Information Sciences 388–389 (2017) 37–47

Contents lists available at ScienceDirect

Information Sciences

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

Fast graph clustering with a new description model for community detection

Liang Bai[a,b,c,*], Xueqi Cheng[b], Jiye Liang[a], Yike Guo[c]

[a] School of Computer and Information Technology, Shanxi University, Taiyuan, 030006, Shanxi, China
[b] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China
[c] Department of Computing, Imperial College London, SW7, London, United Kingdom

ARTICLE INFO

Article history:
Received 13 October 2015
Revised 16 December 2016
Accepted 7 January 2017
Available online 9 January 2017

Keywords:
Graph clustering
Community detection
Community description model
Evaluation criterion
Iterative algorithm

ABSTRACT

Efficiently describing and discovering communities in a network is an important research concept for graph clustering. In the paper, we present a community description model that evaluates the local importance of a node in a community and its importance concentration in all communities to reflect its representability to the community. Based on the description model, we propose a new evaluation criterion and an iterative search algorithm for community detection (ISCD). The new algorithm can quickly discover communities in a large-scale network, due to the average linear-time complexity with the number of edges. Furthermore, we provide an initial method of input parameters including the number of communities and the initial partition before algorithm implementation, which can enhance the local-search quality of the iterative algorithm. The proposed algorithm with the initial method is called ISCD+. Finally, we compare the effectiveness and efficiency of the ISCD+ algorithm with six representative algorithms on several real network data sets. The experimental results illustrate that the proposed algorithm is suitable to address large-scale networks.

1. Introduction

Cluster analysis is a branch in statistical multivariate analysis and unsupervised machine learning. The goal of clustering is to group a set of objects into clusters so that the objects in the same cluster have high similarity but are very dissimilar with objects in other clusters [14]. To solve this problem, various types of clustering algorithms, such as partitional clustering and hierarchical clustering, have been proposed in the literature (e.g., [18] and references therein). Recently, increasing attention has been paid to analyzing cluster structures in complex networks since the data are modeled as networks in many complex systems [38], e.g., social networks and biological networks. In network analysis, cluster structure is also called "community structure" [10,30] which has been shown to be an 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 [11].

[*] Corresponding author.
E-mail addresses: sxbailiang@hotmail.com (L. Bai), cxq@ict.ac.cn (X. Cheng), ljy@sxu.edu.cn (J. Liang), y.guo@imperial.ac.uk (Y. Guo).

http://dx.doi.org/10.1016/j.ins.2017.01.026

To resolve the community detection problem, various graph clustering approaches have been developed, including latent space models, non-negative matrix factorization, block model approximation, spectral clustering, label propagation, and modularity maximization. According to applications for different scientific needs, these models have different definitions of communities or clustering criteria [40]. Latent space models [36] mainly map 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 [24] and linkage [42]. Like the latent space models, non-negative matrix factorization models [22,41,44] transfer the adjacency matrix of a network into a low-dimensional matrix, then cluster it by k-means or linkage. Block model approximations see 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 [7]. Each block represents a community. Spectral clustering models [13,37] view the community detection as a graph partitioning, which apply spectral analysis to obtain the cut minimization. Label propagation models mainly use 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. [32]. 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 [2,16,39]. Modularity maximization models [8,9,11,26,29] transform 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. [5] 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. Additionally, Rosvall and Bergstrom develop an information-theoretic model for community structure [34]. They transfer the problem of community detection into a information coding problem. Furthermore, an information map algorithm of random walks [35] is proposed to solve the optimization problem. There are several studies about the survey of the performance of the existing community detection algorithms, such as [3,15]. The authors compared the performance of these algorithms in real networks and analyzed the strength and weakness of each algorithm.

Many existing community detection models have been successfully applied to different areas. However, there are two main problems in the clustering process. One problem is that most of these models need expensive computing costs including the transformation of a network into a $n \times k$ matrix or hierarchical clustering strategy. These costs limit their efficiency in dealing with large-scale networks. The other is that there is a lack of an effective community description model which is used to summarize and characterize the community. After obtaining the model, we can quickly determine whether a node belongs to the community based on a similarity or distance measure. This can enhance the expandability of community detection for new input nodes in a network. In cluster analysis, the k-prototypes-type algorithms, such as k-means, are a kind of partitional clustering technique and well known for efficiently clustering large-scale spherical data sets. They usually use a virtual or real point on a given data set to represent a cluster. Unfortunately, these types of algorithms mainly deal with object-attribute data sets. It is very difficult for a network without the feature space information to compute the center of a community to represent it. Currently, many algorithms are proposed to transform a network into an object-attribute data set and cluster it by the traditional clustering algorithms. However, feature extraction maybe lead to information loss and high transformation costs. Therefore, it is an important issue to directly handle the raw network data. However, few scholars have discussed how to directly describe a community by using nodes in the k-prototypes-type clustering.

Motivated by the above idea, we design a new k-prototypes-type algorithm for quickly clustering large-scale networks. The major contributions of this paper are as follows:

- Unlike the traditional k-prototypes-type algorithms, we propose a community description model, which does not make use of a node but multiple nodes with different weights (representability) to represent the community. The new description model can sufficiently reflect the characteristics of the communities.
- Based on the description model, we provide a community detection criterion. It evaluates the quality of a partition result from two aspects: the external-link separation among communities and the internal-link compactness within communities.
- Based on the evaluation criterion, we propose an iterative search algorithm for community detection (ISCD) which applies a local search to partition a network into k communities. The new algorithm inherits the advantage of the k-prototypes-type clustering. Namely, the algorithm can address large-scale networks, since its average time complexity is linear with the number of edges.
- Like the traditional k-prototypes-type algorithms, the ISCD algorithm needs initial parameters, including initial partition and the number of communities. We propose an initial method for these parameters to enhance the local-search quality of the ISCD algorithm. The new algorithm with the initial method is called ISCD+.

