Knowledge-Based Systems 000 (2017) 1–7

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A novel community detection algorithm based on simplification of complex networks

Liang Bai[a,b,*], Jiye Liang[a,*], Hangyuan Du[a], Yike Guo[b]

[a] Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan, Shanxi 030006 China
[b] Department of Computing, Imperial College London, SW7, London, United Kingdom

ARTICLE INFO

Article history:
Received 4 August 2017
Revised 3 December 2017
Accepted 7 December 2017
Available online xxx

Keywords:
Graph clustering
Community detection
Network representation
Min-cutting problem

ABSTRACT

Efficiently discovering the hidden community structure in a network is an important research concept for graph clustering. Although many detection algorithms have been proposed, few of them provide a visual understanding of the community structure in a network. In this paper, we define two measurements about the leading and following degrees of a node. Based on the measurements, we provide a new representation method for a network, which transforms it into a simplified network, i.e., weighted tree (or forest). Compared to the original network, the simplified network can easily observe the community structure. Furthermore, we present a detection algorithm which finds out the communities by min-cutting the simplified network. Finally, we test the performance of the proposed algorithm on several network data sets. The experimental results illustrate that the proposed algorithm can visually and effectively uncover the community structure.

1. Introduction

Since the data are modeled as networks in many complex systems [30], e.g., social networks and biological networks, recently increasing attention has been paid to complex networks analysis. Community structure [9,22] is a very important property of networks. Intuitively, a community (cluster) in a network consists of a cohesive group of nodes that are relatively densely connected to each other but sparsely connected to other dense groups. Community detection aims to identify the communities by only using the information encoded in the network topology. It can be seen as a procedure of *graph clustering*.

Community detection becomes one of the most important tasks to explore and understand how the networks work [10]. To solve the community detection problem, various types of algorithms have been proposed and developed, including latent space model, non-negative matrix factorization, block model approximation, spectral clustering, label propagation, and modularity maximization. These algorithms have different definitions of communities or clustering criteria, according to applications for different scientific needs [32]. Many of them have been successfully applied to different areas. The detailed review of these algorithm can be found in Section 2.

However, few of the existing community detection algorithms consider its visual understanding while detecting the community structure in a network. A good visual understanding can help us to easily recognize inherent communities and their intrinsic characteristics. For example, a community generally includes two important zones, i.e., the core and border, which can determine its shape and organization. However, due to the presence of lots of edges in the network, it is difficult for us to directly observe these zones. To overcome the deficiency, we will define leading and following degrees of a node to evaluate its representability and its following relations with other nodes. A node with high leading degree tends to be seen as a representative of some community. Several highly-connected representatives can constitute the core of a community. The border of a community tends to be made up of several nodes with low leading and high following degrees. Based on the leading and following degrees, we will provide a new representation method and community detection algorithm for network data sets. Compared to the original network, we will easily observe the community structure in the re-expressed network. The major contributions of this paper are as follows.

- Transform a complex network into a simplified network, i.e., weighted tree (or forest), which can reflect the core of each

* Corresponding author at: Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan, Shanxi 030006 China.
E-mail addresses: bailiang@sxu.edu.cn (L. Bai), ljy@sxu.edu.cn (J. Liang), duhangyuan@sxu.edu.cn (H. Du), y.guo@imperial.ac.uk (Y. Guo).

https://doi.org/10.1016/j.knosys.2017.12.007
0950-7051/

community and the membership of each node to the communities. This step simplifies the network and makes users easily observe the community structure.
- The community detection problem is seen as a min-cut problem of the obtained tree. Compared to cutting the original network, cutting the tree can be easily solved.

The following is the outline of this paper. Section 2 presents the new community detection algorithm. Section 3 review the related works. Section 4 demonstrates the performance of the proposed algorithm. Finally, we draw conclusions and suggest future work in Section 5.

2. Related works

Currently, many approaches have been proposed to detect the non-overlapping community structures [8]. We introduce the five well-known types of algorithms as follows:

(1) *The feature mapping model* mainly maps nodes of a network into a low-dimensional Euclidean space. The proximity between the network connectivity nodes is kept in the new space; then, the nodes are clustered in the low-dimensional space by using traditional clustering algorithms such as *k*-means [17] and linkage [34]. The representative includes Latent space model [27] non-negative matrix factorization [16,33,35] and spectral clustering algorithms [11,28].
(2) *The block approximation model* sees a community detection problem as a matrix blocking problem, which reorder the index of each node according to their community membership and approximate a given network by a block structure [5]. Each block represents a community.
(3) *The label propagation model* mainly uses the neighbor information of each node to determine its label and do not need any prior knowledge of community structure. The representative algorithm of LPA was proposed by Raghavan et al. [24]. It has greatly received attention for its nearly linear time complexity in finding communities. However, since the label of each node depends on those of other nodes, the algorithm can only linearly propagate the labels. In addition, the convergence speed and clustering effectiveness of the algorithm are very sensitive to the update order of label information. Therefore, several improved LPA algorithms are developed in [1,12,31].
(4) *The modularity maximization model* [6,7,10,19,21] transforms a community detection problem into a modularity maximization problem. Modularity is a commonly used criterion for community detection, which measures the strength of a community partition for real-world networks by taking into account the degree distribution of nodes. The type of the algorithms mainly apply different hierarchical clustering strategies to partition networks, which is very time-consuming. The fast unfolding algorithm proposed by Blondel et al.[2] is a fast heuristic method for the modularity optimization. The algorithm uses the idea of the label propagation models to reduce the computing cost. Compared to other algorithms for modularity maximization, the fast unfolding algorithm has good scalability for large networks.
(5) *The information-theoretical model* is developed by Rosvall et al. for community structure [26]. They transfer the problem of community detection into an information coding problem. Furthermore, an information map algorithm of random walks [25] is proposed to solve the optimization problem.

