Information Sciences 482 (2019) 321–333

Contents lists available at ScienceDirect

Information Sciences

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

A divide and agglomerate algorithm for community detection in social networks

Zhiyuan Liu, Yinghong Ma*

Business School, Shandong Normal University, Shandong, 250014, PR China

ARTICLE INFO

Article history:
Received 4 May 2018
Revised 9 January 2019
Accepted 10 January 2019
Available online 14 January 2019

Keywords:
Community detection
DA algorithm
Constrained AA index
Attraction index

ABSTRACT

Communities, or clusters, are usually subgraphs of nodes densely interconnected but sparsely linked with others. The nodes with similar properties or behaviors are more likely to be in the same community, and vice versa. However, due to the complexity and diversity of networks, the accurate organization or function of communities in many real networks is often extremely difficult to be recognized. Hence, methods for community detection would have immediate impact on understanding the organizations and functions of networks. Therefore, algorithm design becomes a fundamental problem for many networks. In this paper, the local and global information are applied together to propose a divide and agglomerate (DA) algorithm for community detection in social networks. The DA algorithm achieves the result with a two-stage strategy: Dividing a network into small groups according to node pairs' similarities, and merging a group with the other who has the biggest attraction for it until the community criterion is steady. The novel similarity, constrained AA index captures the local and global information ensuring the optimal communities detection. The results of experiments show that DA algorithm obtains superior community results compared with six other widely used algorithms, which indicate that DA algorithm has advantages for community detection.

© 2019 Elsevier Inc. All rights reserved.

1. Introduction

Many complex systems in different fields such as computer science, sociology, biology, transportation and medical science etc. are modeled as networks where nodes represent elements of systems and edges show different relations between nodes. Exploiting the network topological structures, such as the communities, can help us further understand network's characteristics. Communities in Facebook, for example, represent users sharing the same interests or topics. In biochemical networks, a community could be a unit of tissues with the same function. And a community in World Wide Web could be a collection of web sites relating to the same topic.

Usually, communities are treated as the subsets of nodes which are much more densely connected interiorly and sparsely linked exteriorly. Community detection can help us to find out the structurally or functionally similar modules. Therefore, it is important to design efficient and accurate algorithms to detect communities. A variety of community detection approaches based on global or local information have been proposed to reveal community structures.

Many community detection algorithms are designed from the whole network's perspective. Girvan–Newman(GN) algorithm [12] greatly promoted the development of community detection methods, detecting communities by gradually deleting

* Corresponding author.
E-mail address: yinghongma71@163.com (Y. Ma).

https://doi.org/10.1016/j.ins.2019.01.028
0020-0255/© 2019 Elsevier Inc. All rights reserved.

edges with high edge betweenness. The computing accuracy is improved greatly while the high computational demands are required as well. Thereafter, Newman [27] further put forward a fast algorithm to efficiently find communities in large scale networks by joining communities in pairs with the goal of optimizing modularity function. Clauset et al. [9] exploited some shortcuts in optimization following Newman's idea [27] and got communities with a fast greedy algorithm in much more sophisticated networks. Blondel at al. developed a modularity function to multi-level modularity optimization called Louvain algorithm [6], which surpassed modularity based algorithms with time complexity of square of network's size. Random walk based algorithms are also effective global methods. It is assumed that a random walker prefers to stay in the interior rather than the exterior of community. Walktrap [31] is one of the random walk based algorithms and performs well. Using eigenvalues and eigenvectors of adjacency matrix, spectral algorithm [28,32] is the outstanding example in algebraic method. Clustering approaches in information theory are also introduced to detect communities. For example, Infomap [35] decomposed a network into modules by optimally compressing a description of information flows on the network. Radicchi [33] decoded communities using coding-theoretical methods. Kernighan–Lin (KL) is also a classical algorithm using a heuristic procedure to detect communities [15], which is familiar to us because of the high efficiency as well as the wide application. KL algorithm and integer programming [22] were combined by Lin et al. to discover communities. Žalik and Žalik [40] introduced the entropy function as the optimization function and designed a memetic algorithm in the whole network.

Local information of networks is also exploited to detect communities. A weighted local view method based on observation over ground truth was presented by Hu et al. [13], uncovering a pattern that nodes with more similar degrees are more likely to connect. Bai presented a community description model [4] which was a fast community detection approach defined by a model of nodes to represent communities with their importance. Clauset [8] introduced an algorithm by optimizing the local modularity values. Abdelsadek and his collaborators [1] gave an approach depending on the density of triangles in communities. Raghavan et al. [34] proposed a label propagation algorithm (LPA) which was a nearly linear time method even though the results was not always steady. Žalik [39] also designed a bottom up community detection algorithm with low computational cost and high accuracy in which the communities are starting from adjacent nodes' pairs and their maximal similar neighbors. More local structure indexes, such as node's similarities [3,30] and node's adjacency lists [23] were used to detect communities.

It is found via summarizing previous methods on community detection that the algorithms based on global information guarantee the accuracy and the effectiveness. Meanwhile, those algorithms who employ nodes' neighbors information might have a local optimization even though they have low time complexity. It is an interesting problem to make balance between the global and the local information, the accuracy and the time complexity in designing algorithms.

In this paper, the local and global information are considered together and a divide and agglomerate (DA) algorithm for community detection is proposed in social networks. The basic idea of DA algorithm is combined by a two-stage strategy: The first stage is to divide a network into small groups according to node pairs' similarities, and the most similar nodes are in the same group; The second stage is to merge a group and the other when they have the highest attraction, and then go on this stage one by one until the community criterion is steady.

This paper is arranged in four sections, introduction, method, experiments and discussion. In the introduction section, the summary of some related previous works, significance and motivation are included. In Section 2, a divide and agglomerate (DA) algorithm is presented in detail. The novel similarity index (CAA) and the node's attraction (AT) are also defined. Two stages of DA algorithm and an example to illustrate it are also investigated. In Section 3, experimental results demonstrate the efficiency and accuracy of DA algorithm in comparison with six classical methods. Not only real world networks but synthetic networks are analyzed comprehensively. 11 real networks and hundreds of synthetic data are tested. The efficiency and accuracy of DA method are illustrated by tables and figures. In Section 4, comparisons of other methods on different similar indexes and time complexity are discussed. Future works and possible applications are presented in the final part of this paper.

2. The proposed method

In this section, the main idea of the divide and agglomerate algorithm based on the most similar neighboring node is presented. Our approach consists of two stages: the divide and agglomerate strategies. At first, each node chooses to be connected with its most similar node, thus reforming the most similar neighbor groups with all nodes in the original network but edges connecting each node with its most similar neighbor. After that, taking into consideration of the community criterion and the attraction based groups mergence principle, the final communities are detected. At last, the computational complexity is analyzed.

2.1. Similarity index

A network in this paper is denoted by an undirected and unweighted graph $G = (V, E)$ with a finite node set $V = \{v_1, v_2, \cdots, v_N\}$ which represents objects and an edge set E that models relationships between objects. $A = (A_{i,j})_{N\times N}$ is the adjacency matrix of G, where $A_{i,j} = 1$ means that there is a connection between nodes v_i and v_j. Given any two nodes in a network G, the node similarity index calculates to what extent the two nodes are like to each other. There are an abundance of similarity indexes with global or local information, such as Salton [36], Jaccard [14], HPI [5] and so on.