The following is the outline of this paper. Section 2 introduces a community description model. Section 3 presents an evaluation criterion and an iterative algorithm for community detection. Section 4 provides an initial method for parameters

Table 1
Description of the main symbols used in this paper.

Symbol	Description
$G =< V, E >$	A network including a vertex set V and a edge set E
n	The number of vertices in G
m	The number of edges in G
v_i	The ith vertex in V
NG_i	A vertex set including the neighbors of v_i
d_i	The degree of v_i
Ω	A partition of G
V_l	The lth community in Ω
n_l	The number of vertices in V_l
k	The number of communities in Ω
d_{li}	The number of link edges between v_i and V_l
f_{li}	The local importance of v_i in V_l
δ_i	The importance concentration of v_i
R	A matrix including all the community description models
R_l	The lth community description model
R_{li}	The representability of v_i to V_l
$Sim(.,.)$	The similarity measure between two nodes
$M(.,.)$	The similarity measure between a node and a community
$WES(\Omega)$	The evaluation criterion of the external-link separation
$WIC(\Omega)$	The evaluation criterion of the internal-link compactness
$F(\Omega)$	The objective function of community detection
$P_l(v_i)$	The possibility of v_i as an exemplar of V_l

setting. Section 5 demonstrates the performance and scalability of the proposed algorithm. Finally, we draw conclusions and suggest future work in Section 6.

2. The community description model

Suppose that $G = \langle V, E\rangle$ is an undirected network with V which is a set of n vertices and E which is a set of m edges. $NG_i = \{v_j| < v_i, v_j >\in E\}$ is a vertex set including all the neighbors of $v_i \in V$. The degree of node v_i is $d_i = |NG_i|$. $\Omega = \{V_1, V_2, \ldots, V_k\}$ is a partition of V, where V_l is the lth community including n_l nodes and k is the number of communities. $d_{li} = |\{v_j| < v_i, v_j >\in E, v_j \in V_l\}|$ is the number of edges between v_i and all the nodes in V_l. The main symbols used in this paper are summarized in Table 1.

Due to the lack of the feature space in a network, we cannot compute the center of a community by k-means to represent it. Therefore, we hope to use several nodes with different weights (i.e., representability) to describe the community. It is a key issue on how to measure the representability of a node to a community. We consider the following two factors to evaluate the representability.

- *Local importance*: the more neighbors of a node occur in the community, the more important the node is for the community;
- *Importance concentration*: if a node has only high “local importance” in the community rather than other communities, the node has high representability to it.

“Local importance” characterizes the internal importance of a node in a community, which is formally defined as

$$f_{li} = \frac{d_{li}}{n_l}, \tag{1}$$

where i is the tag of node v_i and l is the tag of community V_l. We have $0 \le f_{li} \le 1$. If the f_{li} value is high, the associativity between node v_i and community V_l is large. Thus, f_{li} is used to measure the representability of node v_i to community V_l. However, we can not only consider the “local importance” of a node to reflect its representability to a community. For example, Fig. 1 shows three communities (A, B and C). In community B, nodes 5 and 6 have the same “local importance”, i.e., $f_{25} = f_{26} = \frac{4}{5}$. In fact, node 6 has two edges from community C, except for community B. Node 5 only has edges from community B. Therefore, node 5 has better representability than node 6 to community B.

Furthermore, we need to continue the example in Fig. 1 to explain why not measure the “neighbor concentration” but the “importance concentration” of a node while evaluating its representability. According to the figure, we see that nodes 3 and 6 both have four edges from community B and two edges from other communities. If we consider the “concentration” of their edges, nodes 3 and 6 have the same “concentration”. However, since the number of nodes in community A are more than community C, the “local importance” of node 6 in community A is less than that of node 3 in community C. Thus, we think that the representability of node 6 to community B is higher than that of node 3.

According to the above example, we see that the “importance concentration” of a node is a very important factor for measuring its representability to a community. Next, we introduce how to measure the “importance concentration” of a

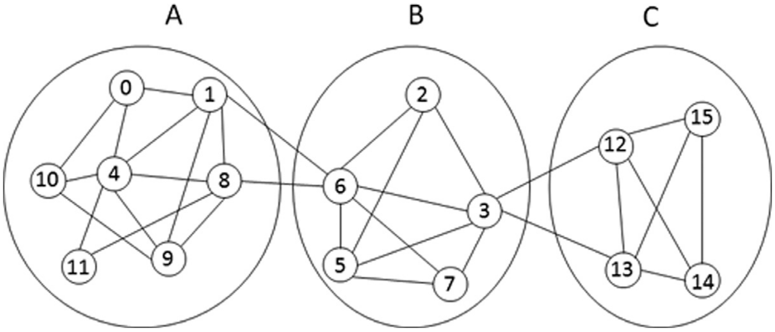

Fig. 1. The communities *A*, *B* and *C* in an example.