Except for the above types, some new techniques are applied to community detection. For example, the parallel and distributed algorithms [13,29] are proposed to fast deal with large-scale networks. The unified methods [3,4,18] are developed to detect non-overlapping and overlapping communities. These new methods can more effectively tackle complex networks.

3. The community detection algorithm

Given an original network, we can observe link relation between nodes but do not easily see its community structure. Therefore, we provide a new representation method of the network to better reflect its community structure. In the proposed method, we employ two measurements , i.e., leading and following degrees, to measure the representability of a node and its following relations with other nodes. We assume that a community is made up of leading nodes and following nodes. The leading nodes in the community are seen as its representatives and have high leading degrees. The higher the leading degree of a node is, the more representability it has in the community it belongs to. For the following nodes in the community, they have lower representability but higher following degrees to the leading nodes. If the following degree of a node to other node is high, they possibly belong to the same community. Therefore, we can easily observe the community structure in the re-expressed network.

First, we provide some related notations and definitions on the proposed method. Suppose that $G = \langle V, E, A\rangle$ is an undirected network with V which is a set of n vertices and E which is a set of m edges. $A = \{A_{ij}\}_{1\le i,j\le n}$ is an adjacency matrix, where A_{ij} is the weight of edge $< v_i, v_j >$. For an unweighted graph, if there is an edge between nodes v_i and v_j, $A_{ij} = 1$, otherwise, $A_{ij} = 0$. We assume $A_{ii} = 1$ for $1 \le i \le n$. $N_i = \{v_j | < v_i, v_j >\in E\}$ is a vertex set including all the neighbors of $v_i \in V$. $d(v_i) = \sum_{v_j\in N_i} A_{ij}$ is degree of node v_i . For any two nodes, we use the number of common neighbors between two nodes to simply reflect their similarity. The similarity measure is formalized as

$$\delta(v_i, v_j) = \sum_{v_z\in N_i\cup N_j} A_{iz}A_{jz}. \tag{1}$$

The more the number of their common neighbors is, the more similar they are.

Next, we introduce how to evaluate the leading degree of a node. In this, we mainly consider the leadership of a node to its neighbors. We assume that a node only can lead such its neighbors that have lower influences than it and high similarity with it. If some of its neighbors have higher influences than it, it cannot lead them. In addition, its leadership to its neighbors with lower influences depends on its similarity with them. If the similarity is high, its leadership to them should be large. Here, we use the degree of a node to reflect its influence. The more its degree is, the more the number of its neighbors is, the more it has high influence. Therefore, the leading degree of a node is defined as

$$L(v_i) = \sum_{d(v_j)<d(v_i), v_j\in N_i} \delta(v_i, v_j). \tag{2}$$

Let us use an example the network from Zachary's karate club which is shown in Fig. 1 to explain the leading degree. We consider node 32 in this network. According to Fig. 2, we see that node 32 links other six nodes. We think that it only may lead nodes 25, 26 and 29 whose degrees are less than it. Thus, we use the sum of the similarity between it and nodes 25, 26 and 29 to reflect its leading degree. Furthermore, we introduce how to evaluate the following degrees of a node with other nodes. We assume that a node only follows such nodes that have higher leadership than it. Its following degree to some node depends on the proportion of their common neighbors within its neighbors. For a node, the more its neighbors have edges with its following node, the higher

Please cite this article as: L. Bai et al., A novel community detection algorithm based on simplification of complex networks, Knowledge-Based Systems (2017), https://doi.org/10.1016/j.knosys.2017.12.007

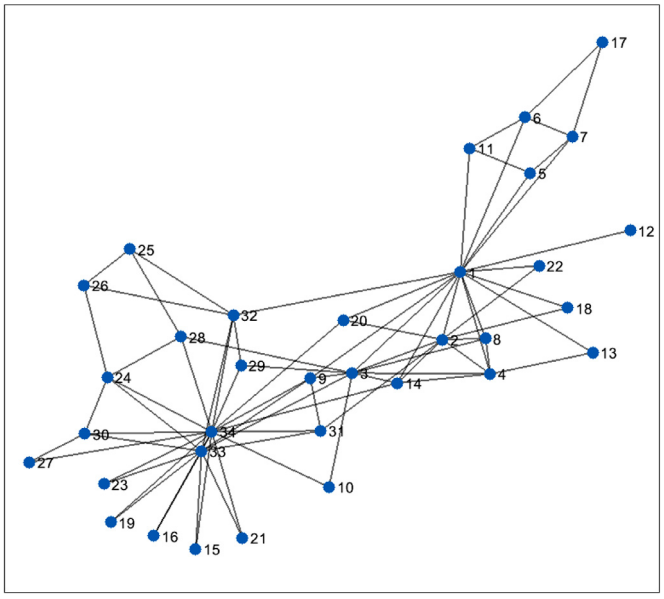


Fig. 1. Original network of Zachary's karate club.