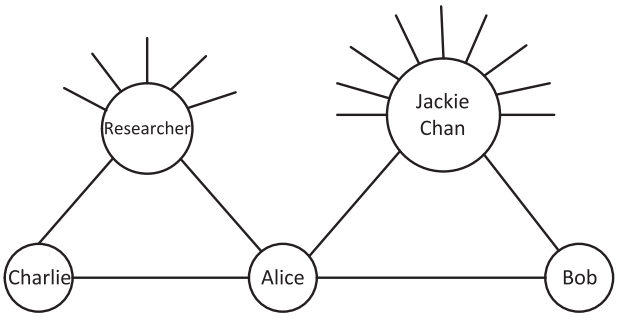


Fig. 1. An example of twitter following network.

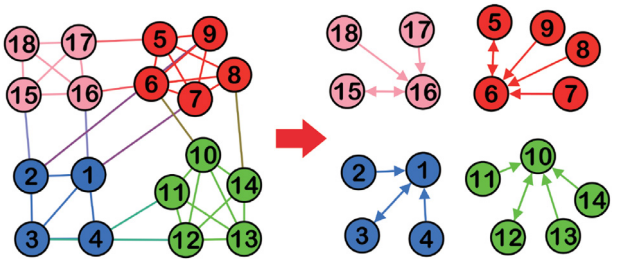


Fig. 2. An example of a network and its most similar neighbor groups. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

AA index [2] is a kind of similarity index proposed by Adamic and Adar. It also takes the common neighbors' degrees into consideration. It believes that the contribution of the small degree common neighbor node is greater than that of the big degree one. Then the index is defined as $Sim_{AA}(v_1, v_2) = \sum_{v \in CN_{v_1 v_2}} \frac{1}{\lg(k_v)}$, where $CN_{v_1 v_2}$ is the common neighbors' set for nodes v_1 and v_2. Each node pair has a similarity value, no matter they are connected or not.

To better understand the basic idea of AA index, an example in twitter following network is given in Fig. 1. Both Alice and Bob follow a famous Chinese kongfu star Jackie Chan who has lots of followers. At the same time, both Alice and Charlie follow an ordinary researcher who does some good work on network structure analysis. Then there will be more chances for Alice and Charlie to be in the same community. According to the AA index, the similarity value between Alice and Charlie is bigger than that of Alice and Bob.

In community detection, it is desirable that connected nodes can influence each other directly. If two nodes do not connect to each other, the information delivery between them needs at least two steps. Therefore, we constrain AA index in that only when two nodes connect directly can they have similarity. The constrained AA index, simply CAA, is defined in Eq. (1).

$$Sim_{CAA}(v_1, v_2) = \begin{cases} \sum_{v \in CN_{v_1 v_2}} \frac{1}{\lg(k_v)}, & \text{if } A_{v_1, v_2} = 1; \\ 0, & \text{Otherwise.} \end{cases} \tag{1}$$

2.2. Divide: Forming the most similar neighbor groups

For each node, it can choose to be in the same community as its most similar node, and CAA index is adopted here. Therefore, each node selects its most similar neighbor according to Eq. (2) and connects with it.

$$MSN(v_i) = \arg\max_{v_j} Sim_{CAA}(v_i, v_j), \tag{2}$$

where $MSN(v_i)$ represents the set of the most similar neighbor for v_i. If v_i has only one neighbor, then the only neighbor will be its most similar one. If a node has more than one most similar neighbors, then it randomly chooses one from them. After connecting with the most similar neighbor for each node, the most similar neighbor groups are formed with all nodes in the original network but with edges connecting nodes with their most similar neighbors. Therefore, we get preliminary partitions of a network. An example network illustrates how CAA works: Fig. 2 shows the network and the most similar neighbor groups of a network after calculating similarities for each pair of nodes using CAA index. The direction on edges displays the node pointing to its most similar node. For instance, node 2 pointing to node 1 means node 1 is the most similar node of node 2; Node 3 and node 1 pointing to each other means node 3 and node 1 are the most similar to each other. Therefore, the new forming groups are composed of four connected parts.

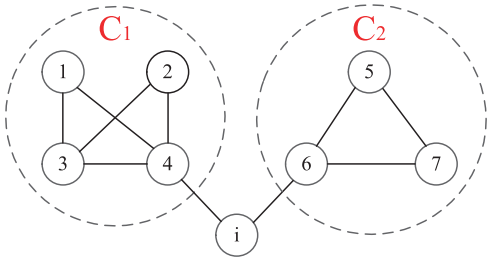


Fig. 3. An example of calculating attraction index *AT*.

2.3. Agglomerate: Merging groups by attraction

The divide step reveals community structures to some extent. Especially in some networks with obvious community structure, the most similar neighbor groups are exactly the ground truth communities of networks. Generally, to get more accurate community detection results, we can do further optimization. At first, we need to check whether those pre-partitioned most similar neighbor groups are communities or not according to the community criterion. For those who do not satisfy, the attraction based group mergence principle is proposed to optimize the results after divide strategy.

2.3.1. Community criterion

The widely accepted definition of community is a group of densely interconnected nodes that are sparsely connected with the other parts of network. Liu et al. [23] defined two different communities according to the ratio of internal connections to external's. By this definition, a strong community is that its internal connections are more than its external connections. A weak community is that its internal connections are equal or less than the external ones, while the number of internal links are larger than connections with any other communities. Therefore, the community criterion is that *both strong and weak communities are communities*. That is, a sub-graph satisfies the definition of the weak community or the strong community. Therefore, we treat it as a community in this paper.

An example of strong and weak communities is shown in Fig. 2. The sub-graphs in pink, red and green are all strong communities since their internal edges are more than external. As to the blue partition that includes nodes 1, 2, 3, and 4, although its internal number of edges is less than the external's, it is more than the edges connecting with any other communities. Hence, we say the blue one is a weak community. Therefore, all the most similar neighbor groups are treated as communities in Fig. 2.

2.3.2. Attraction index

In a network, each node plays a different role in forming the community to which it belongs. The more important the node is, the more possible it attracts nodes to join in its community. For example, a national academician can easily attract people to join in his research team, but it is not easy for an ordinary researcher to do so. Here, to measure a node v's attractiveness or importance in community C, a parameter $Attr_{v,C}$ is defined by the ratio of v's interior number of edges d_v^{in} in the community to the total community's interior number of edges d_C.

$$Attr_{v,C} = \frac{d_v^{in}}{d_C}. \tag{3}$$

An attraction index $AT_{u,C}$ measures to what extent a community C appealing to a node u. It is defined as

$$AT_{u,C} = \sum_{v \in C} Attr_{v,C} \times A_{u,v} (u \notin C). \tag{4}$$

$AT_{u,C}$ means only those nodes in community C that are connected with node u can bring about attractiveness. Similarly, the attraction index for the sub-graph C_i to attract C_j is also defined in the following,

$$AT_{C_i,C_j} = \sum_{u \in C_j} AT_{u,C_i}. \tag{5}$$

The attraction index AT is used to merge pre-partitioned groups into communities. A pre-partitioned group C_1 has the biggest attraction for another group C_2, then C_2 is combined to C_1. Since the attraction of a community is the ratio of the interior edges of the connected node to the total interior edges, therefore, small communities can also attract nodes. For example, the network in Fig. 3 consists of 8 nodes, in which there are two communities C_1 and C_2. As to node i, it can join the community with bigger AT. After calculating $AT_{i,C_1} = \frac{3}{5}$, $AT_{i,C_2} = \frac{2}{3}$, it will be more possible for node i to join in C_2.