node. The "importance concentration" characterizes the distribution of the "local importance" of a node in all the communities. We make use of the complementary entropy [23] to evaluate the "importance concentration" of a node, which is formalized as

$$H(X)=\sum_{l=1}^{k}p_l(1-p_l),$$

where X is a variable with k values and p_l is the occurrence probability of the lth value. The less the $H(X)$ value is, the more certain the value of the variable is. Here, let $X=\{f_{1i},f_{2i},\ldots,f_{ki}\}$ be a set including the k "local importance" of node v_i and $p_l=\frac{f_{li}}{\sum_{j=1}^{k}f_{ji}}$ be the proportion of the lth "local importance" from all the "local importance" of v_i. We have

$$\begin{aligned}H(X)&=\sum_{l=1}^{k}\frac{f_{li}}{\sum_{j=1}^{k}f_{ji}}\left(1-\frac{f_{li}}{\sum_{j=1}^{k}f_{ji}}\right)\\&=1-\sum_{l=1}^{k}\left(\frac{f_{li}}{\sum_{j=1}^{k}f_{ji}}\right)^2.\end{aligned}$$

Therefore, we define an evaluation measure for the "importance concentration" of node v_i as

$$\delta_i=\sqrt{\sum_{l=1}^{k}\left(\frac{f_{li}}{\sum_{j=1}^{k}f_{ji}}\right)^2},\tag{2}$$

where $\sqrt{1/k}\le\delta_i\le 1$. We can see that $H(X)$ is inversely proportional to δ_i. The larger the δ_i value is, the greater the "local importance" of the node is concentrated in a few communities.

Based on the above two factors, we devise a representability measurement for node v_i to community V_l, which is defined as

$$R_{li}=f_{li}\delta_i.\tag{3}$$

For node v_i, the larger its f_{li} and δ_i values are, the more its representability to community V_l is. Note that if f_{li} is equal to 0, the representability of v_i is 0. This indicates that we only select the nodes which have edges in V_l to represent it. Furthermore, we suppose that $R=[R_{li}]_{k\times n}$ is a matrix including the k description community models and R_l is the lth row of R which denotes the description model of the lth community.

3. The community detection algorithm

Based on the new community description model, we want to find an effective partition with k communities from a network. The "effectiveness" denotes the quality of the communities. It is thought that a good partition should include such communities that have high external-link separation and internal-link compactness.

First, we discuss how to evaluate the external-link separation among communities. For node v_i, δ_i reflect not only its "importance concentration" but also the difference of its representability among communities. If the "local importance" of node v_i to each community is similar, the description models of these communities are also similar from the view of v_i. Therefore, we use the difference of the description models to evaluate the external-link separation among communities. An evaluation criterion is defined as

$$WES(\Omega)=\sum_{i=1}^{n}\gamma_i\delta_i,\tag{4}$$

where γ_i is the weight of node v_i which is used to reflect the importance the node played in evaluating the partition Ω. *WES*(Ω) is viewed as a weighted evaluation for the difference among communities. The larger the *WES*(Ω) value is, the more different the structures of communities are to each other. If each γ_i is set to 1, each node is thought to have equal weight in evaluating the separation. In this case, the evaluation criterion is rewritten as

$$ES(\Omega) = \sum_{i=1}^{n} \delta_i. \tag{5}$$

Next, we provide an evaluation criterion of internal-link compactness in a community, which is defined as

$$C_l = \sum_{v_i \in V_l} \sum_{v_j \in V_l} \frac{Sim(v_i, v_j)}{n_l}, \tag{6}$$

where *Sim*(., .) is a similarity measure between two nodes. Here, we use the number of common neighbors between two nodes to simply reflect their similarity, which is formalized as

$$Sim(v_i, v_j) = |NG_i \cap NG_j|. \tag{7}$$

The more the number of their common neighbors is, the more similar they are. C_l is the sum of the average similarity between all the nodes in the community V_l. If the C_l value is large, the internal-link compactness in the community is strong. For a partition of the network, the overall internal-link compactness is defined as

$$IC(\Omega) = \sum_{l=1}^{k} C_l. \tag{8}$$

The larger the *IC*(Ω) value is, the stronger its internal-link compactness is.

Furthermore, we have

$$IC(\Omega) = \sum_{i=1}^{n} \sum_{l=1}^{k} \frac{d_{li}^2}{n_l} = \sum_{i=1}^{n} I_i, \tag{9}$$

where $I_i = \sum_{l=1}^{k} d_{li} f_{li}$, $1 \le i \le n$. From the view of node v_i, I_i reflects the internal similarity within communities. To maximize I_i, we should assign most of its neighbors into the community with its highest "local importance", since

$$\max \sum_{l=1}^{k} d_{li} f_{li} \le d_i \max_{l=1}^{k} f_{li},$$

where $d_i = \sum_{l=1}^{k} d_{li}$. Based on this analysis, we see that *IC*(Ω) is an unweighted evaluation criterion, i.e., each node plays an equal role in measuring the effectiveness of the partition. However, each node in the network often has a different importance. Therefore, we embed a weight for each node into the criterion which can be rewritten as

$$WIC(\Omega) = \sum_{i=1}^{n} w_i I_i, \tag{10}$$

where w_i is a weight of node v_i which is used to reflect the importance of the node. If w_i is set to 1 for $1 \le i \le n$, *WIC*(Ω) is equal to *IC*(Ω).

According to the definitions of *WES* and *WIC*, we see that each of them consider only one of the two factors of "local importance" and "importance concentration" in measuring the effectiveness of a partition. Therefore, we integrate them into an evaluation criterion which is described as

$$F(\Omega) = \sum_{i=1}^{n} \delta_i I_i. \tag{11}$$

The relations of *F*(Ω), *WES*(Ω) and *WIC*(Ω) are as follows.