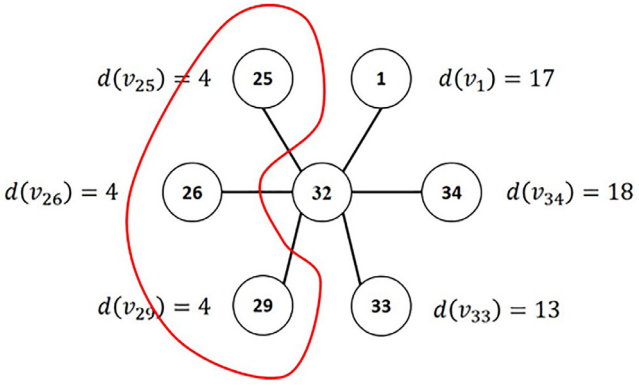


Fig. 2. Explanation for the leading degree L.

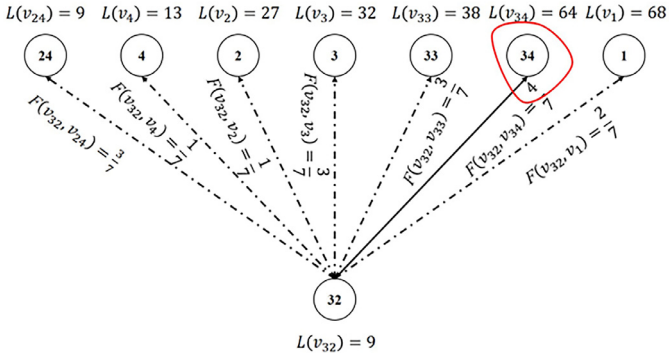


Fig. 3. Explanation for the following degree F.

its following degree is. Therefore, the following degree of a node is defined as

$$F(v_i, v_j) = \begin{cases} \dfrac{\delta(v_i, v_j)}{d(v_i)}, & if \quad L(v_j) \geq L(v_i) \\ 0, & otherwise. \end{cases} \tag{3}$$

Let us continue taking the karate network for an example. In Fig. 3, we can see that node 32 only may follow nodes 1, 2, 3, 4, 24, 32, 33 and 34 whose leading degrees are no less than it. According to Eq. (3), we compute its following degrees to these nodes.

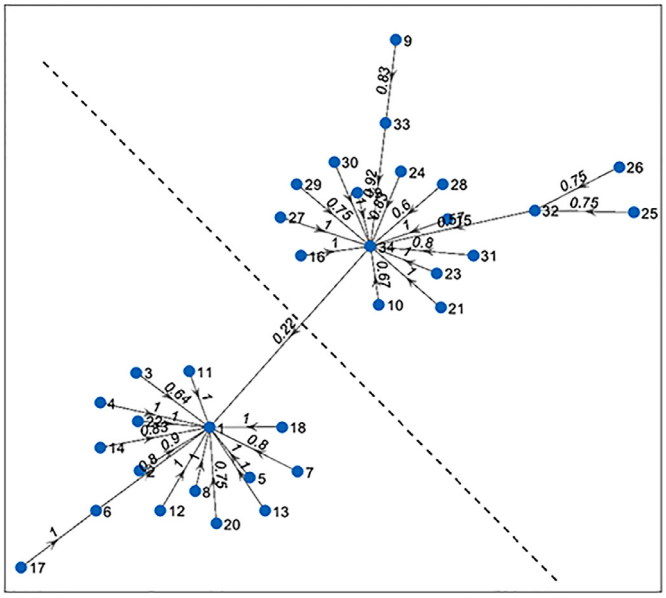


Fig. 4. Simplified network of Zachary's karate club.

Based on $L(.)$ and $F(.,.)$, we propose a new representation of a network which maps it to a weighted and directed network. The re-expressed network $G' =< V', E', A' >$ is defined as follows.

$$\begin{cases} V' = V, \\ E' = \{< v_i, v_q > | F(v_i, v_q) = \max\limits_{v_j \in V} F(v_i, v_j) \ and \ F(v_i, v_q) > 0, v_i \in V\}, \\ A' = \{A'_{ij}\}_{1 \leq i,j \leq n}, \ where \ A'_{ij} = \begin{cases} F(v_i, v_j), & if < v_i, v_j > \in E', \\ 0, & otherwise. \end{cases} \end{cases} \tag{4}$$

According to the above definition, we can see that the re-expressed network is a tree structure. G' has the following properties.

Property 1. *The number of edges in G' is no more than $n-1$, where n is the number of nodes.*

Property 2. *Let T be a subtree of G', since $L(v_j) \geq L(v_i)$ for any $< v_i, v_j >$ in T, its root has the maximum L value in T.*

Property 3. *For any two nodes v_i and v_j, if there is a path between them in G', a path between them exists in the original network G.*

Property 4. *Let T be a subtree of G', V_T be the set of all the nodes and E_T be the set of all the edges in T. If we partition the nodes of T into two clusters, T has the following property*

$$\min_{C \in V_T} \sum_{v_i \in C, v_j \in V_T - C} A'_{ij} = \min_{< v_p, v_q > \in E_T} A'_{pq}. \tag{5}$$

In the following, we discuss whether G' can easily reflect the community structure, according to these properties. Since we only keep the maximum following degree of each node, seen in Fig. 3, G' has no more than $n-1$ edges. G' realizes the simplification of G, which is convenient for observing the community structure. Fig. 4 shows the simplified network of karate. Compared to the original network, we can clearly see that there are two obvious communities. According to Property 2, we can see that the root has the maximum leadership in a subtree. Therefore, if each subtree is seen as a community in G', its root and leaves can be viewed as the center and borders of the community, respectively. On the community, the weight of an edge reflects the membership of a node to the community. Fig. 4 shows nodes 1 and 34 are the centers of the two communities, respectively. For any two nodes in

Algorithm 1: The STCD algorithm.