Table 1
Nodes and their most similar neighbors (MSN) of karate network.

Node	1	2	3	4	5	6	7	8	9	10	11	12
MSN(i)	2	1	1	1	1	7	6	1	33	34	1	1
Node	13	14	15	16	17	18	19	20	21	22	23	24
MSN(i)	1	1	34	34	6	1	34	1	34	1	34	34
Node	25	26	27	28	29	30	31	32	33	34		
MSN(i)	32	32	34	34	34	34	34	34	34	33		

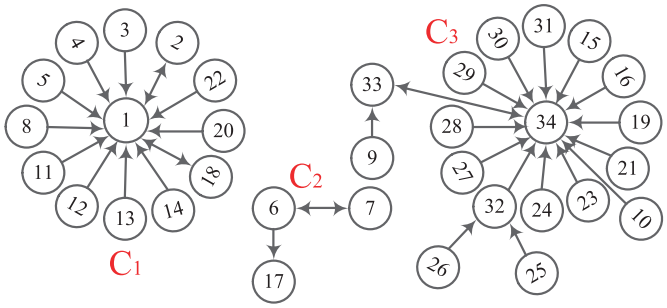


Fig. 4. The most similar neighbor groups for karate network by CAA.

2.4. A divide and agglomerate algorithm

Our algorithm is based on the idea that two nodes with larger similarity will have more chances to be in the same community. Therefore, similarities of each pairs of nodes are calculated at first. Then form the nearest neighbor groups (simplified by DAPre) by complying with the principle that a node and its most similar neighbors are in the same community. After that, we label the community if the group satisfies the criterion, while merge the groups to form communities by *AT* indexes. The DA algorithm is designed in detail as shown in table of Algorithm 1.

Algorithm 1 A divide and agglomerate algorithm.

Input: The adjacency matrix A of the network.
Output: Communities.
Step 1, divide: Construct the most similar neighbor groups DAPre.
Step 1.1: Calculate each node pairs' similarities according to equation 1.
Step 1.2: Find the most similar neighbor for each node according to equation 2.
Step 1.3: Form groups by the most similar neighbors of each node in the original network, and edges connecting each node with its most similar neighbors.
Step 2, agglomerate: Merge groups into communities.
Step 2.1: Check each group to see whether it satisfies the community criterion or not.
Step 2.2: If a group satisfy the community criterion, then output it as a community; Otherwise, if the group is a complete graph, merge it with the most attractive one; Else, randomly select a group and merge it with the one with the biggest attraction of it such that the modularity of the merged group does not decrease.
Step 2.3: Repeat Step 2.1 and 2.2 until there is no increment of network's modularity.

An example to show how the algorithm works for detecting communities with karate club network (Karate) is described step by step in the following. Zachary's karate club [38] network is a most frequently used empirical network in community detection, and it was first compiled by Zachary when he observed the relationship among karate members in a US university during the year 1970 to 1972. The network consists of 34 nodes and 78 edges, and each node represents a club member, while an edge between two nodes means that these two members often took part in the same activities other than the club activities. Due to the club course pricing problem, the club president John and the coach Mr. Hi had a conflict, separating the whole club into two parts.

According to the proposed algorithm, each node first calculates similarity with neighbors using CAA index and then find the most similar neighbor (MSN). Table 1 gives the nodes and their corresponding most similar neighbors. Following this table, each node connects with its most similar node and 3 most similar neighbor groups are formed as shown in Fig. 4. Based on the community structure criterion, only C_1 and C_3 satisfy it, leaving C_2 to find the most attractive part to join in.

In the agglomerate process, we need to calculate the attraction indexes of C_1 and C_3 to attract C_2 respectively. As to C_2, node6/s neighbors in C_1 are 1 and 11 in the original network, and for node 7, its neighbors are 1 and 5. Then $AT_{C_2,C_1} =$

Table 2
The statistics of real-world networks.

Networks	Nodes	Edges	Description
Karate	34	78	Zackary's Karate club [38]
Dolphins	62	159	Dolphin social network [24]
Football	115	613	US college football [12]
Polbooks	105	441	Books about US politics [16]
Email	1133	5451	Email network [7]
Netscientist	1589	2742	Scientists cooperation network [28]
Facebook	4039	88,234	Facebook dataset [25]
GR_QC	4158	13,428	Arxiv GR-QC collaborators [20]
GC_Hep_TH	8638	24,827	Arxiv HEP-TH collaborators [20]
GC_Hep_PH	11,204	117,649	Arxiv HEP-PH collaborators [20]
Co-authors	1033	2554	Co-author's subset in China

$\sum_{i\in C_2} AT_{i,C_1} = \frac{d_1^{in}}{d_{C_1}} + \frac{d_{11}^{in}}{d_{C_1}} + \frac{d_1^{in}}{d_{C_1}} + \frac{d_5^{in}}{d_{C_1}} = \frac{12}{52} + \frac{2}{52} + \frac{12}{52} + \frac{2}{52} \approx 0.54$. But for C_3, since C_2 has no neighbors in it, $AT_{C_2,C_3} = 0$. Therefore, C_2 is merged with C_1, and the karate network is divided into two communities.

2.5. Complexity analysis

DA algorithm is composed of two stages: the divide stage that forms the most similar neighbor groups as well as the agglomerate stage that merges them. In the first stage, it takes $O(|E|d_{\max})$ in the calculation of similarities, where $d_{\max}$ is the maximal number of a nodes neighbors and $|V|$ is the nodes number in network. It costs $O(d_{\max}|V|)$ in choosing one most similar neighbor and forming the most similar neighbor group. As to agglomerate stage, it costs $O(e(k-1)^2)$ where k is the number of pre-partitions, e is the maximal exterior edges of a partition. Therefore, the total time complexity of the proposed algorithm is $O(d_{\max}|V| + d_{\max}|E| + e(k-1)^2)$.

3. Experiments

To evaluate the performance of DA algorithm, both the real world and synthetic networks with ground truth are tested, and six other community detection algorithms, Infomap [35], LPA [34], Fastgreedy [9], Walktrap [31], Louvain [6] and LeadingEigen [28] are compared with DA method. All the experiments and simulations are conducted in RStudio using *'igraph'* package. All the experiments are taken on a PC with an Intel 2.4GHz i7-5500U CPU and 8GB RAM. The software platform is R in windows.