- If we set $\gamma_i = I_i$ for $1 \le i \le n$, $F(\Omega) = WES(\Omega)$. In this case, *F*(Ω) can be seen as a weighted criterion of the external-link separation.
- If we set $w_i = \delta_i$ for $1 \le i \le n$, $F(\Omega) = WIC(\Omega)$. In this case, *F*(Ω) can be seen as a weighted criterion of the internal-link compactness.

These relations tell us that *F*(Ω) simultaneously considers the external-link separation among communities and the internal-link compactness within communities. Therefore, we select *F*(Ω) as our objective function in detecting communities and try to maximize it.

Next, we design an iterative search algorithm for community detection to locally search a partition of a network in order to maximize *F*(Ω). We name the algorithm as ISCD, which is described in the steps below.

Step 1. Input the number of communities k, the initial model R, and the maximum number of iterations t_m, and set $t = 0$;
Step 2. Assign each node into a community based on a similarity measure (which will be introduced later) and obtain a new partition Ω;
Step 3. Update the description model R, according to Section 2;
Step 4. If $t \geq t_m$ or the $F(\Omega)$ value does not change, then stop; otherwise set $t = t + 1$ and goto Step 2.

In Step 2, we need a similarity measure to assign each node into a community, which is defined as

$$M(v_i, R_l) = \sum_{v_j \in NG_i} R_{lj}. \tag{12}$$

$M(v_i, R_l)$ is the sum of the representability of all the nodes linked with v_i to community V_l. We suppose that if the neighbors of node v_i have high representability to community V_l, the node very possibly belongs to the community. Therefore, we use $M(v_i, R_l)$ to evaluate the similarity between node v_i and community V_l. Because

$$\begin{aligned}\sum_{l=1}^{k} \sum_{v_i \in V_l} M(v_i, R_l) &= \sum_{l=1}^{k} \sum_{i=1}^{n} d_{li} R_l \\ &= F(\Omega),\end{aligned} \tag{13}$$

we apply the following rule to determine the assignment of node v_i:

$$L(v_i) = \arg \max_{l=1}^{k} M(v_i, R_l), \tag{14}$$

where $L(v_i)$ is a community label of node v_i.

Furthermore, we analyze the average and worst computational complexity of the algorithm. The basic operation of the proposed algorithm is $M(.,\ .)$ whose computational cost is $\bar{d}$ which is the average degree of the nodes in a network. Since $\bar{d} = 2m/n$, the average computational complexity of the proposed algorithm is $O(kn\bar{d}t)$ or $O(kmt)$. For a network, since the maximal possible number of edges m is qual to n^2, the worst computational complexity is $O(kn^2 t)$. If k is set to $\sqrt{n}$, which is the maximum possible number of clusters [4], the worst complexity becomes $O(n^{2.5}t)$. However, since $m \ll n^2$ and $k \ll \sqrt{n}$ in most of the real networks, the proposed algorithm is very efficient for dealing with large-scale data.

4. The setting of initial parameters

The proposed algorithm uses a local search to obtain a partition of a network. We need to set an initial R which the algorithm performance depends on. Therefore, we propose an initial method to enhance the local-search effectiveness of the algorithm.

In this method, we first hope to select k nodes with very high representability as exemplars. We suppose that an exemplar represents a community. Based on the similarity between nodes and exemplars, we assign them into initial communities whose description model R is the initial parameter of the iterative algorithm. In the initial method, selecting exemplars is a key issue. For this, we use two indices to evaluate the representability of a node. The one is the degree d_i of node v_i. The larger d_i is the greater the number of its neighbors. The other is the similarity measure $Sim(.,\ .)$. We select the first k nodes with a high degree and low similarity with other exemplars. Therefore, we define a possibility of the node v_i as the lth exemplar as follows.

$$P_l(v_i) = \begin{cases} d_i, & l = 1, \\ \dfrac{d_i}{\max_{j=1}^{l-1} Sim(v_i, e_j) + 1}, & l > 1, \end{cases} \tag{15}$$

where e_j is the jth exemplar obtained, $1 \leq j \leq l - 1$. According to the definition, we select the node with the highest degree as the first exemplar. While determining whether a node is the lth exemplar ($l > 1$), the possibility P_l of the node is proportional to its degree and inversely proportional to its maximum similarity with the first $l - 1$ exemplars. The larger its P_l value is, the more possible it becomes the lth exemplar. The initial method is described in the following steps.

Step 1. Input the desired number of communities k, and set $l = 1$ and $Ex = \emptyset$;
Step 2. If $l \leq k$, then choose node e_l from all the nodes as the lth exemplar, where e_l satisfies

$$P_l(e_l) = \max_{v_i \in V - Ex} P_l(v_i),$$

then, set $Ex = Ex \cup \{e_l\}$ and $l = l + 1$ and goto Step 2, otherwise goto Step 3;

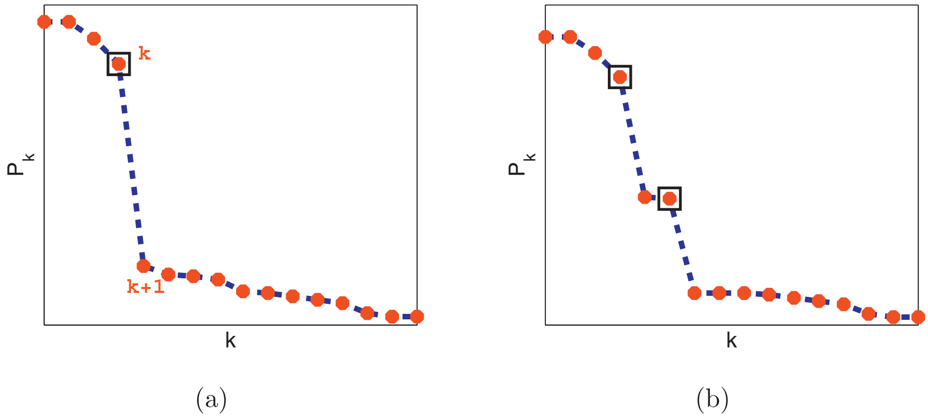

Fig. 2. (a) Obtain a candidate of k; (b) Obtain multi candidates of k.

