Local Community Detection via Edge Weighting

Weiji Zhao[1,2], Fengbin Zhang[1(✉)], and Jinglian Liu[2]

[1] School of Computer Science and Technology, Harbin University of Science and Technology, Harbin, People's Republic of China
sdzhaoweiji@163.com, zhangfb@hrbust.edu.cn
[2] School of Information Engineering, Suihua University, Suihua, People's Republic of China
{sdzhaoweiji,datamining}@163.com

Abstract. Local community detection aims at discovering a community from a seed node by maximizing a given goodness metric. This problem has attracted a lot of attention, and various goodness metrics have been proposed in recent years. However, most existing approaches are based on the assumption that either nodes or edges in network have equal weight. In fact, the usage of weights of both nodes and edges in network can somewhat enhance the algorithmic accuracy. In this paper, we propose a novel approach for local community detection via edge weighting. In detail, we first design a new node similarity measure with full consideration of adjacent nodes' weights. We next develop an edge weighting method based on this similarity measure. Then, we define a new goodness metric to quantify the quality of local community by integrating the edge weights. In our algorithm, we discover local community by giving priority to shell node which has maximal similarity with the current local community. We evaluate the proposed algorithm on both synthetic and real-world networks. The results of our experiment demonstrate that our algorithm is highly effective at local community detection compared to related algorithms.

Keywords: Local community detection · Community structure · Edge weighting · Node similarity

1 Introduction

Network is a data structure composed of a series of nodes interconnected by edges, and widely used to model many complex systems, such as social networks [6, 8, 20], collaboration networks [13], the Internet [4], and E-mail networks [21]. A common property of these networks is community structure. Community structure refers to the division of network nodes into groups within which the edges are dense but between which they are sparse [5, 6, 17, 18]. Community detection has many applications in the field of analyzing online social networks, collaborative tagging systems, biological networks [23].

Traditional community detection methods aim at discovering all communities in network based on the global network structure [3, 6, 14, 16, 19, 21]. For some huge

S. Ma et al. (Eds.): AIRS 2016, LNCS 9994, pp. 68–80, 2016.
DOI: 10.1007/978-3-319-48051-0_6

networks, such as social network and Web network, they are too huge to get the entire network structure nowadays [7]. The global based methods do not work on these huge networks. For solving this problem, local community detection was proposed.

Local community detection aims at discovering a community from a seed node requiring only the information of local network structure, and several algorithms have been proposed in recent years [1, 2, 7, 11, 12, 23]. These algorithms explore local community by maximizing a certain goodness metric. However, most existing goodness metrics are based on the assumption that either nodes or edges in network have equal weight. To ignore the weight of both nodes and edges in network is to throw out a lot of data that could help us to detect local community more accurately.

In this paper, we first design a new node similarity measure with full consideration of adjacent nodes' weights. We next develop an edge weighting method based on this similarity measure. Via edge weighting, every edge in network is assigned with a weight which represents the similarity between two nodes associated with this edge. Furthermore, we propose a new *Closeness-Isolation* metric to quantify the quality of a local community by integrating the edge weights. Finally, we propose our local community detection algorithm. We evaluate the proposed algorithm on both synthetic and real-world networks with ground-truth community structure. The results of our experiment demonstrate that our algorithm is highly effective at local community detection compared to related algorithms.

The rest of the paper is organized as follows. Section 2 introduces the problem definition of local community detection and reviews the existing methods. Section 3 introduces the edge weighting method and a novel local community quality metric *Closeness-Isolation*. We describe our algorithm in Sect. 4 and report experimental results in Sect. 5, followed by conclusions in Sect. 6.

2 Related Work

During the past decades, several local community detection algorithms have been proposed, such as [1, 2, 7, 11, 12, 23]. Most existing algorithms discover local community from a seed node by maximizing a goodness metric. How to design the goodness metric becomes a core problem in local community detection algorithms. In this section, we first introduce the problem definition of local community detection in network, and then review some representative goodness metrics.

2.1 Definition of Local Community in Network

In this subsection, we first give the definition of network, and then present the problem of local community detection in network.