To evaluate the performance of algorithms, criterions for measuring the accuracy of community partitions are needed. One of the most used index is the modularity Q proposed by Newman and Girvan [9].

$$Q = \sum (e_{ii} - a_i^2), \tag{6}$$

where e_{ii} and a_i are the interior and the exterior number of edges of community C_i respectively. Another popular index NMI [10] (normalized mutual information) measures how close are the detected communities and the ground truth.

$$NMI(X|Y) = \frac{H(X) + H(Y) - H(X,Y)}{(H(X) + H(Y))/2}, \tag{7}$$

where X is the ground truth, Y is the predicted communities by algorithms. $H(X)$ and $H(X, Y)$ mean the entropy of community X and the joint entropy of X and Y respectively.

3.1. Real world networks

Eleven real world networks with diverse scales from tens to ten thousands are all analyzed by DA algorithm as well as six other community detection method. The network description and community detection results are shown in Tables 2, 3 and 5 respectively.

Dolphin network [24] (Dolphins) is formed by the frequency of the bottlenose dolphins played together. This network is divided into 4 communities by DA algorithm.

The American college football network [12] (Football) describes 115 college football teams' matches during the regular fall season of 2000, and these teams are grouped into 12 different conferences. The intra-conference's matches are much more often than inter-conferences'. Using DA algorithm, 12 conferences are detected which are almost the same as the reality. Although Infomap, LPA, Walktrap and Louvain achieve the same modularity value, DA algorithm outperforms others in NMI value compared to the ground truth.

The Political books (Polbooks) data was compiled by Krebs [16]. The nodes represent 105 books about US politics sold by on-line seller Amazon.com. The edges is the frequent co-purchasing of books by the same buyers. Books can be divided

Table 3
Comparisons of modularity values of DA algorithm on real-world networks with other six methods.

Networks	Algorithms															
	DA		DAPre		Infomap		LPA		Fastgreedy		Walktrap		Louvain		LeadingEigen	
	N_c	Q	N_c	Q	N_c	Q	N_c	Q	N_c	Q	N_c	Q	N_c	Q	N_c	Q
Karate	2	0.37	3	0.37	3	0.4	2	0.37	3	0.38	5	0.35	4	0.42	4	0.39
Dolphins	4	0.51	6	0.5	6	0.52	4	0.52	4	0.5	4	0.49	5	0.52	5	0.49
Football	12	0.6	33	0.25	12	0.6	11	0.6	6	0.55	10	0.6	10	0.6	8	0.49
Polbooks	4	0.52	6	0.49	6	0.52	4	0.52	4	0.5	4	0.51	4	0.52	4	0.47
Email	46	0.49	93	0.4	70	0.52	4	0.09	12	0.51	49	0.53	11	0.54	7	0.49
Netscientist	440	0.93	445	0.92	442	0.92	450	0.91	403	0.96	416	0.96	406	0.96	404	0.95
Facebook	9	0.73	9	0.73	92	0.81	60	0.82	13	0.78	77	0.81	17	0.83	42	0.75
GR_QC	371	0.77	461	0.69	374	0.77	363	0.78	74	0.8	448	0.76	43	0.85	18	0.8
GC_Hep_TH	701	0.69	996	0.59	678	0.67	510	0.6	109	0.7	853	0.64	56	0.75	5	0.1
GC_Hep_PH	520	0.62	632	0.59	760	0.61	366	0.43	155	0.57	630	0.54	47	0.65	8	0.54
Co-authors	11	0.83	11	0.83	73	0.76	41	0.79	15	0.82	12	0.83	12	0.81	14	0.82

N_c represents the number of communities.

according to the attitude into 3 categories, which are conservative, liberal and neutral. This network is divided into 4 communities by DA method. Except the conservative and liberal, we detect two more communities that are the books with inclination to conservative and liberal respectively.

The email network (Email) was composed by Alexandre Arenas [7]. It describes the email interchanges between members of the University Rovira i Virgili (Tarragona). The Louvain method performs the best, but LPA can hardly detect its community structures. There is not much difference for other methods.

The Netscientist network (Netscientist) is a co-authorship network of scientists working on network theory and experiment, as compiled by Newman [28]. After the pre-partition by DA method, a near good result for community detection is obtained. As to the final results, all algorithms' performance have little difference.

The facebook network dataset (Facebook) was collected and compiled by Mcauley and Leskovec [25] from survey participants using Facebook app, and it includes node features, friends lists and ego networks. Actually, this network combines 10 ego networks into one network using edges from all ego-nets. The combined network consists of 4039 nodes and 88,234 edges. From the community detection results in Table 3, the preliminary partition DAPre is 9 communities, and so for the final result. Although the modularity is not the biggest, the number of communities by DA is more close to the real number of ego networks. For Infomap, LPA and Walktrap algorithms, their modularities are all bigger than DA. However, their results contain at least 20 small communities which are less than 5 nodes. As to Louvain method, it achieves the best modularity score, but it also has four small communities with less than 50 nodes, which is not in accordance with reality.

The GR_QC, GC_Hep_TH and GC_Hep_PH collaboration networks are all compiled by Leskovec etc. [20]. They are from the e-print arXiv and covers scientific collaborations between authors papers submitted to General Relativity and Quantum Cosmology, High Energy Physics - Theory and Phenomenology category respectively. The community detection results show that the modularity optimization algorithms such as Louvain and Fastgreedy achieve good results with relative small number of communities. Other algorithms except LeadingEigen get communities numbers that are almost 9 times than that of Louvain.

The co-authors' network (Co-authors) is a data set collected by the authors in this paper. Co-authors is a subset of the cooperators for papers in management science and engineering of China from the year 2000 to 2016. 11 core committee members are chosen in the society of management science named Jingwen Li, Ziyou Gao, Qingguo Ma, Yijun Li, Haijun Huang, Yuejin Tan, Guoqing Chen, Ershi Qi,Yuan Li, Yanzhong Dang and Jiuping Xu. Their published papers indexed by SCIE, SSCI and CSSCI in Web of Science and CNKI are collected at 30 April 2017. It concludes 1033 authors and 2554 cooperations, shown in Fig. 5. Clearly, this cooperation network is ego-centered with small shortest paths which have small-world nature. Using DA algorithm, we can detect 11 communities centered on the 11 committee members just after preliminary partition. The results are shown in Table 3.

3.2. Synthetic networks

The synthetic networks are the LFR [17] benchmarks. Those networks have power-law distributions of both node degree and community size, which are features of real-world networks. Therefore, it is always considered as a substitution of real-world network with community structure and is appropriate to be used to evaluate the performance of community detection algorithms. The parameters of LFR networks, such as the number of nodes N, the average degree $< k >$, the power-law exponents for the degree α and the size of communities β, the mixing parameter μ, are set as follows. Where the mixing parameter μ is defined by the fraction of links connecting each node in a community to nodes in other communities to the total degree of nodes. Namely, the internal degree of node v_i in a community is $(1-\mu)k_{v_i}$ in average. The higher value of μ corresponds to the more ambiguous community structure. The parameters of LFR networks in following calculation are set as follows.