Step 3. For each node v_i, we assign it to the community V_r which satisfies

$$r = \arg\max_{l=1}^{k} Sim(v_i, e_l);$$

Step 4. Compute the initial R of the initial partition.

The basic operation of the proposed method is $Sim(.,\ .)$ whose computational cost is also $\bar{d}$. The average computational complexity of the proposed method is $O(n\bar{d}k^2)$ or $O(mk^2)$. The ISCD algorithm with the initial method is called ISCD+.

In addition, we need to input the desired number of communities k in the ISCD+ algorithm. In most cases, k is set by the user's experience and knowledge. However, if k is unknown, we can evaluate k by the selected possible exemplars. Suppose that the best community structure has k communities in a network. After obtaining k exemplars, if we continue selecting a node as the $k+1$ potential exemplar e_{k+1}, intuitively, the node may be representative of the same community as one of the first k exemplars. In this case, $P_{k+1}(e_{k+1}) = \max_{v_i \in V} P_{k+1}(v_i)$ should be far smaller than $P_l(e_l)$, $(1 \le l \le k)$. This means that the values from k to $k+1$ should have a dramatic change. However, the values from $k+1$ to $k+q(k+q \le n)$ should be much less distinctive, because these chosen $k+2, k+3, \ldots, k+q$ potential exemplars also may be representative of the same communities as the first k exemplars. This heuristic result tells us that the value of $P_l(e_l)$ could reflect the possibility of the lth community existing. The higher the value is, the more probable the lth community exists. Therefore, we can explore the changes of the P values with different k to find the number of communities. Fig. 2 shows a curve of P_l values. While the change of $P_k(e_k)$ to $P_{k+1}(e_{k+1})$ is dramatic, the curve from $k+1$ goes into a plateau. In this case, we consider k as a candidate of the number of communities.

In Fig. 2(a), we view only one obvious point. However, while dealing with real networks, we often discover many non-ideal cases. For example, the curve may have more than one obvious point. In Fig. 2(b), there are two obvious points 4 and 6. This curve implies that the number of communities may not be unique. All the obvious points in function P_l can be as candidates of k. In addition, we may see that there are not any obvious points in a curve. In that case, we should set a threshold λ. If $P_l(e_l) > \lambda$ and $P_{l+1}(e_{l+1}) \le \lambda$, it is thought that the number of communities is l.

5. Experimental analysis

5.1. Experimental environment

In this section, we present two experiments to evaluate the performance of the ISCD+ algorithm on ten real networks. These networks are downloaded from [20,27] and shown in Table 2. In the first experiment, we test the detection accuracy of the algorithm on the first four networks. Since the tested networks include the "true" class labels, we compare the similarity between the detection result and the "true" partition in each network. In the second experiment, we analyze the scalability of the algorithm. The hardware environment of these experiments is a PC with an Intel 2.5 Hz i7-4710MQ CPU and 16 GB RAM. The software platform is Matlab 2012b in Windows 7 x64.

In these experiments, we compare the ISCD+ algorithm with six other algorithms including the fast modularity maximization (FMM) [26], the normalized spectral clustering (NSC) [37], the nonnegative matrix factorization (NMF) [22], the label propagation algorithm (LPA) [32], the fast unfolding communities (FUC) [5] and the infomap of random walks (INF) [35]. Except for the INF algorithm, these algorithms are coded in Matlab 2012b. For the INF algorithm, we only found its C++ code from the homepage of Martin Rosvall.

Table 2
The real networks.

Data sets	Vertices	Edges	Communities
Polbooks [21]	105	441	3
Adjnoun [28]	112	425	2
Football [11]	115	613	12
Polblogs [1]	1,490	16,718	2
Email [12]	1,133	5,451	NA
Reactome [19]	6,327	147,547	NA
PGP [6]	10,680	24,316	NA
DBLP [43]	317,080	1,049,866	NA
Amazon [43]	334,863	925,872	NA
LiveJournal [25]	5,204,176	49,174,620	NA

Table 3
Notation for the contingency table for comparing two partitions.

C\P	p_1	p_2	$\cdots$	$p_{k'}$	*Sums*
c_1	n_{11}	n_{12}	$\cdots$	$n_{1k'}$	b_1
c_2	n_{21}	n_{22}	$\cdots$	$n_{2k'}$	b_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
c_k	n_{k1}	n_{k2}	$\cdots$	$n_{kk'}$	b_k
Sums	d_1	d_2	$\cdots$	$d_{k'}$	

5.2. Detection accuracy

We test the detection accuracy of the ISCD+ algorithm on the four networks with labels. We employ the two widely-used criteria to measure 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), the overlappings between C and P can be summarized in a contingency table (Table 3) where n_{ij} denotes the number of common nodes of groups c_i and p_j: $n_{ij} = |c_i \cap p_j|$. The adjusted rand index [33] is defined as