Input: G, λ
Output: Ω
for $1 \le i \le n$ **do**
  Compute $L(v_i)$;
for $1 \le i, j \le n$ **do**
  Compute $F(v_i, v_j)$;
Obtain $G' =< V', E', A' >$ by Eq. (4)
for $< v_i, v_j > \in E'$ **do**
  if $A'_{ij} < \lambda$ **then**
    Delete edge $< v_i, v_j >$ from E' and set $A'_{ij} = 0$;
Find out all the subgraphs who are non-connected each other;
All the nodes in the lth subgraph belong to V_l;

the original network, there may be several paths. Property 3 illustrates that the simplified network G' retains a path between any two nodes, which mainly reflects their following relations. Therefore, compared to the original network, we can easily observe the community structure in the re-expressed network.

Based on the simplified network G', we describe the community detection problem as follows.

$$\min \left[\Delta(\Omega) = \sum_{v_i \in V_l, v_j \in V_h, 1 \le l \ne h \le k} A'_{ij} \right], \tag{6}$$

where $\Omega = \{V_1, V_2, \ldots, V_k\}$ is a partition of V, where V_l is the lth community and k is the number of communities. Compared to min-cutting the original network, it is not difficult to solve the problem. According to Property 4, we can conclude

$$\min \Delta(\Omega) = \min_{Q \subset E', |Q| = k-1} \sum_{< v_i, v_j > \in Q} A_{ij}. \tag{7}$$

Therefore, we can minimize $\Delta(\Omega)$ by deleting the edges with the first $k-1$ lowest weights from G' to find out k communities. However, since the number of communities k is often unknown for a network, we replace k with a parameter λ which should be in the interval [0, 1]. We delete all the edges whose weights are no more than λ from G' to automatically determine k. The k value depends on the λ value. The higher the λ value is, The more the k value may be.

The Simplified Tree-based Community Detection algorithm is described in Algorithm 1, called STCD. The basic operation of the proposed algorithm is $\delta(.,.)$. We know that the computing cost of the similarity between two nodes is linearly relevant to their degrees. Thus, in order to compute L and F, the algorithm needs to get the similarity between all the nodes whose computing cost is $O(2nm)$. Besides, cutting the simplified network needs $O(n)$ operations. Therefore, the time complexity of the algorithm is $O(2nm + n)$.

4. Experiment analysis

In this section, we test the performance of the STCD algorithm on several synthetic and real networks. The synthetic networks are produced by LFR benchmark [15] which provides a rich set of parameters to control the network topology. The real networks are downloaded from Newman and co-workers [14,20]. The detail description of these networks is shown in Table 1. The hardware environment of the experiment is a PC with an Intel 2.5Hz i7-4710MQ CPU and 16G RAM. The software platform is Matlab R2016b in Windows 10×64.

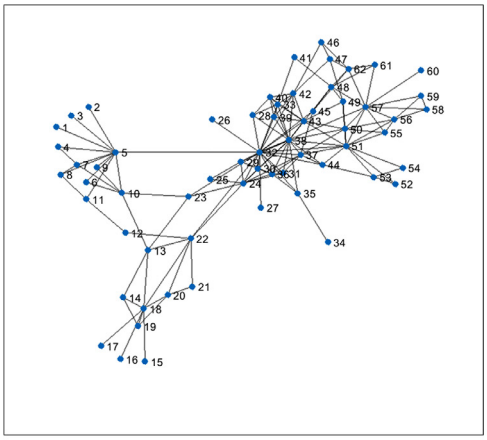
In order to show the effectiveness of the STCD algorithm, we first test it on Dolphins, Jazz and Email networks. Figs. 5–7 show their topological structures of the original, simplified and cut networks. According to Fig. 5(a), we see that there are three obvious centers of communities, i.e., nodes 32, 5 and 18 in Dolphins network. By using the STCD algorithm, we transform the original network into a tree. We can easily observe the centers and shapes of communities from Fig. 5(b). While setting $\lambda = 1/3$, we can detect the three communities, shown in Fig. 5(c). Next, we analyze Jazz network according to Fig. 6. From Fig. 6(a), we see that there are two obvious communities. However, we do not see the centers of communities from the original network. Fig. 6(b) shows the simplified network where we easily observe two centers, i.e., node 7 and 67. If we set $\lambda = 0.56$, we can obtain the two communities, seen in Fig. 6(c). Finally, we show the visual understanding of Email network. According to Fig. 7(a), we do not observe the community structure, due to the fact that there are many edges in this network. We use the STCD algorithm to map the original network into a tree, seen in Fig. 7(b). Here, we set $\lambda = 1/3$ to cut the tree into several subtrees. Fig. 7(c) illustrates there are many communities with different shapes. According to the above analysis, we see that the STCD algorithm can provide a good visual understanding of complex networks, compared to the original networks.