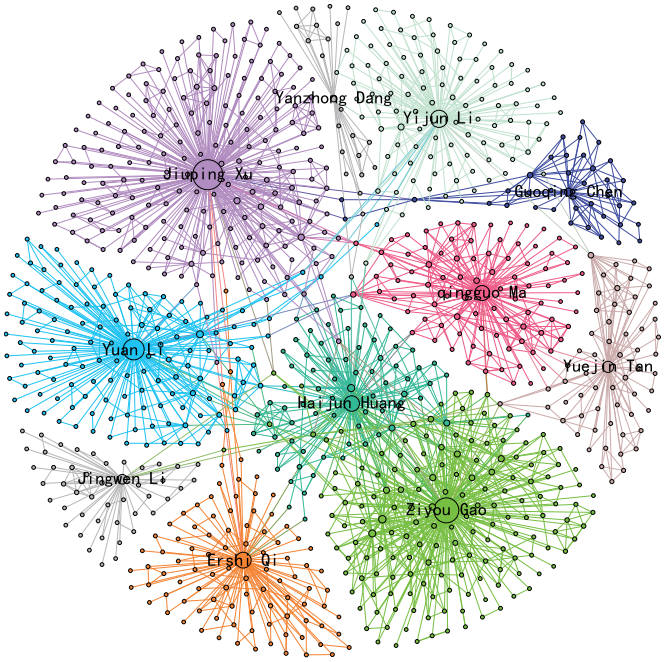


Fig. 5. Communities of management science co-authorship network in China.

- The number of nodes N: Set $N = 1000, 5000$ and $10,000$ respectively;
- The average degree $< k >= 15$ and the upper bound of degree $kmax = 0.1N$;
- The power-law exponents for the size of communities is set to $\beta = 1$ or 2 respectively;
- The power-law exponents for the degree of nodes is set to $\alpha = 2$;
- The maximum for the community size is set to $maxc = 0.1N$, and the minimum for the community size is set to $minc = 10$ for $N = 1000$, and $minc = 20$ for $N = 5000$ and $10,000$;
- The mixing parameter μ is set to $\mu = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7$.

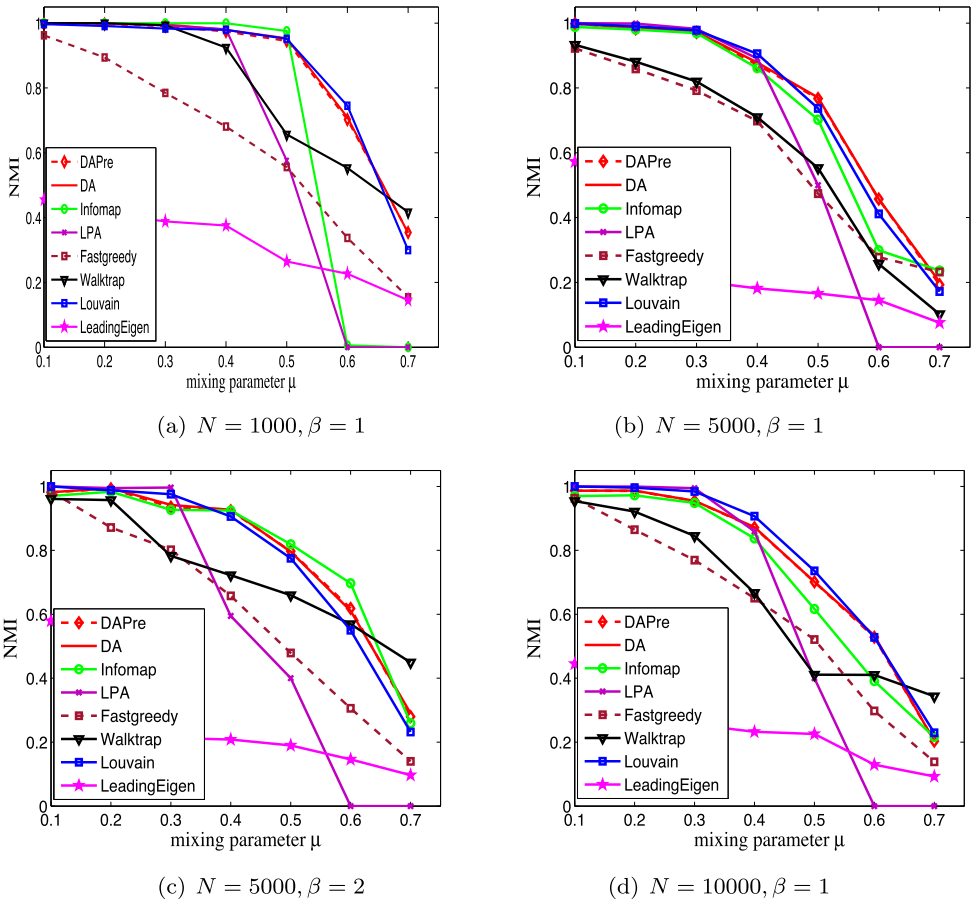
For each set of parameters, 50 networks are generated and the average community detection results are shown in Fig. 6. The uniform trends of the four panels in Fig. 6 show that NMI values decrease with the increasing of mixing parameter μ. That's because the greater μ means the more ambiguous network structure. Therefore, it gets more and more difficult to detect accurate communities as μ increases.

Panel 6(a) and 6(b) shows the results for LFR networks with the same communities distribution and different network' size respectively. The pre-partition results DAPre denoted by red dotted line with diamonds are almost the same as the final results that are represented by red solid line. This means the divide stage of DA obtains a relative good performance. For $\mu \leq 0.4$, except for the fastgreedy and LeadingEigen methods, all others' results have little difference, and almost reach to 1 for NMI. When μ equals to 0.5, LPA begins to have a sharp decrease and then fall to 0 since $\mu = 0.6$. Infomap's performance is steady when $\mu \leq 0.5$, but it rapidly declined to 0 when $\mu \geq 0.6$. Walktrap is not as good as Louvain or DA when $\mu \leq 0.6$, but it outperforms the others at $\mu = 0.7$ which means the community structures are ambiguous. While DA algorithm and Louvain are not bad at $\mu = 0.7$.

Panel 6(b) and 6(c) show the results of increasing community size distribution parameter β from 1 to 2. In panel 6(c), almost all the NMIs are just a little bit different than those in panel 6(b). This means the dissimilarities in community sizes do not affect the accuracy of the algorithm and the blurred modularity structure.

With the size increasing of network, the results for $N = 10,000$ are shown in panel 6(d). Louvain and DA algorithms still performs well. Infomap's NMI values are a little bit lower than the first two, but with no big difference. Walktrap performs relatively well but it outperforms others when $\mu = 0.7$. LPA is still not good for finding communities when structures are blur.