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i\binom{b_i}{2}\sum_j\binom{d_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i\binom{b_i}{2} + \sum_j\binom{d_j}{2}] - [\sum_i\binom{b_i}{2}\sum_j\binom{d_j}{2}]/\binom{n}{2}} \qquad (16)$$

where n_{ij}, b_i, d_j are values from the contingency table (Table 3). The normalized mutual information (NMI) [31] is defined as

$$NMI = \frac{2\sum_i\sum_j n_{ij}\log\frac{n_{ij}n}{b_i d_j}}{-\sum_i b_i\log\frac{b_i}{n} - \sum_j d_j\log\frac{d_j}{n}}. \qquad (17)$$

If the detection result is close to the “true” partition, then their ARI and NMI values are high. For the NSC and NMF algorithms, we do not select the k-means but the ward-linkage algorithm which is a representative hierarchical clustering algorithm to cluster the transformed data sets. Compared to k-means, the ward-linkage does not need initial centers and can obtain a global-search result. For the number of communities k, the FMM, LPA, FUC and INF algorithm can automatically give its value. In the NSC, NMF and ISCD algorithm, we apply the proposed initial method in Section 4 to estimate the k value. For a network, if we obtain more than one candidate of k, we select their maximum value as k. According to Fig. 3, we see that the numbers of communities on Polbooks and Polblogs are estimated correctly. On Adjnoun, since the first exemplar has a much larger P value than the other exemplars, the representability of other exemplars is very elusive. In this case, we set $k = 2$ by default. On Football, the candidates of k are 6, 9 and 13, which are inconsistent with the true number of communities. For this, we set $k = 13$ for the network. These estimated results indicate that the estimation method only plays a supporting role in determining k. The detection accuracies of different algorithms are shown in Table 4. According to the validity measurement values, we can see that the proposed algorithm is obviously better than the FMM, NSC, NMF and PLA algorithms on these given data sets. On Polbooks, the INF algorithm is slightly better than the proposed algorithm. On Football and Polblogs, the detection accuracy of the proposed algorithm is indistinguishable from the best result of the other six algorithms. Based on comparing these results, we see that although the proposed algorithm cannot guarantee obtaining the highest ARI and NMI values on each of these data sets, its clustering results are superior or close to the best results of the other six algorithms. These experiments illustrate the proposed algorithm is very effective in clustering these networks.

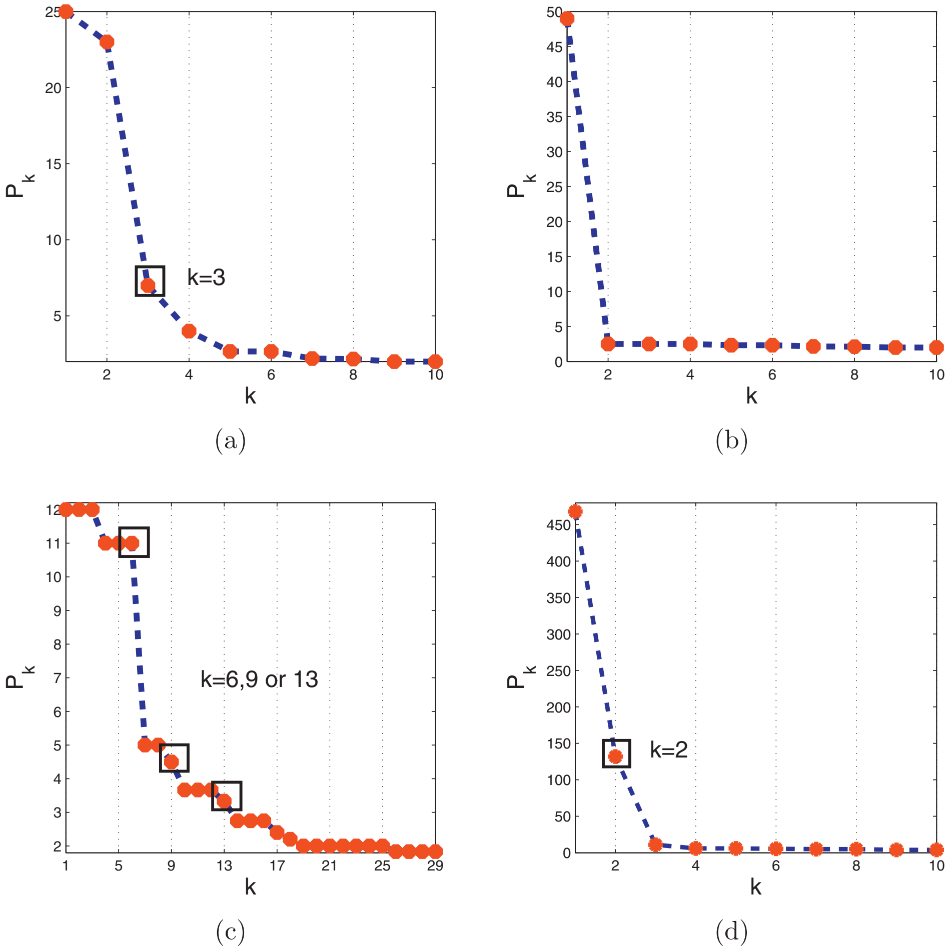

Fig. 3. Estimating the numbers of communities. (a) Polbooks; (b) Adjnoun; (c) Football; (d) Polblogs.

Table 4
The compared results of different algorithms.