Furthermore, we compare the STCD algorithm with six algorithms including the fast modularity maximization (FMM) [19], the normalized spectral clustering (NSC) [28], the label propagation algorithm (LPA) [24], the fast unfolding communities (FUC) [2], the integrated community algorithm (IEDC) [18], and the MaxPerm algorithm (MP) [3]. While running the NSC algorithm, we set k to the true number of classes. Other algorithms can automatically determine the k value. For the STCD algorithm, we set $\lambda = 1/3$ in the comparison. These algorithms are carried out on six synthet-

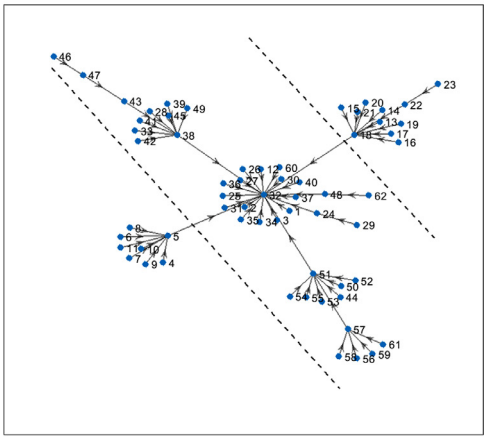
Table 1
Description of networks.

Source	Data sets	Description
Benchmark	S1	$n = 1000, \bar{d} = 15, d_i \le 50, 10 \le n_l \le 50, \mu = 0.3, \gamma = 2, \beta = 1$
	S2	$n = 1000, \bar{d} = 15, d_i \le 50, 10 \le n_l \le 50, \mu = 0.4, \gamma = 2, \beta = 1$
	S3	$n = 1000, \bar{d} = 15, d_i \le 50, 10 \le n_l \le 50, \mu = 0.5, \gamma = 2, \beta = 1$
	S4	$n = 1000, \bar{d} = 15, d_i \le 50, 10 \le n_l \le 50, \mu = 0.3, \gamma = 3, \beta = 2$
	S5	$n = 1000, \bar{d} = 15, d_i \le 50, 10 \le n_l \le 50, \mu = 0.4, \gamma = 3, \beta = 2$
	S6	$n = 1000, \bar{d} = 15, d_i \le 50, 10 \le n_l \le 50, \mu = 0.5, \gamma = 3, \beta = 2$
Real networks	Dolphins	$n = 62, m = 159, k = NA$
	Jazz	$n = 198, m = 2742, k = NA$
	Email	$n = 1133, m = 5451, k = NA$
	Grassweb	$n = 75, m = 113, k = 5$
	Football	$n = 115, m = 613, k = 12$
	Polbooks	$n = 105, m = 441, k = 3$
	Polblogs	$n = 1490, m = 16718, k = 2$

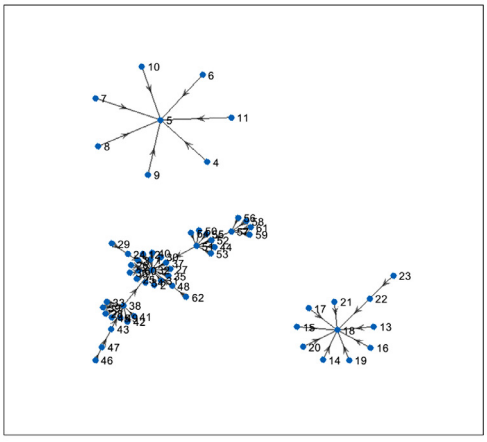
Please cite this article as: L. Bai et al., A novel community detection algorithm based on simplification of complex networks, Knowledge-Based Systems (2017), https://doi.org/10.1016/j.knosys.2017.12.007



(a)

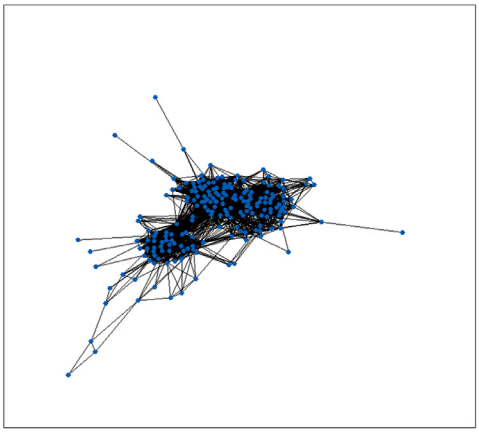


(b)

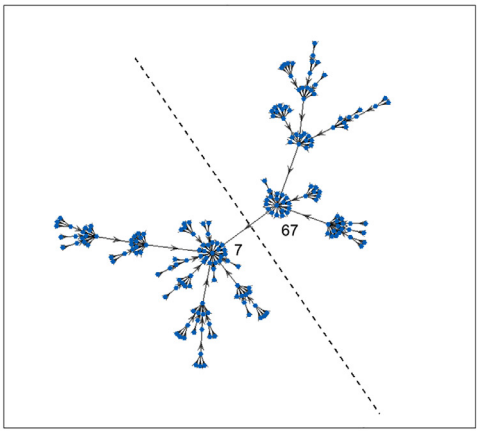


(c)

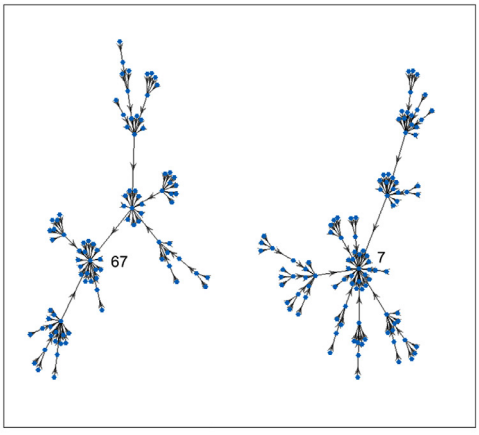
Fig. 5. Dolphins: (a) original network; (b) simplified network; (c) cut result.