Definition 1 (Network). Let $G = (V, E)$ be an undirected graph, V is the set of nodes and E is the set of edges in G. $n = |V|$ is the number of nodes in G. For two nodes, x, $y \in V$, $(x, y) \in E$ indicates there is an edge between nodes x and y. $m = |E|$ is the number of edges in G. The set of nodes adjacent to node x is denoted by $\Gamma(x)$, $\Gamma(x) = \{y \mid y \in V, (x, y) \in E\}$. The degree of node x is the number of nodes in $\Gamma(x)$, denoted by k_x.

The problem of local community detection can be presented as: For a network $G = (V, E)$, given a goodness metric for local community quality, local community detection starts from a seed node s ($s \in V$), the work is to discover the community D that s belongs to. As shown in Fig. 1, we can dynamically divide the entire network into three parts: local community D, D's shell node set N and the unknown node set U, $U = V - D - N$. Each node in N has at least one adjacent node in D. D has two subsets: the core node set C and the boundary node set B. The nodes in C are only connected by nodes in D, but any node in B has at least one neighbor node in N.

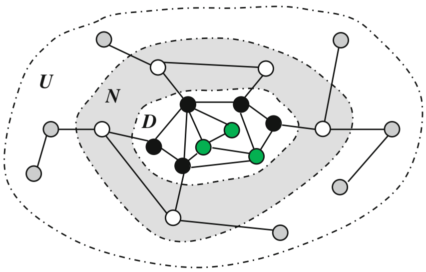


Fig. 1. An illustration of division of a network into local community D (Core Node Set C (green nodes) and Boundary Node Set B (black nodes)), D's Shell Node Set N (white nodes) and Unknown Node Set U (grey nodes) (Color figure online)

During the process of detecting local community, we have perfect knowledge of the connectivity of nodes in $D \cup N$, but have no knowledge of the connectivity of nodes in U. When local community detection algorithm starts, $D = \{s\}$, $N = \Gamma(s)$. In general, local community detection algorithm continuously starts from D and expand outward by absorbing external nodes from N into D until the given goodness metric stops improving [22]. Finally, D is the local community that node s belongs to. Similar definitions of local community detection can be found in [2, 10, 22].

2.2 Goodness Metrics that Assume All Edges are Equal

This kind of goodness metrics assume that all edges in network have equal weight. The representative goodness metrics are R and M.

Clauset [2] defined a local community quality metric called R by only considering the linkage information of boundary nodes in B.

$$R = \frac{B_{in}}{B_{in} + B_{out}} \tag{1}$$

where B_{in} is the number of inward edges that connect boundary nodes in B to other nodes in D, while B_{out} is the number of edges that connect boundary nodes in B to nodes in N. R measures the fraction of inward edges in all edges with one or more nodes in B.

Luo et al. [11] defined a local community quality metric called M, which focuses on the ratio of the number of internal edges and external edges.

$$M = \frac{E_{in}}{E_{out}} \tag{2}$$

where E_{in} is the number of edges with two nodes in local community D, while E_{out} is the number of edges with one node in D and the other in N. M measures the fraction of edges with both nodes in D in edges with one node in D and the other in N.

Both M and R assume that the edges in network have equal weight. In fact, the edge weights are different due to the similarities between each pairs of connected nodes are different. To ignore the edge weights is to throw out a lot of data that could help us to discover local community structure better.

2.3 Goodness Metrics that Assume All Nodes are Equal

This kind of goodness metrics focus on the internal similarity and external similarity of local community. For a local community D, the internal similarity of D is the sum of similarities between any two adjacent nodes both in D, while the external similarity of D is the sum of similarities between any two adjacent nodes with one node in D and the other in N. The representative metrics are *tightness* and *Compactness-Isolation*.

Huang et al. [7] adopted the node similarity measure, as shown in Formula (3), to evaluate the similarity between nodes x and y. Based on this measure, they introduced a metric for local community quality called *tightness*, and present a local community detection algorithm LTE via local optimization of the *tightness* measure.

$$s_{xy} = \frac{|\Gamma(x) \cap \Gamma(y)|}{\sqrt{|\Gamma(x)| \cdot |\Gamma(y)|}} \tag{3}$$

Ma et al. [12] introduced a d-neighbors based similarity measure called s^d_{xy} which takes into account non-adjacent nodes within a distance away. s^d_{xy} is defined in Formula (4). Based on this measure, they introduced a metric for local community quality called *Compactness-Isolation*, and proposed a local community detection algorithm called GMAC by maximizing *Compactness-Isolation*.

$$s^d_{xy} = \frac{\left|\Gamma(x)^d \cap \Gamma(y)^d\right|}{\left|\Gamma(x)^d \cup \Gamma(y)^d\right|} \tag{4}$$

$\Gamma(x)^d$ is a set of nodes whose shortest path length to node x is within d.