In order to further compare those algorithms for discovering communities in ambiguous networks, we fix $\mu = 0.6, 0.7$ and increase network sizes from 1000 to 10,000 with interval 1000 respectively, and results are shown in Fig. 7. Panel 7(a) shows the results for $\mu = 0.6$. DA and Louvain achieve better results than others in different network scales. Sometimes, Louvain outperforms DA, while when $N = 2000, 5000, 6000$ and 7000, DA method is better than Louvain. Walktrap's results are relatively steady, which is quite different from Infomap that increases from $N = 1000$ to 6000 and decreases from then on. Fastgreedy and LeadingEigen's results are not quite as good as we hope, and LPA do not detect any communities in



(a) $N = 1000, \beta = 1$ (b) $N = 5000, \beta = 1$

(c) $N = 5000, \beta = 2$ (d) $N = 10000, \beta = 1$

Fig. 6. Comparisons of the NMI values among our proposed algorithm and six other algorithms based on LFR network. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4
The statistics of LFR networks.

	nodes	edges	$< k >$	μ
LFR128	128	348	2.7	0.3
LFR256	256	705	2.7	0.4
LFR512	512	2503	4.9	0.3
LFR1000	1000	2635	2.6	0.2

this test. Panel 7(b) describes the comparison for $\mu = 0.7$. In this case, walktrap's results exceed all others. This is also in correspondence with results in Fig. 6. DA algorithm achieves a relative good performance when μ is bigger than 0.5.

3.3. Comparison of real data and synthetic networks

In order to compare with real world networks with ground truth, we generate four LFR networks with ground truth that are similar to real world networks. The four LFR networks, LFR128, LFR256, LFR512 and LFR1000 with different sizes are also used for testing. The statistical of the 4 networks are shown in Table 4.

Table 5 shows the NMI values of community detection results for networks with ground truth communities.

Results show that DA algorithm outperforms others except for the Polbooks network in which the attributes beyond the structure are included.

4. Discussion

Based on node similarity and attraction index, an efficient and effective DA algorithm is developed to detect communities with two stages' strategy: The first stage is to form preliminary partitions based on the similarity; And the second stage is

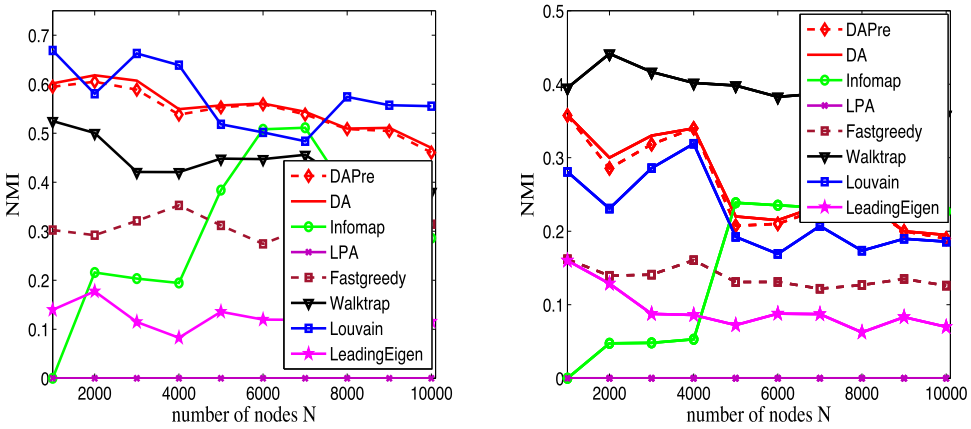


Fig. 7. NMI values of algorithms on benchmark networks with different scales ranging from 1000 to 10,000.

Table 5
Comparisons of NMI values of DA algorithm with other six methods on networks with ground truth.

Networks	Algorithms															
	DA		DAPre		Infomap		LPA		Fastgreedy		Walktrap		Louvain		LeadingEigen	
	N_c	NMI	N_c	NMI	N_c	NMI	N_c	NMI	N_c	NMI	N_c	NMI	N_c	NMI	N_c	NMI
Karate	2	1	3	0.86	3	0.70	2	1	3	0.7	5	0.5	4	0.59	4	0.68
Dolphins	4	0.51	6	0.58	6	0.54	6	0.57	4	0.56	4	0.58	4	0.55	5	0.53
Football	12	0.93	33	0.80	12	0.92	11	0.91	6	0.70	10	0.89	10	0.89	8	0.70
Polbooks	4	0.50	6	0.55	6	0.50	4	0.51	4	0.53	4	0.54	4	0.51	4	0.52
Co-authors	11	0.96	11	0.96	73	0.767	41	0.85	15	0.91	12	0.94	12	0.91	14	0.93
LFR128	5	0.88	5	0.88	8	0.82	6	0.85	5	0.78	6	0.76	5	0.84	5	0.67
LFR256	19	0.91	21	0.88	20	0.92	16	0.87	15	0.79	21	0.89	16	0.90	17	0.72
LFR512	28	0.94	31	0.93	31	0.89	26	0.88	20	0.81	28	0.89	20	0.87	24	0.67
LFR1000	85	0.66	87	0.65	143	0.6	89	0.44	67	0.5	117	0.51	68	0.51	78	0.43

N_c represents the number of communities.

Table 6
Some similarity indexes.

Salton [36]	$\frac{CN_{v_1v_2}}{\sqrt{k_{v_1}k_{v_2}}}$	Jaccard [14]	$\frac{CN_{v_1v_2}}{\Gamma_{v_1}\cup\Gamma_{v_2}}$
HPI [5]	$\frac{CN_{v_1v_2}}{min\{k_{v_1},k_{v_2}\}}$	HDI [5]	$\frac{CN_{v_1v_2}}{max\{k_{v_1},k_{v_2}\}}$
LHN [19]	$\frac{CN_{v_1v_2}}{k_{v_1}k_{v_2}}$	AA [2]	$\sum_{v\in CN_{v_1v_2}}\frac{1}{\lg(k_v)}$

to agglomerate a node with the other who is the most attractive neighbor and satisfying community criterion. The superior of DA method is that it do not need to know the whole structure of the network. The experiments on both real world and synthetic networks show that DA algorithm is rather efficient to discover community structure of networks. The future work is to apply DA algorithm on other fields, such as image and clustering technology.

4.1. Why choose CAA rather than other similarity indexes

The similarity index plays an important role in the dividing stage of DA method. How to select an effective and efficient index is crucial to the proposed method. At present, there exists a lot of similarity indexes using local information that can be seen from the following Table 6. As well as CAA, we restrict all these indexes to have values bigger than 0 only when two nodes are connected. To identify which one is much more suitable for our algorithm, we compare our method with other six similarity indexes together with common neighbor (CN) on LFR networks with $N = 1000$, and the result is shown in Fig. 8.

In Fig. 8, CN index outperforms others when $\mu \leq 0.5$. However, CAA and HPI indexes' NMI results are just a little bit lower than those of CN's. Salton, HDI and Jaccard are in the following, and AA index is the worst. All these indexes are effective in community detections, and the results are not quite different.

In order to identify which index is more appropriate for our method, we further compares the number of the most similar neighbor groups, and the results are shown in Table 7. It is easily found that the CAA and CN indexes' pre-partition numbers are much more close to the ground truth, while others are not. For CAA index, the average difference between the

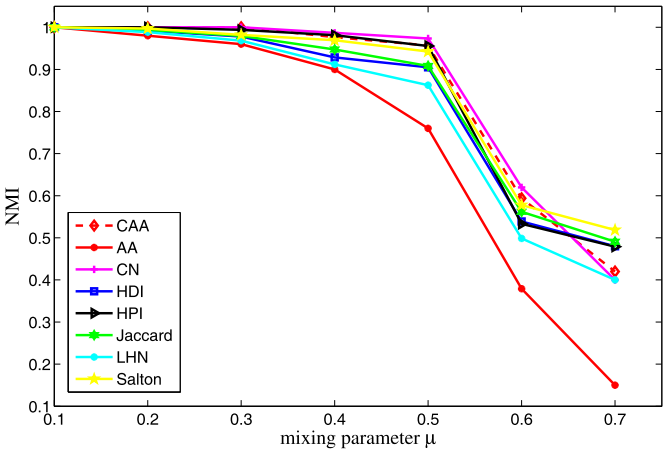


Fig. 8. Comparisons of different similarity indexes used by DA algorithm on LFR networks with $N = 1000$.