(a)



(b)



(c)

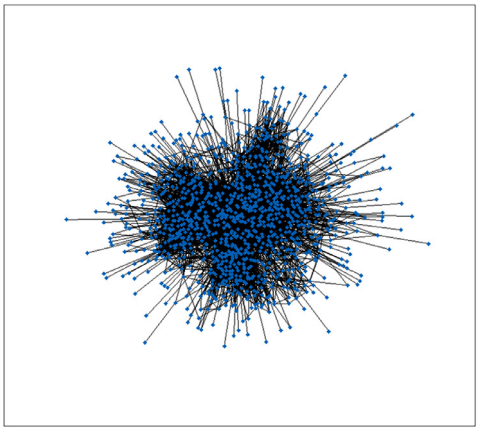
Fig. 6. Jazz musicians: (a) original network; (b) simplified network; (c) cut result.

ical and four real networks. To compare the effectiveness of different algorithms, we employ an external measure, i.e., the normalized mutual information (NMI) [23] and an internal measure, i.e., the modularity (Q) [19]. The normalized mutual information (NMI) is used to evaluate the similarity between a detection result and the "true" partition on each of the given networks. Given a set V of n nodes and two partitions, namely $C = \{c_1, c_2, \cdots, c_k\}$ (the detection result) and $P = \{p_1, p_2, \cdots, p_{k'}\}$ (the "true" partition), n_{ij} denotes the number of common nodes of groups c_i and p_j: $n_{ij} = |c_i \cap p_j|$. The normalized mutual information (NMI) is described as [23]

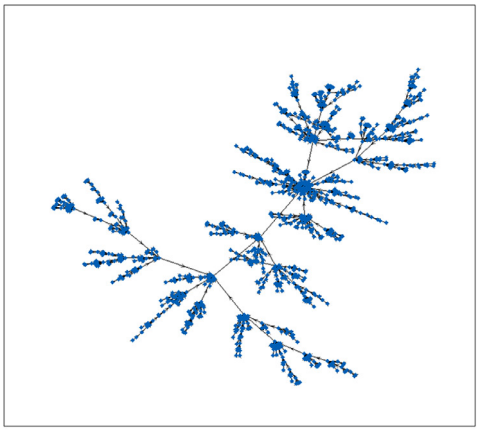
$$NMI = \frac{2\sum_i \sum_j n_{ij} \log \frac{n_{ij} n}{b_i d_j}}{-\sum_i b_i \log \frac{b_i}{n} - \sum_j d_j \log \frac{d_j}{n}}.$$

If the detection result is close to the "true" partition, then the NMI value is high. The modularity is used to evaluate the compactness within the obtained communities, which is described as [19]

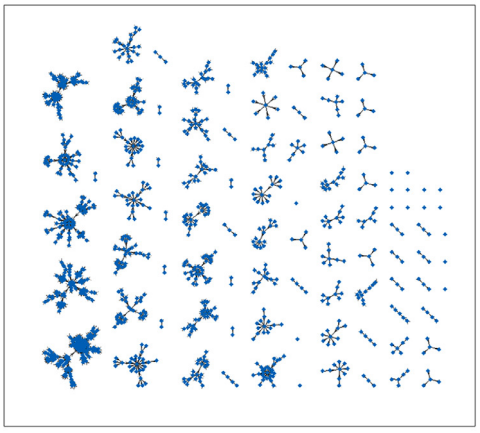
$$Q = \frac{1}{2|E|} \sum_{i,j} \left[A_{ij} - \frac{d(v_i)d(v_j)}{2|E|} \right] I(s_i, s_j),$$



(a)



(b)



(c)

Fig. 7. Email communication: (a) original network; (b) simplified network; (c) cut result.

where s_i is the label of the community which v_i belongs to, $I(s_i, s_j) = 1$, if $s_i = s_j$, otherwise, $I(s_i, s_j) = 0$.

First, we analyze the performance of different algorithms on the synthetic networks which have different degree distributions. Their detection accuracies on these data sets are shown in Table 2. According to the NMI values, we can see that the performances of the STCD algorithm are more robust and better than other algorithms on the synthetic networks. The experimental results illustrate that the proposed algorithm is suitable for dealing with this type of benchmark networks. We also test these algorithms on four real networks. The comparison results are shown in Table 3. We can see that the detection accuracy of the proposed algorithm on the network football is very close to the best result of the other algorithms. On other real networks, the STCD algorithm can obtain the highest NMI values, compared to other algorithms. According to the above analysis, we can see that the STCD algorithm is superior to other algorithms, in terms of detection accuracy.