Data set	Index	FMM	NSC	NMF	PLA	FUC	INF	ISCD+
Polbooks	ARI	0.6328	0.6321	0.0378	0.5191	0.6280	**0.6463**	0.6390
	NMI	0.5342	0.5339	0.2397	0.5107	0.6280	**0.5369**	0.5245
Adjnoun	ARI	−0.0074	0.0063	0.0003	0.0000	−0.0109	0.0035	**0.5319**
	NMI	0.0014	0.0020	0.0694	0.0000	0.0056	0.0321	**0.4387**
Football	ARI	0.4325	0.8255	0.8689	0.3943	0.7921	**0.8967**	0.8868
	NMI	0.7160	0.9087	0.9105	0.7187	0.8813	0.9242	**0.9263**
Polblogs	ARI	0.0000	0.0238	0.0000	0.4246	**0.5108**	0.4533	0.4812
	NMI	0.0014	0.1265	0.0014	0.3276	0.3704	0.3399	**0.4402**

5.3. Detection scalability

We analyze the efficiency of the FMM, NSC, NMF, LPA, FUC, INF and ISCD+ algorithms on the last six networks in Table 2. For NSC, NMF and ISCD+ algorithms, we continue applying the proposed initial method to estimate the k value. The number of clusters k on the six networks is estimated as 5, 6, 4, 5, 12 and 5, respectively. When comparing the efficiency of the NSC and NMF algorithms, we use the k-means algorithm for them, instead of the ward-linkage algorithm. This is because the k-means algorithm is more suitable for dealing with large-scale data sets [17] compared to the ward-linkage algorithm.

Table 5 shows the running times of the algorithms on these given networks. We note that if the scale of a data set is large, some algorithms cannot obtain the clustering results within an acceptable time. Therefore, in our experiment, if the running time of an algorithm is greater than 10 hours, we terminate it and set the running time as ‘NA’. According to Table 5, we see that the FMM and NSC algorithms only can address the smaller networks, such as Email and Reactome which have less than 10,000 nodes. On the first three networks, the clustering speed of the LPA algorithm is faster than

Table 5
The running time (seconds) of different algorithms.

Data set	FMM	NSC	NMF	LPA	FUC	INF	ISCD+
Email	16.76	6.93	1.49	1.84	1.55	0.27	0.03
Reactome	4531.30	3800.8	16.78	21.50	51.23	2.73	0.41
PGP	NA	NA	33.97	21.57	1064.10	1.54	0.32
DBLP	NA	NA	NA	NA	NA	240.06	23.77
Amazon	NA	NA	NA	NA	NA	257.77	49.60
LiveJournal	NA	NA	NA	NA	NA	NA	363.34

those of the FMM, NSC, FUC and NMF algorithms. However, when applying the LPA algorithm on the last three large-scale networks, we can not obtain the detection results within 10 hours. On DBLP and Amazon, only the INF and ISCD+ algorithms quickly finished the clustering tasks. We also see that the proposed algorithm is obviously faster than the INF algorithm. While testing these algorithms on LiveJournal, only the running time of the ISCD+ algorithm is acceptable.

We know that the proposed algorithm is a type of the *k*-prototypes algorithms. It inherits the advantage of the *k*-prototypes-type clustering which can quickly address large-scale data. The above experimental results have illustrated that the proposed algorithm is very efficient, compared to the other six algorithms. However, like the *k*-prototypes-type clustering, the proposed algorithm needs additional parameters. The proposed initial method in this paper is heuristic for setting these parameters. Although we cannot guarantee that it is the best method, it is suitable for this new algorithm. The experimental results also have shown that its clustering results are superior or close to the best results of the other six algorithms on the tested data sets. The other six algorithms are mainly types of the hierarchical clustering or feature-extraction clustering algorithms. Compared to the *k*-prototypes clustering, the type of the hierarchical algorithms can obtain a global clustering result. . This result needs less parameters but expensive computational costs. Since everything has two alternatives, we do not determine which one of these algorithms is the best. If users need a clustering result with high accuracy and less parameters, they should select the hierarchical clustering option. However, if users hope to address large-scale data networks, they should select the proposed algorithm.

6. Conclusions

In the paper, we present an iterative search algorithm for community detection (ISCD) on networks. In the proposed algorithm, we provide a community description model that simultaneously considers the “local importance” of a node in a community and the “importance concentration” of the node in all the communities. Furthermore, we propose an initial method to solve setting the input parameters including initial partition and the number of communities. The proposed iterative algorithm with the initial method is called ISCD+. In the experimental analysis, we compare the ISCD+ algorithm with six other detection algorithms. The comparison results illustrate that the ISCD+ algorithm compared to other algorithms can more effectively and efficiently partition large-scale networks. In the future, we would like to further improve the initialization for input parameters. However, the key issue is how to balance between the detection quality and computational cost. Thus, we wish to boost the effectiveness of the proposed algorithm in an acceptable time.

Acknowledgement

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. 61305073, 61432011, 61472400, 61573229, U1435212), the National Key Basic Research and Development Program of China (973)(Nos. 2013CB329404, 2014CB340400), the Foundation of Doctoral Program Research of Ministry of Education of China (No. 20131401120001), the Technology Research Development Projects of Shanxi (No. 2015021100) and Scientific and Technological Innovation Programs of Higher Education Institutions in Shanxi (No. 2014104, 2015107).