There are other node similarity measures, such as Common Neighbors and Jaccard Index [25]. The measure of Common Neighbors is directly counting the number of common neighbors two nodes have, while Jaccard Index is the ratio of the number of their common neighbors to the number of their union. All these methods assume that all adjacent nodes have equal weight. In fact, for any node, it has different similarities with its adjacent nodes.

3 Preliminaries

There are two subproblems in local community detection algorithm: how to design a proper goodness metric for local community quality and how to choose node in N as a member of D. The third problem hidden in these two subproblems is that how to evaluate the edge weights more accurately. In this section, we focus on these three subproblems, and give our solutions.

3.1 Edge Weighting

The weight of edge depends on the similarity between two nodes associated with this edge. In this subsection, we first give a new node similarity measure based on weighted neighbor nodes, and then introduce our edge weighting method.

Definition 2 (Node Similarity Based on Weighted Neighbor Nodes). Let $G = (V, E)$ be a network, for a node x, $o \in \Gamma(x)$, we define the weight of o as s_{xo}. s_{xo} can be calculated by methods in Subsect. 2.3. For a pair of nodes, x, $y \in V$, we define the similarity between x and y based on weighted neighbor nodes as ws_{xy}. ws_{xy} is defined as follows.

$$ws_{xy} = \frac{\sum\limits_{z \in \Gamma(x) \cap \Gamma(y)} (s_{xz} + s_{yz})}{\sum\limits_{u \in \Gamma(x)} s_{xu} + \sum\limits_{v \in \Gamma(y)} s_{yv}} \tag{5}$$

where the numerator is the sum of their common neighbors' weights, and the denominator is the sum of their neighbors' weights. The range of ws_{xy} is $[0, 1]$. When nodes x and y have no common neighbors, $ws_{xy} = 0$, and while they share the same neighbor nodes, $ws_{xy} = 1$.

Based on the above node similarity measure, we introduce our edge weighting method. For a pair of nodes, x and y, the similarity measure ws_{xy} considers the weights of their adjacent nodes, but neglects the fact that whether nodes x and y are directly connected or not. Two nodes tend to have higher similarity if they are directly connected. So our edge weighting method is given as follows.

Definition 3 (Edge Weighting). Let $G = (V, E)$ be a network, for any edge $(x, y) \in E$, let $w_{(x,\ y)}$ denote the weight of edge (x, y). $w_{(x,\ y)}$ is defined as follows.

$$w_{(x,y)} = ws_{xy} + \frac{k_x \times k_y}{2m} \tag{6}$$

For the weight of edge (x, y), on the basis of ws_{xy}, we plus the probability of these two nodes being connected to each other to $w_{(x,\ y)}$. Via edge weighting, we assign every edge in network with a weight.

3.2 Our Local Community Quality Metric *Closeness-Isolation*

Inspired by [7, 11, 12], we propose a new local community quality metric *Closeness-Isolation* (*CI* for short) based on the edge weights.

Definition 4 (*Closeness-Isolation* Metric). Let G = (V, E) be a network, the weight of edge (x, y) is $w_{(x,y)}$. For a local community D with shell node set N, the *Closeness-Isolation* Metric of D, denoted by $CI(D)$, is defined as

$$CI(D) = \frac{\sum\limits_{x,y \in D,(x,y) \in E} w_{(x,y)}}{1 + \sum\limits_{u \in D, v \in N,(u,v) \in E} w_{(u,v)}} \tag{7}$$

where the numerator is the sum of weights of all edges in D, and the denominator is one plus the sum of weights of all edges with one node in D and the other in N.

Instead of assuming the edges with equal weights, *CI* takes into account the edge weights, and is more reasonable than the other metrics. We use *CI* to measure local community quality in our algorithm.

3.3 Similarity Between Shell Node and Local Community

We define the similarity between shell node and local community in weighted network as follows.

