its superior performance in both undirected/directed and unweighted/weighted networks. Further, it enables us to uncover the overlapping community structure and identify the overlapping nodes and the roles of individual members in each community. All of these essential tasks are naturally integrated in one framework.

In our influence diffusion model, it is implicitly assumed that the nodes are homogeneous. In the future work, it would be interesting to extend the model to the network with nodes of heterogeneous roles. The main drawback of our IGSK is that it requires the pre-specified number of communities. In addition, although IGSK is fairly efficient, it does not scale well enough on large-scale networks. It is desirable to investigate the combination of this influence-based approach with other clustering techniques to avoid pre-specifying the number of communities and further improve the efficiency. It would be a promising direction to combine this approach with content analysis, namely considering both the network graph topology and the nodes' profile information. Finally, we point out that the *influence centrality* and the *SIN similarity* introduced in this paper provide important implications for viral marketing and link prediction in social networks. A lot of work can follow.

Acknowledgments We thank the anonymous reviewers for their insightful remarks and suggestions that allow us to improve significantly the quality of this paper.

References

Ahn Y, Bagrow J, Lehmann S (2010) Link communities reveal multiscale complexity in networks. arXiv:0903.3178v3 [physics.soc-ph]

Blondel V, Guillaume J-L, Lambiotte R, Lefebvre E (2008) The louvain method for community detection in large networks. J Stat Mech Theory Exp 10:P10008

Bonacich P (1972) Factoring and weighting approaches to status scores and clique identification. J Math Sociol 2:113–120

Brandes U, Fleischer D (2005) Centrality measures based on current flow. In: 22nd annual conference on theoretical aspects of computer science, pp 533–544

Christakis NA, Fowler JH (2007) The spread of obesity in a large social network over 32 years. N Engl J Med 357:370–379

Clauset A, Newman M, Moore C (2004) Finding community structure in very large networks. Phys Rev E 70:066111

Dhillon I, Modha D (2001) Concept decompositions for large sparse text data using clustering. Mach Learn 42(1):143–175

Dhillon I, Guan Y, Kulis B (2005) A fast kernel-based multilevel algorithm for graph clustering. In: 11th ACM conference on knowledge discovery and data mining, pp 629–634

Donetti L, Muñoz M (2004) Detecting network communities: a new systematic and efficient algorithm. J Stat Mech 2004:P10012

Easley D, Kleinberg J (2010) Networks, crowds, and markets: reasoning about a highly connected world. Cambridge University Press, Cambridge

Estrada E, Hatano N (2009) Communicability graph and community structures in complex networks. J Appl Math Comput 214:500–511

Fortunato S (2010) Community detection in graphs. Phys Rep 486(3–5):75–174

Fortunato S, Barthelemy M (2007) Resolution limit in community detection. Proc Natl Acad Sci USA 104(1):36–41

Freeman LC (1977) A set of measures of centrality based on betweenness. Sociometry 40:35–41

Freeman LC, Borgatti SP, White DR (1991) Centrality in valued graphs: a measure of betweenness based on network flow. Soc Netw 13:141–154

Gehrke J, Ginsparg P, Kleinberg JM (2003) Overview of the 2003 kdd cup. SIGKDD Explor 5:149–151

Gil-Mendieta J, Schmidt S (1996) The political network in mexico. Soc Netw 18(4):355–381

Girvan M, Newman M (2002) Community structure in social and biological networks. Proc Natl Acad Sci USA 99(12):7821–7826

Guimera R, Amaral L (2005) Functional cartography of complex metabolic networks. Nature 433:895–900

Guimera R, Sales-Pardo M, Amaral L (2004) Modularity from fluctuations in random graphs and complex networks. Phys Rev E 70:025101

Hajibagheri A, Alvari H, Hamzeh A, Hashemi S (2012) Community detection in social networks using information diffusion. In: 2012 IEEE/ACM international conference on advances in social networks analysis and data mining, pp 702–703

Hajibagheri A, Hamzeh A, Sukthankar G (2013) Modeling information diffusion and community membership using stochastic optimization. In: 2013 IEEE/ACM international conference on advances in social networks analysis and data mining, pp 175–182

Jarvis RA, Patrick EA (1973) Clustering using a similarity measure based on shared nearest neighbors. IEEE Trans Comput C–22(11):1025–1034

Jiang P, Singh M (2010) Spici: a fast clustering algorithm for large biological networks. Bioinformatics 26(8):1105–1111