References

[1] L.A. Adamic, N. Glance, The political blogosphere and the 2004 US election, in: Proceedings of the WWW-2005 Workshop on the Weblogging Ecosystem, 2005.
[2] M.J. Barber, J.W. Clark, Detecting network communities by propagating labels under constraints, Phys. Rev. E 80 (2009) 026129.
[3] P. Bedi, C. Sharma, Community detection in social networks, in: Wiley Interdisciplinary Reviews Data Mining and Knowledge Discovery, 2016, pp. 496–500. 2174–2177.
[4] J.C. Bezdek, N.R. Pal, Some new indexes of cluster validity, IEEE Trans. Syst. Man Cybern. Part B 28 (3) (1998). 301-15.
[5] V.D. Blondel, J.L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech: Theory Exp. 10 (2008) P10008.
[6] M. Boguna, R. Pastor-Satorras, A. Diaz-Guilera, A. Arenas, Models of social networks based on social distance attachment, Phys. Rev. E 70 (5) (2004) 056122.
[7] J. Chen, Y. Saad, Dense subgraph extraction with application to community detection, IEEE Trans. Knowl. Data Eng. 24 (7) (2012) 1216–1229.
[8] H. Djidjev, M. Onus, Scalable and accurate graph clustering and community structure detection, IEEE Trans. Parallel Distrib. Syst. 24 (5) (2013) 1022–1029.
[9] J. Duch, A. Arenas, Community detection in complex networks using extremal optimization, Phys. Rev. E 72 (2) (2005) 027104.
[10] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (3–5) (2010) 75–174.

[11] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, in: Proceedings of the National Academy of Sciences of the United States of America, volume 99, 2002, pp. 7821–7826.
[12] R. Guimera, L. Danon, A. Diaz-Guilera, F. Giralt, A. Arenas, Self-similar community structure in a network of human interactions, Phys. Rev. E 68 (6) (2003) 065103.
[13] Z. Habil, S.P. G. A., R. Brinkman, Data reduction for spectral clustering to analyze high throughput flow cytometry data, BMC Bioinformatics 11 (1) (2010) 403.
[14] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2001.
[15] S. Harenberg, G. Bello, L. Gjeltema, S. Ranshous, J. Harlalka, R. Seay, et al., Community detection in large-scale networks: a survey and empirical evaluation, Wiley Interdiscipl. Rev. Comput. Stat. 6(6) (2014) 426–439.
[16] M. He, M. Leng, F. Li, et al., A node importance based label propagation approach for community detection, Knowl. Eng. Manag. 214 (2014) 249–257.
[17] T.H. Sarma, P. Viswanath, B. Reddy, A hybrid approach to speed-up the k-means clustering method, Int. J. Mach. Learn. Cybern. 4(2) (2013) 107–117.
[18] A. Jain, R. Dubes, Algorithms for Clustering Data, Prentice Hall, 1988.
[19] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, G. Gopinath, G. Wu, L. Matthews, et al., Reactome: a knowledgebase of biological pathways, Nucleic Acids Res. 33 (suppl 1) (2005) D428–D432.
[20] , KONECT, Network Dataset, 2015. http://konect.uni-koblenz.de/networks.
[21] V. Krebs, Books about US Politics, 2004. http://www.orgnet.com/.
[22] 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.
[23] J. Liang, K. Chin, C. Dang, R. Yam, A new method for measuring uncertainty and fuzziness in rough set theory, Int. J. Gen. Syst. 31 (4) (2002) 331–342.
[24] 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.
[25] A. Mislove, M. Marcon, K.P. Gummadi, P. Druschel, B. Bhattacharjee, Measurement and analysis of online social networks, in: Proc. Internet Measurement Conf, 2007.
[26] M. Newman, Fast algorithm for detecting community structure in networks, Phys. Rev. E 69 (6) (2004) 066133.
[27] M. Newman, Network dataset, 2015. http://www-personal.umich.edu/~mejn/netdata/.
[28] M.E.J. Newman, Finding community structure in networks using the eigenvectors of matrices, Phys. Rev. E 74 (3) (2006) 036104.
[29] M.E.J. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. U.S.A. 103 (23) (2006) 8577–8582.
[30] M.A. Porter, O. J.-P., P.J. Mucha, Communities in networks, Notices Am. Math. Soc. 56 (2009) 1164–1166. 1082–1097
[31] W.H. Press, T.S.A.V.W. T., B.P. Flannery, Section 14.7.3. Conditional entropy and mutual information, Numerical Recipes: The Art of Scientific Computing, third ed., Cambridge University Press, New York, 2007.
[32] 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.
[33] W.M. Rand, Objective criteria for the evaluation of clustering methods, J. Am. Stat. Assoc. 66 (336) (1971) 846–850.
[34] 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.
[35] M. Rosvall, C.T. Bergstrom, Maps of random walks on complex networks reveal community structure, Natl. Acad. Sci. USA 105 (2008) 1118–1123.
[36] 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.
[37] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.
[38] S.H. Strogatz, Exploring complex networks, Nature 410 (6825) (2001) 268–276.
[39] M. Bajec, L. Subelj, Unfolding communities in large complex networks: combining defensive and offensive label propagation for core extraction, Phys. Rev. E 83 (2011) 036103.
[40] L. Tang, W. X., H. Liu, Community detection via heterogeneous interaction analysis, Data Min. Knowl. Discov. 25 (2012) 1–13.
[41] 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, New York, USA: ACM Press, 2008, pp. 307–314.
[42] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufman, San Francisco, 2005.
[43] J. Yang, J. Leskovec, Defining and evaluating network communities based on ground-truth, in: Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics, ACM, 2012, p. 3.
[44] L. Yang, J.D. W. X., X. Cao, Active link selection for efficient semi-supervised community detection, Sci. Rep. 5 (2015) 9039.