Definition 5 (Similarity Between Shell Node and Local Community). Let G = (V, E) be a network, the weight of edge (x, y) is $w_{(x,y)}$. For a local community D with shell node set N, for a node $z \in N$, we denote the similarity between z and local community D by $sim(z, D)$. $sim(z, D)$ can be calculated as follows.

$$sim(z, D) = \sum_{v \in \Gamma(z) \cap D} w_{(z,v)} \tag{8}$$

$sim(z, D)$ is the sum of weights of all edges connecting z and nodes in D. Inspired by the fact that nodes in the same community are more likely to have higher similarities with each other, we choose the node in N which has highest similarity with local community D as candidate node.

4 Discover Local Community via Edge Weighting

With the edge weighting based local community quality metric *CI*, we propose our local community detection algorithm.

4.1 Our Local Community Detection Algorithm

Our local community detection algorithm starts from a given node s without any manual parameters. The pseudo code is described in Algorithm 1. Firstly, initialize $D = \{s\}$ and $N = \Gamma(s)$ (line (1)). In the while-loop (lines (2)–(16)), our algorithm keeps choosing the node $a \in N$ which has maximal similarity with local community D as candidate node (lines (3)–(10)), the similarity between node in N and local community D is calculated by Formula (8). If agglomerating the candidate node into D will cause an increase in *CI*, then add it to D and update N, otherwise, remove it from N (lines (11)–(15)), repeat until N is empty. Finally, return D as the local community of s (line (17)).

```
Algorithm 1: Local Community Detection
Input: a given node s, a network G =(V, E);
Output: local community D;
Describe:
  1) initialize D={s}, N=Γ(s);
  2) while N is not empty do
  3)   create a new dictionary variable dic_sim to store the similarities of
       nodes belonging to N with D;
  4)   for each i∈ N do
         //calculate similarity between node i and local community D
  5)     dic_sim[i] = 0;
  6)     for each j ∈Γ(i) ∩D do
  7)       dic_sim[i]+=w_(x,y); // see Formula (6)
  8)     end for
  9)   end for
 10)   find a such that dic_sim[a] is maximum;
 11)   if CI(D∪a)>CI(D) then
 12)     add a to D and update N;
 13)   else
 14)     remove a from N;
 15)   end if
 16) end while
 17) return D;
```

4.2 Time Complexity Analysis

In our algorithm, each node in network is denoted by a unique identifier. A network is stored by a hash table of nodes in the graph, and each node associates with a vector of its adjacent nodes. The values in vectors are sorted for faster access.

The running time of our algorithm depends on the size of the union of local community and its shell node set rather than that of the entire graph. Let t denote the size of $D \cup N$, E_{in} denote the number of edges with two nodes in D, E_{out} denote the number of edges with one node in D and the other in N, k denote the mean node degree of nodes in $D \cup N$. The computational cost of our algorithm mainly consists of two parts: calculating the weight of edges with one or more node in D and choosing a node in N as candidate node. For calculating the weight of edges, we need to compute t nodes' neighbor nodes, the weight of neighbor nodes of t nodes, and then compute $(E_{in} + E_{out})$ edges' weights. Their time complexity is $O(k \cdot t)$, $O(k^2 \cdot \log k \cdot t)$ and $O(k \cdot \log k \cdot (E_{in} + E_{out}))$ respectively. Adding these together, the time complexity is $O((k + k^2 \cdot \log k) \cdot t + k \cdot \log k \cdot (E_{in} + E_{out}))$. The most computational expensive steps is in lines (4)–(9), which is the time to find $a \in N$ having the maximal similarity with the current local community D. In each while-loop, the time complexity is $O(|N| \cdot |D| \cdot \log k)$.

5 Experiments

In this section, we evaluate the effectiveness of our algorithm on synthetic as well as real-world networks.

5.1 Related Methods and Evaluation Criteria

We compare our algorithm with three representative local community detection algorithms: (1) Clauset's algorithm [2] is a classic algorithm by maximizing metric R. Note that the same as [12, 22], we improve its stopping criteria by detecting changes in R. (2) Luo et al.'s algorithm [11] (LWP for short) is a famous algorithm to find the sub-graph with maximum metric M. (3) GMAC algorithm [12] is the most popular algorithm which uses d-neighbors to represent node. We fix $d = 3$ as suggested by authors. Our algorithm uses Jaccard Index to calculate neighbor nodes' weights.