Katz L (1953) A new status index derived from sociometric index. Psychometrika 18:39–43

Kempe D, Kleinberg J, Tardos E (2003) Maximizing the spread of influence through a social network. In: 9th ACM conference on knowledge discovery and data mining, pp 137–146

Lancichinetti A, Fortunato S (2009a) Community detection algorithms: a comparative analysis. Phys Rev E 80:056117(1–11)

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

Lancichinetti A, Fortunato S, Radicchi F (2008) Benchmark graphs for testing community detection algorithm. Phys Rev E 78:046110

Leskovec J, Lang KJ, Mahoney MW (2010) Empirical comparison of algorithms for network community detection. In: 19th international conference on world wide web, pp 631–640

Leung I, Hui P, Liò P, Crowcroft J (2009) Towards real-time community detection in large networks. Phys Rev E 79:066107

Lusseau D, Schneider K, Boisseau OJ, Haase P, Slooten E, Dawson SM (2003) The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. Behav Ecol Sociobiol 54:396–405

Malliaros F, Vazirgiannis M (2013) Clustering and community detection in directed networks: a survey. Phys Rep 533:95–142

Michael J, Massey J (1997) Modeling the communication network in sawmill. For Prod J 47:25–30

Nadler B, Lafon S, Coifman R, Kevrekidis I (2005) Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. In: 19th annual conference on neural information processing systems

Newman M (2004) Analysis of weighted networks. Phys Rev E 70:056131

Newman M (2005) A measure of betweenness centrality based on random walks. Soc Netw 27:39–54

Newman M (2006) Modularity and community structure in networks. Proc Natl Acad Sci 103(23):8577–8582

Newman M, Girvan M (2004) Finding and evaluating community structure in networks. Phys Rev E 69:026113

Noh JD, Rieger H (2004) Random walks on complex networks. Phys Rev Lett 92:11870

Page L, Brin S, Motwani R, Winograd T (1999) The pagerank citation ranking: bringing order to the web. In: Technical report, Stanford InfoLab. Stanford University, California

Palla G, Derenyi I, Farkas I, Vicsek T (2005) Uncovering the overlapping community structure of complex networks in nature and society. Nature 435:814–818

Pons P, Latapy M (2006) Computing communities in large networks using random walks. J Graph Algorithm Appl 10(2):191–218

Radicchi R, Castellano C, Cecconi F, Parisi D (2004) Defining and identifying communities in networks. Proc Nal Acad Sci USA 101:2658–2663

Raghavan UN, Albert R, Kumara S (2007) Near linear time algorithm to detect community structures in large-scale networks. Phys Rev E 76:03106

Rosvall M, Bergstrom C (2007) An information-theoretic framework for resolving community structure in complex networks. Proc Natl Acad Sci 104:7327–7331

Rosvall M, Bergstrom C (2008) Maps of random walks on complex networks reveal community structure. Proc Natl Acad Sci USA 105:1118–1123

Sabidussi G (1966) The centrality index of a graph. Psychometrika 31:581–603

Stephenson KA, Zelen M (1989) Rethinking centrality: methods and examples. Soc Netw 11:1–37

Sun Y, Han J, Zhao P, Yin Z, Cheng H, Wu T (2009) RankClus: integrating clustering with ranking for heterogeneous information network analysis. In: EDBT, Saint Petersburg, Russia

van Dongen S (2000) Graph clustering by flow simulation. PhD thesis, University of Utrecht

Wang W, Street WN (2014) A novel algorithm for community detection and influence ranking in social networks. In: 2014 IEEE/ACM international conference on advances in social networks analysis and data mining, pp 555–560

Wasserman S, Faust K (1994) Social network analysis: methods and applications. Cambridge University Press, Cambridge

Xie J, Kelley S, Szymanski BK (2013) Overlapping community detection in networks: the state-of-the-art and comparative study. ACM Comput Surv 45(4):1–35

Yang Y, Sun Y, Pandit S, Chawla N, Han J (2011) Is objective function the silver bullet? A case study of community detection algorithms on social networks. In: 2011 IEEE/ACM international conference on advances in social networks analysis and data mining, pp 394–397

Yen L, Fouss F, Decaestecker C, Francq P, Saerens M (2009) Graph nodes clustering with the sigmoid commute-time kernel: a comparative study. J Data Knowl Eng 68:338–361

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