Table 7
Comparison of groups using different similarity measures for $N = 1000$.

μ	GT	CAA	AA	CN	HDI	HPI	Jaccard	LHN	Salton
0.1	25	28	51	25	134	92	119	154	103
0.2	27	30	41	29	165	99	137	186	118
0.3	25	28	46	38	165	110	142	200	129
0.4	22	24	32	23	168	92	144	209	125
0.5	31	39	55	22	152	129	130	199	111
0.6	22	24	22	42	163	103	141	204	120
0.7	26	20	21	30	135	98	125	180	111

GT denotes the number of ground truth communities.

number of nearest groups and the ground truth is 3.9 and the corresponding standard deviation is 2.1. But for CN index, its average gap is 8.1, and the standard deviation is 7.7. We also test these indexes on LFR networks with $N = 5000$, and the results also show that the CAA index is much more precise to the ground truth. Therefore, CAA index is adopted as the similarity index.

4.2. The complexity analysis

The divide and agglomerate algorithm consists of two strategies: the divide strategy that forms the most similar neighbor groups as well as the agglomerate part that merges them.

In the divide step, calculating similarity usually needs to find common neighbors for node pairs. For example, in order to make sure node v_i's neighbor v_l is also a neighbor of node v_j, we need to search in v_j's neighbor, which requires d_{v_j} comparisons. If we use hash table, then the complexity of searching v_l will reduce to $O(1)$. Therefore, for node v_i, it has d_{v_i} neighbors, and finding common neighbors for v_i and v_j needs d_{v_i} comparisons. If $d_{v_j} \leq d_{v_i}$, it's better to search from v_i's neighbors. Thus, finding common neighbors for v_i and v_j needs $O(min(d_{v_i}, d_{v_j}))$. In the overall network, the comparison number is $\sum_{(v_i,v_j)\in E} \min(d_{v_i}, d_{v_j}) < d_{\max} \times |E|$, and the similarity calculation's complexity is $O(d_{\max}|E|)$. What's more, it costs $O(d_{\max}|V|)$ in choosing one most similar neighbor and forming the most nearest neighbor network, where $|V|$ is the number of nodes in networks, and $|E|$ is the number of edges.

As to the agglomerate step, for each agglomeration, it roughly costs $O(e(k-1))$ where k is the number of most similar neighbor groups, e is the maximal exterior edges of a partition. The iteration number can reach the maximal at the extreme case which would be $k-1$. Therefore, the total complexity for this part is $O(e(k-1)^2)$.

In summary, the total time complexity of the DA method is $O(d_{\max}|E| + d_{\max}|V| + e(k-1)^2)$. In the extreme cases, k can be the number of the network nodes $|V|$. Therefore, in the worst case, the total time complexity of the proposed algorithm is $O(|V|^3)$. In fact, during DA algorithm work, when the divide stage finished, most nodes are classified into communities, a few nodes need to merge (Table 8).

Comparing with other six algorithms, LPA's complexity is the lowest, and it is nearly linear time complexity. Besides, Louvain, Infomap and Leading eigenvector algorithms have $O(|V|^2)$ complexity. Walktrap and DA algorithm's complexity are almost the same, while DA's divide part's complexity is almost the linear time complexity.

Table 8
Time complexity of our algorithm and other classical ones.

Algorithm	Time complexity
Fastgreedy	$O(\|E\| \cdot \|V\| \cdot \log \|V\|)$
Infomap	$O(\|V\| \cdot (\|E\| + \|V\|))$
LeadingEigen	$O(\|V\|^2)$
LPA	$O(\|E\| + \|V\|)$
Walktrap	$O(\|E\| \cdot \|V\|^2)$
Louvain	$O(\|V\|^2)$
DA	$O(d_{\max}\|V\| + d_{\max}\|E\| + e(k-1)^2)$

Table 9
The node's size of diving and AggloRatio.

Network	Nodes	Pre_nodes	AggloRatio
Karate	34	31	0.088
Dolphins	62	57	0.081
Polbooks	105	86	0.181
Co-authors	1033	1033	0
Email	1133	855	0.245
Netscientist	1589	1575	0.009
GR_QC	4158	3820	0.081
GC_Hep_TH	8638	7904	0.085
GC_Hep_PH	11,204	8307	0.259
LFR, $\mu = 0.1$	1000	993	0.007
LFR,$\mu = 0.2$	1000	988	0.002
LFR,$\mu = 0.3$	1000	982	0.018
LFR, $\mu = 0.4$	1000	973	0.027
LFR, $\mu = 0.5$	1000	942	0.058
LFR,$\mu = 0.6$	1000	928	0.072
LFR,$\mu = 0.7$	1000	485	0.542

Finally, we pay more attention to agglomerate stage. Let the *Pre_nodes* be the number of nodes that are already divided into communities, and *AggloRatio* be the ratio of nodes to be agglomerated,

$$AggloRatio = 1 - \frac{|Pre_nodes|}{\text{the number of total nodes}}.$$

The pre-partition results of *Pre_nodes* and *AggloRatio* are shown in Table 9.

In real data, the maximum *AggloRatio* of tested real-world networks is 0.259. And for LFR networks with size $N = 1000$, the maximum $AggloRatio = 0.542$ when $\mu = 0.7$. As we know the boundary of communities are blurring when $\mu \geq 0.5$. Hence, the ratio of nodes to be agglomerated could be small when there are obvious community structures. On this point, the proposed DA algorithm performs with very low time complexity.

4.3. Future works

Basically, the divide and agglomerate algorithm is a two-stage strategy to detect communities. The first stage is to divide the whole network into most similar groups by the most similarity of pairs of nodes, and the second stage is to merge two groups into one if they have the highest attraction and halt till no increment.

The idea of similarity of nodes might be used to image segmentation, which aims to partition a grid of pixels into contiguous regions corresponding to objects. In this sense, each segmented region can be viewed as a community, and it might get preliminary clusters of pixels. The optimized methods of the modularity function and application of Bayes nearest neighbor to settle image segmentation and classification problem [21,42,43] shed light on future application of DA algorithm. Using community membership as a useful information source for recommendation [11,18] enlightens another application on recommendation systems since communities can be looked as clusters if we ignore links. In addition, the personalized recommendations of local interesting venues [41] for tourists and user attributes learning [29] can be done based on communities.

The community structure also affects social contagions and epidemics through structural trapping. A virus spreads readily within a community and tends not to spread from one community to another [26]. Communities allow us to estimate how much the spreading pattern of a meme deviates from that of infectious diseases [37]. Therefore, applications of DA algorithm is not just a kind of partition approach, but rather a strategy in analysis of structure data which is useful in many domains.