We use three evaluation measures to compare algorithmic performance: *precision*, *recall* and *F-score*, which are widely adopted by other community detection methods [10, 12]. The *precision* and *recall* are calculated as follows.

$$Precision = \frac{|C_F \cap C_R|}{|C_F|} \tag{9}$$

$$recall = \frac{|C_F \cap C_R|}{|C_R|} \tag{10}$$

where C_R is the set of nodes in real local community which contains the given node and C_F is the set of nodes discovered by local community detection algorithm which starts from the given node.

F-score is the harmonic mean of *precision* and *recall*. Its formula is as follows.

$$F - score = 2 \times \frac{precision \times recall}{precision + recall} \tag{11}$$

5.2 Evaluation on Synthetic Networks

For comparing the performance of various local community detection algorithms, we first generate 10 synthetic networks with ground-truth community structure. There are 5000 nodes in every network.

LFR benchmark networks, introduced by Lancichinetti et al. [9], are widely used to test community detection methods [7, 12]. The important properties of this network generating model are defined as follows: the number of nodes is denoted by n, the average degree of nodes is denoted by k, the maximum degree is denoted by k_{max}, mixing parameter is denoted by μ, minus exponent for the degree sequence is denoted by *t1*, minus exponent for the community size distribution is denoted by *t2*, number of overlapping nodes is denoted by *on*, number of memberships of the overlapping nodes is denoted by *om*, minimum for the community sizes is denoted by *minc*, maximum for the community sizes is denoted by *maxc*. The parameters are set as follows: n = 5000, k = 10, k_{max} = 50, others except μ use default values. Mixing parameter u is the fraction of edges of each node outside its community, which is used to control the difficulty of community detection [19]. So we generate 10 networks with different mixing parameter μ ranging from 0.05 to 0.5 with a span of 0.05. These LFR benchmark networks are generated together with ground-truth community structure.

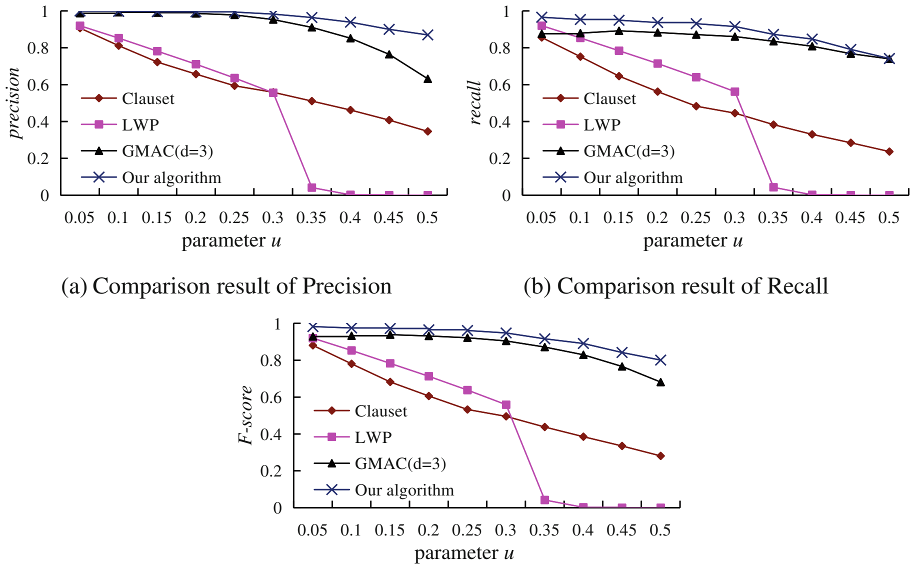
For every network in our experiments, we use each node in this network as a seed node once, and repeat the local community detection experiments for 5000 times which start from different node every time, then report algorithmic average *precision*, *recall* and *F-Score* on this network. Figure 2 shows the comparison results of *precision*, *recall*, *F-score* for four algorithms on these networks, respectively.

We discuss the experiments result in detail. Firstly, along with μ becomes larger, all the four algorithms suffer varying degrees of performance degradation and become ineffective to detect community structure. This is because the higher the mixing parameter u of a network, the weaker community structure it has.