Furthermore, we compare the modularity values of different algorithms on these given networks. The comparative results are shown in Tables 4 and 5. We can see that the FMM and FUC algorithms can obtain very high modularity values on these data sets. The main reason is that they directly use the modularity measure as their objective function to find out a partition result with the maximum modularity value. Since other algorithms use another objective functions, they have lower modularity values than the FMM and FUC algorithms. However, the experimental analysis in these tables also tells us that a community partition with a high modularity value does not necessarily have high similarity with the truth partition on a network. The main reason is that the modularity is an internal validity measure which mainly evaluates the connectivity within communities. According to Tables 4 and 5, we can see that the modularity values of the proposed algorithm are equal or very close to the highest values of the tested algorithms on these data sets. Therefore, the experimental results illustrate that the communities obtained by the proposed algorithm also have very good internal compactness.

For the STCD algorithm, we need to set the parameter λ. We know that the more the λ value is, the more the number of com-

Table 2
NMI values of different algorithms on synthetical networks.

Data set	FMM	NSC	PLA	FUC	IEDC	MP	STCD
S1	0.73	0.79	0.79	0.95	0.58	0.86	**1.00**
S2	0.63	0.70	0.63	0.95	0.76	0.80	**1.00**
S3	0.55	0.33	0.43	0.92	0.80	0.73	**0.97**
S4	0.77	0.89	0.72	0.95	0.71	0.84	**1.00**
S5	0.65	0.95	0.71	0.93	0.78	0.79	**0.99**
S6	0.52	0.90	0.71	0.67	0.77	0.75	**0.98**

Table 3
NMI values of different algorithms on real networks.

Data set	FMM	NSC	PLA	FUC	IEDC	MP	STCD
Grassweb	0.07	0.06	0.11	0.05	0.05	0.13	**0.23**
Football	0.74	**0.92**	0.71	0.88	0.78	0.69	0.91
Polbooks	0.53	0.35	0.46	0.53	0.22	0.43	**0.56**
Polblogs	**0.37**	0.20	0.33	**0.37**	0.01	0.21	**0.37**

Table 4
Modularity values of different algorithms on synthetical networks.

Data set	FMM	NSC	PLA	FUC	IEDC	MP	STCD
S1	0.32	0.24	0.28	0.34	0.28	0.20	**0.34**
S2	0.27	0.15	0.21	**0.29**	0.26	0.12	**0.29**
S3	**0.23**	0.09	0.16	**0.23**	0.16	0.10	**0.23**
S4	0.32	0.25	0.28	**0.34**	0.32	0.17	**0.34**
S5	0.28	0.20	0.21	**0.29**	0.26	0.13	**0.29**
S6	0.23	0.21	0.16	**0.24**	0.22	0.12	0.23

Table 5
Modularity values of different algorithms on real networks.

Data set	FMM	NSC	PLA	FUC	IEDC	MP	STCD
Grassweb	**0.37**	0.18	0.32	0.35	0.29	0.32	0.35
Football	**0.32**	0.30	0.28	**0.32**	**0.32**	0.24	0.31
Polbooks	**0.35**	0.33	0.29	0.34	0.35	0.30	**0.35**
Polblogs	**0.34**	0.27	0.26	**0.34**	0.22	0.04	**0.34**

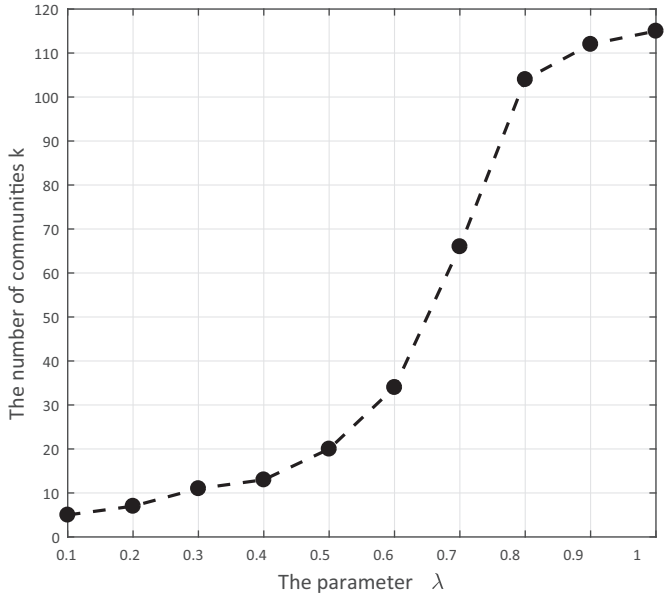


Fig. 8. The parameter λ versus the number of communities *k*.

munities *k* is. Fig. 8 shows the relation between the λ values and the *k* values in Football network. For a network, setting λ should be related to its fill that is the proportion of edges to the total number of possible edges, i.e., m/n^2. A high fill value indicates that each node is closely connected to each other. In this case, we need a big λ value to cut the network. In this paper, we found by the experimental analysis that the STCD algorithm with the parameter λ in the interval [0.3, 0.6] can obtain the number of communities *k* which is very close to the real *k* on these tested networks.

5. Conclusions

In the paper, we present a new community detection algorithm, called STCD. In the new algorithm, we map an original network to a simplified network, i.e., weighted tree or forest, by using the leading and following degrees of nodes. The simplified network provides a very good visual understanding of the community structure. Furthermore, we propose a cutting-tree method to obtain the communities. In the experimental analysis, we show the STCD algorithm is very effective for the visualization of the networks. We also compare the STCD algorithm with other six detection algorithms. The comparison results illustrate that the proposed algorithm can better partition networks, compared to other algorithms.