Acknowledgments

We would like to thank the anonymous reviewers for the constructive comments and suggestions, which undoubtedly improved the presentation of this paper. We show our great appreciation to all the authors who collected and shared the data, such as Karate, Dolphins, Football, Polbooks, Email, Netscientist, Facebook, GR-QC, GR-Hep-TH, GR-Hep-PH and the LFR to be benchmark networks. Finally, we would like to thank the National Science Foundation of China (no. 71471106) that supports this research.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ins.2019.01.028.

References

[1] Y. Abdelsadek, K. Chelghoum, F. Herrmann, I. Kacem, B. Otjacques, Community extraction and visualization in social networks applied to twitter, Inf. Sci. 424 (2018) 204–223.
[2] L. Adamic, E. Adar, Friends and neighbors on the web, Soc. Netw. 25 (3) (2003) 211–230.
[3] S. Ahajjam, M. Haddad, H. Badir, A new scalable leader-community detection approach for community detection in social networks., Soc. Netw. 54 (2018) 41–49.
[4] L. Bai, X. Cheng, J. Liang, Y. Guo, Fast graph clustering with a new description model for community detection, Inf. Sci. 388 (2017) 37–47.
[5] A. Barabási, E. Ravasz, Z. Oltvai, Hierarchical organization of modularity in complex networks, LNP 625 (2003) 46–65.
[6] V. Blondel, J. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks., J. Stat. Mech-theory. E 2008 (10) (2008) 155–168.
[7] M. Bogũná, R. Pastor-Satorras, A. Díaz-Guilera, A. Arenas, Models of social networks based on social distance attachment, Phys. Rev. E 70 (2004) 056122.
[8] A. Clauset, Finding local community structure in networks., Phys. Rev. E 72 (2005) 026132.
[9] A. Clauset, M. Newman, C. Moore, Finding community structure in very large networks., Phys. Rev. E 70 (2004) 066111.
[10] L. Danon, A. Díaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification, J. Stat. Mech-theory. E 2005 (09) (2005) 09008.
[11] H. Feng, J. Tian, H. Wang, M. Li, Personalized recommendations based on time-weighted overlapping community detection, Inf. Manag. 52 (7) (2015) 789–800.
[12] M. Girvan, M. Newman, Community structure in social and biological networks., Proc. Natl. Acad. Sci. USA 99 (2002) 7821–7826.
[13] Y. Hu, B. Yang, H. Wong, A weighted local view method based on observation over ground truth for community detection, Inf. Sci. 355–356 (2016) 37–57.
[14] P. Jaccard, Etude comparative de la distribution florale dans une protion des alpes et des jura, Bull. Soc. Vaudoise Sci. Nat. 37 (1901) 547–579.
[15] B. Kernighan, S. Lin, An efficient heuristic procedure for structure graphs, Bell Syst. Tech. J. 49 (1970) 291–307.
[16] V. Krebs, Uspolbooks, http://www.orgnet.com.
[17] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Phys. Rev. E 78 (2) (2008) 046110.
[18] D. Lee, P. Brusilovsky, Improving personalized recommendations using community membership information, Inform. Process. Manag. 53 (5) (2017) 1201–1214.
[19] E. Leicht, P. Holme, M. Newman, Vertex similarity in networks, Phys. Rev. E 73 (2) (2006) 026120.
[20] J. Leskovec, J. Kleinberg, C. Faloutsos, Graph evolution:densification and shrinking diameters, ACM TKDD 1 (1) (2007) 2.
[21] S. Li, D. Wu, Modularity-based image segmentation, IEEE T. Circ. Syst. Vid. 25 (4) (2015) 570–581.
[22] C. Lin, J. Kang, J. Chen, An integer programming approach and visual analysis for detecting hierarchical community structures in social networks, Inf. Sci. 299 (C) (2015) 296–311.
[23] W. Liu, M. Pellegrini, X. Wang, Detecting communities based on network topology, Sci. Rep. 4 (4) (2014) 5739.
[24] D. Lusseau, K. Schneider, O. Boisseau, P. Haase, E. Slooten, S. Dawson, The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations, Behav. Ecol. Sociobiol. 54 (4) (2003) 396–405.
[25] J. Mcauley, J. Leskovec, Learning to discover social circles in ego networks, in: International Conference on Neural Information Processing Systems, 2012, pp. 539–547.
[26] S. Melnik, M. Porter, P. Mucha, J. Gleeson, Dynamics on modular networks with heterogeneous correlations, Chaos 24 (2) (2014) 1082–R.
[27] M. Newman, Fast algorithm for detecting community structure in networks., Phys. Rev. E 69 (2004) 066133.
[28] M. Newman, Finding community structure in networks using the eigenvectors of matrices., Phys. Rev. E 74 (2006) 036104.
[29] L. Nie, L. Zhang, M. Wang, R. Hong, A. Farseev, T. Chua, Learning user attributes via mobile social multimedia analytics, ACM Trans. Intell. Syst. Tech. 8 (3) (2017) 36.
[30] Y. Pan, D. Li, J. Liu, J. Liang, Detecting community structure in complex networks via node similarity, Phys. A 389 (14) (2012) 2849–2857.
[31] P. Pons, M. Latapy, Computing communities in large networks using random walks, in: International Conference on Computer and Information Sciences, 2005, pp. 284–293.
[32] A. Pothen, H. Simon, K. Liou, Partitioning sparse matrices with eigenvectors of graph., SIAM J. Matrix. Anal. A 11(3) (1990) 430–452.
[33] F. Radicchi, Decoding communities in networks, Phys. Rev. E 97 (2018) 022316.
[34] U. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks., Phys. Rev. E 76 (2) (2007) 036106.
[35] M. Rosvall, C. Bergstrom, Maps of information flow reveal community structure in complex networks, Proc. Natl. Acad. Sci. USA (2007) 1118–1123.
[36] G. Salton, M. McGill, Introduction to modern information retrieval, 1983.
[37] L. Weng, F. Menczer, Y. Ahn, Virality prediction and community structure in social networks, Sci. Rep. 3 (8) (2013) 2522.
[38] W. Zachary, An information flow model for conflict and fission in small groups, J. Anthropol. Res. 33 (4) (1977) 452–473.
[39] K. Žalik, Maximal neighbor similarity reveals real communities in networks, Sci. Rep. 5 (2015) 18374.
[40] K. Žalik, B. Žalik, Memetic algorithm using node entropy and partition entropy for community detection in networks, Inf. Sci. 445 (2018) 38–49.
[41] Y. Zhao, L. Nie, X. Wang, T. Chua, Personalized recommendations of locally interesting venues to tourists via cross-region community matching, ACM Trans. Intell. Syst. Tech. 5 (3) (2014) 1–26.
[42] L. Zhu, H. Jin, R. Zheng, X. Feng, Effective naive Bayes nearest neighbor based image classification on gpu, J. Supercomput. 68 (2) (2014) 820–848.
[43] L. Zhu, H. Jin, R. Zheng, X. Feng, Weighting scheme for image retrieval based on bag-of-visual-words, Image Process. Iet 8 (9) (2014) 509–518.