Secondly, along with μ becomes larger, the performance of both LWP and Clauset drops rapidly, meanwhile GMAC and our algorithm drop slowly. This is because both LWP and Clauset simply depend on the number of edges incident to the node, neglect the fact that the weight of external edges are smaller than the internal edges.

Thirdly, our algorithm takes node weights in account, so it outperforms GMAC algorithm which neglects the node weights.

The *precision*, *recall*, and *F-score* of the LWP algorithm is zero or nearly zero when $\mu \geq 0.35$. This is because all the local communities discovered by LWP



(a) Comparison result of Precision

(b) Comparison result of Recall

(c) Comparison result of *F-score*

Fig. 2. Comparison results on LFR benchmark networks

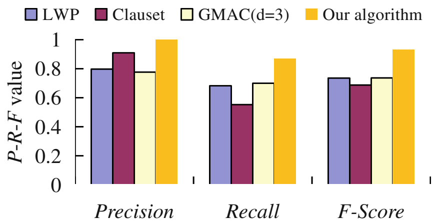
algorithm satisfy $M > 1$, which means the number of internal edges should be more than the number of external edges. However, almost no local community can satisfy $M > 1$ when $\mu \geq 0.35$, so LWP algorithm performs badly in this case. This conclusion is in accordance with the results reported in Ref. [7].

In general, because our algorithm takes into account the weight of nodes and edges, it performs best on LFR benchmark networks.

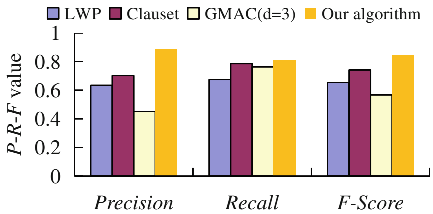
5.3 Evaluation on Real-World Networks

So far, we have presented the experimental results of the proposed algorithm on synthetic networks. In this subsection, we use additional three real-world networks to evaluate the performance of our algorithm. (1) The first network is Zachary Karate Club Network (Karate for short) [24], in which $n = 34$ and $m = 78$. It describes the friendships among 34 members of a karate club at a US university. (2) The second is NCAA football network (NCAA for short) [6], in which $n = 115$ and $m = 613$. It describes American football games between Division IA colleges during regular season Fall 2000. (3) The third is Books about US politics (Polbooks for short) [15], in which $n = 105$ and $m = 441$. It is a network of books about US politics published around the time of the 2004 presidential election and sold by Amazon.com. All of them are available at http://www-personal.umich.edu/~mejn/netdata/.

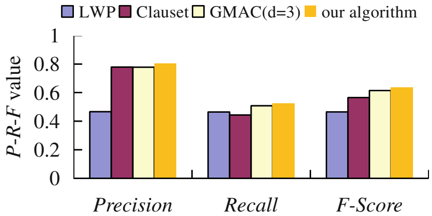
In our experiment, we use every node in these network as a seed node once, and repeat the local community detection experiments for n times which start from different node every time, where n is the number of nodes in this network, and report algorithmic



(a) Comparison Results on Karate



(b) Comparison Results on NCAA



(c) Comparison Results on Polbooks

Fig. 3. Comparison results on real-world networks

average *precision*, *recall* and *F-Score* on this network. The comparison result on real-world networks is reported in Fig. 3. Compared with the other three algorithms, our algorithm has highest *precision*, *recall*, and *F-score* at the same time on these real-world networks. Our algorithm makes use of the weight of both nodes and edges in network, and enhances the algorithmic accuracy. So it outperforms the other algorithms on real-world networks.

6 Conclusion and Future Work

Discovering local community is an important work in network analysis and many algorithms have been proposed to identify local community from a given node. Different from the existing local community detection methods that neglect the weight of both nodes and edges, we take into account the information to enhance the algorithmic accuracy. In this paper, we first propose an edge weighting method based on a new node similarity measure. Then, we introduce a framework for local community detection based on the edge weights. This framework opens a rich space for research, all algorithms can be embedded into this framework differing only in the similarity measures. Compared with other related algorithms, our algorithm doesn't need any manual parameters, and achieves good performance on both synthetic and real-world networks.

In future, we will apply our algorithm on real-world networks to discover local community and also study the community detection problem in heterogeneous networks.