Acknowledgment

The authors are very grateful to the editors and reviewers for their valuable comments and suggestions. This work is supported by the National Natural Science Foundation of China (Nos. 61773247, 61432011, 61573229, U1435212), the Technology Research Development Projects of Shanxi (Nos. 2015021100, 201601D202036, 201701D221097) and Scientific and Technological Innovation Programs of Higher Education Institutions in Shanxi (No. 2015107).

References

[1] M.J. Barber, J.W. Clark, Detecting network communities by propagating labels under constraints., Phys.Rev. E 80 (2009) 026129.
[2] V.D. Blondel, J.L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. 10 (2008) P10008.
[3] T. Chakraborty, S. Srinivasan, N. Ganguly, A. Mukherjee, S. Bhowmick, Permanence and community structure in complex networks, ACM Trans. Knowl. Discovery Data 11 (2) (2016) 14:1–14:34.
[4] T. Chakraborty, S. Srinivasan, N. Ganguly, A. Mukherjee, S. Bhowmick, Genperm: a unified method for detecting non-overlapping and overlapping communities, IEEE Trans. Knowl. Data Eng. 28 (8) (2016) 2101–2114.
[5] J. Chen, Y. Saad, Dense subgraph extraction with application to community detection, IEEE Trans. Knowl. Data Eng. 24 (7) (2012) 1216–1229.
[6] H. Djidjev, M. Onus, Scalable and accurate graph clustering and community structure detection, IEEE Trans. Parallel Distrib. Syst. 24 (5) (2013) 1022–1029.
[7] J. Duch, A. Arenas, Community detection in complex networks using extremal optimization, Phys.Rev. E 72 (2) (2005) 027104.
[8] S. Fortunato, D. Hric, Community detection in networks: a user guide, Phys. Rep. 659 (2016) 1–44.
[9] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (3–5) (2010) 75–174.
[10] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. U.S.A. 99 (2002) 7821–7826.
[11] Z. Habil, S. P, G. A, R. Brinkman, Data reduction for spectral clustering to analyze high throughput flow cytometry data, BMC Bioinf. 11 (1) (2010) 403.
[12] M. He, M. Leng, F. Li, A node importance based label propagation approach for community detection, Knowl. Eng. Manage. 214 (2014) 249–257.
[13] M.M.D. Khomami, A. Rezvanian, M.R. Meybodi, Distributed learning automata-based algorithm for community detection in complex networks, Int. J. Mod. Phys. B 30 (8) (2016) 1650042.
[14] KONECT, Network dataset, 2015, http://konect.uni-koblenz.de/networks.
[15] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Phys. Rev. E 78 (2008) 046110.
[16] D.D. Lee, H.H. Seung, Algorithms for non-negative matrix factorization, in: Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference, MIT Press, 2001, pp. 556–562.
[17] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, 1967, pp. 281–297.
[18] H. Mahdi, H. Zare, H. Bobarshad, IEDC: An integrated approach for overlapping and non-overlapping community detection, Knowl. Based Syst. 123 (2017) 188–199.
[19] M. Newman, Fast algorithm for detecting community structure in networks, Phys. Rev. E 69 (6) (2004) 066133.
[20] M. Newman, Network dataset, 2015, http://www-personal.umich.edu/~mejn/netdata/.
[21] M.E.J. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. U.S.A. 103 (23) (2006) 8577–8582.
[22] M.A. Porter, O. J.-P, P.J. Mucha, Communities in networks, Not. Am. Math. Soc. 56 (1082–1097) (2009) 1164–1166.
[23] W.H. Press, T.S. A, V.W. T, B.P. Flannery, Numerical Recipes: The Art of Scientific Computing, (3rd ed.), Cambridge University Press, New York, 2007. Section 14.7.3. Conditional Entropy and Mutual Information
[24] U.N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, Phys. Rev. E 76 (2007) 036106.
[25] M. Rosvall, C.T. Bergstrom, Maps of random walks on complex networks reveal community structure, Natl. Acad. Sci. USA 105 (2008) 1118–1123.
[26] M. Rosvall, C.T. Bergstrom, An information-theoretic framework for resolving community structure in complex networks, Natl. Acad. Sci. USA 104 (18) (2007) 7327–7331.
[27] P. Sarkar, A.W. Moore, Dynamic social network analysis using latent space models, in: SIGKDD Explorations, Special Issue on Link Mining, 2005, pp. 31–40.
[28] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 888–905.
[29] C.L. Staudt, H. Meyerhenke, Engineering parallel algorithms for community detection in massive networks, IEEE Trans. Parallel Distrib. Syst. 27 (1) (2016) 171–184.
[30] S.H. Strogatz, Exploring complex networks, Nature 410 (6825) (2001) 268–276.
[31] L. Subelj, M. Bajec, Unfolding communities in large complex networks: combiningdefensive and offensive label propagation for core extraction, Knowl. Eng. Manage. 83 (2011) 036103.
[32] L. Tang, W. X., H. Liu, Community detection via heterogeneous interaction analysis, Data Min. Knowl. Discovery 25 (2012) 1–13.
[33] D. Wang, L. T., Z. S., C. Ding, Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization, in: The 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, USA: ACM Press, New York, 2008, pp. 307–314.
[34] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufman, San Francisco, 2005.
[35] L. Yang, J. D., W. X., X. Cao, Active link selection for efficient semi-supervised community detection, Sci. Rep. 5 (2015) 9039.