Acknowledgments. The project is supported by National Natural Science Foundation of China (61172168).

References

1. Bagrow, J., Bolt, E.: A local method for detecting communities. Phys. Rev. E **72**(4), 046108-1–046108-10 (2005)
2. Clauset, A.: Finding local community structure in networks. Phys. Rev. E **72**(2), 026132 (2005)
3. Clauset, A., Newman, M.E., Moore, C.: Finding community structure in very large networks. Phys. Rev. E Stat. Nonlin. Soft Matter Phys. **70**(6), 264–277 (2004)
4. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationships of the internet topology. In: SIGCOMM 1999, pp. 251–262 (1999)
5. Fortunato, S.: Community detection in graphs. Phys. Rep. **486**(3/5), 75–174 (2010)
6. Girvan, M., Newman, M.: Community structure in social and biological networks. Proc. Natl. Acad. Sci. USA **99**(12), 7821–7826 (2002)
7. Huang, J., Sun, H., Liu, Y., Song, Q., Weninger, T.: Towards online multiresolution community detection in large-scale networks. PLoS ONE **6**(8), 492 (2011)
8. Jia, G., Cai, Z., Musolesi, M., Wang, Y., Tennant, D.A., Weber, R.J., Heath, J.K., He, S.: Community detection in social and biological networks using differential evolution. In: Hamadi, Y., Schoenauer, M. (eds.) LION 2012. LNCS, vol. 7219, pp. 71–85. Springer, Heidelberg (2012)
9. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. Phys. Rev. E **78**(4), 046110-1–046110-5 (2008)
10. Liu, Y., Ji, X., Liu, C., et al.: Detecting local community structures in networks based on boundary identification. In: Mathematical Problems in Engineering, pp. 1–8 (2014). http://dx.doi.org/10.1155/2014/682015
11. Luo, F., Wang, J., Promislow, E.: Exploring local community structures in large networks. Web Intell. Agent Syst. (WIAS) **6**(4), 387–400 (2008)
12. Ma, L., Huang, H., He, Q., Chiew, K., Wu, J., Che, Y.: GMAC: a seed-insensitive approach to local community detection. In: DaWaK, pp. 297–308 (2013)
13. Newman, M.: The structure of scientific collaboration networks. Work. Pap. **98**(2), 404–409 (2000)
14. Newman, M.: Fast algorithm for detecting community structure in networks. Phys. Rev. E Stat. Nonlin. Soft Matter Phys. **69**(6), 066133-1–066133-5 (2004)
15. Newman, M.: Modularity and community structure in networks. Proc. Natl. Acad. Sci. **103**(23), 8577–8582 (2006). http://www-personal.umich.edu/~mejn/netdata/
16. Newman, M., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E Stat. Nonlin. Soft Matter Phys. **69**(2), 026113-1–026113-15 (2004)
17. Radicchi, F., Castellano, C., Cecconi, F., et al.: Defining and identifying communities in networks. Proc. Natl. Acad. Sci. USA **101**(9), 2658–2663 (2004)
18. Schaeffer, S.: Graph clustering. Comput. Sci. Rev. (CSR) **1**(1), 27–64 (2007)
19. Shao, J., Han, Z., Yang, Q., Zhou, T.: Community detection based on distance dynamics. In: Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1075–1084 (2015)
20. Takaffoli, M.: Community evolution in dynamic social networks - challenges and problems. In: ICDM Workshops 2011, pp. 1211–1214 (2011)

21. Tyler, J.R., Wilkinson, D.M., Huberman, B.A.: Email as spectroscopy: automated discovery of community structure within organizations. Inf. Soc. **21**(2), 143–153 (2005)
22. Wu, Y., Huang, H., Hao, Z., Chen, F.: Local community detection using link similarity. J. Comput. Sci. Technol. (JCST) **27**(6), 1261–1268 (2012)
23. Wu, Y., Jin, R., Li, J., Zhang, X.: Robust local community detection: on free rider effect and its elimination. In: VLDB 2015, pp. 798–809 (2015)
24. Zachary, W.: An information flow model for conflict and fission in small groups. J. Anthropol. Res. **33**(4), 452–473 (1977)
25. Zhou, T., Lü, L., Zhang, Y.: Predicting missing links via local information. Eur. Phys. J. B **71**(4), 623–630 (2009)