

Research Article

Detecting Local Community Structures in Networks Based on Boundary Identification

Liu Yang,¹ Ji Xin-sheng,¹ Liu Caixia,¹ and Wang Ding²

¹ National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450002, China

² School of Electronics Engineering and Computer Science, Peking University, Beijing, China

Correspondence should be addressed to Liu Yang; liuyang198610@163.com

Received 4 July 2014; Revised 5 August 2014; Accepted 11 August 2014; Published 1 September 2014

Academic Editor: Bo Shen

Copyright © 2014 Liu Yang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Detecting communities within networks is of great importance to understand the structure and organizations of real-world systems. To this end, one of the major challenges is to find the local community from a given node with limited knowledge of the global network. Most of the existing methods largely depend on the starting node and require predefined parameters to control the agglomeration procedure, which may cause disturbing inference to the results of local community detection. In this work, we propose a parameter-free local community detecting algorithm, which uses two self-adaptive phases in detecting the local community, thus comprehensively considering the external and internal link similarity of neighborhood nodes in each clustering iteration. Based on boundary nodes identification, our self-adaptive method can effectively control the scale and scope of the local community. Experimental results show that our algorithm is efficient and well-behaved in both computer-generated and real-world networks, greatly improving the performance of local community detection in terms of stability and accuracy.

1. Introduction

Nowadays, most real world networks demonstrate that the nodes contained in their certain parts are densely connected to each other, which are usually called communities. Efficiently identifying those communities can help us to know the nature of those networks better and facilitate the analysis on those social networks. In many applications [1–3], the discovery of communities is an important issue in the analysis of complex networks, such as email networks, the World Wide Web, and citation networks. To better understand the topological properties and organizational structures of the complex systems [4, 5], many community detection algorithms have recently been developed by using techniques such as information theory [6], hierarchical clustering [7], and modularity optimization [8].

But when we have not got the entire information or structure of the networks due to their large scale and dynamics, these above global algorithms may not be applicable for detecting modules, especially in incomplete networks. Therefore, it is desirable to use limited available information of network graphs to discover the local community from

a certain source node. Recently, numbers of community identification algorithms [9–14] were proposed and widely applied to discover the local structure under the partial knowledge of the networks. However, most of these methods need to preset some parameters like local similarity threshold for node clustering and provide no automated way to find them. The stopping criteria of the local community detection greatly depend on the predefined parameters which control how strong the community should be. More seriously, the performances of these algorithms are greatly limited by the source node's position seriously, especially when the source node belongs to an obscure community that may introduce a large number of outliers.

In this paper, we suggest a new parameter-free method to discover the local community by using the incomplete information of known structure in networks. In our method, the local community C is regarded as a supernode. From the view of every potential clustering node having links with the local community, we calculate its link similarity with every neighborhood node, respectively, instead of just considering its link relationship with local community as most existing algorithms do and then cluster the node having the maximum

link similarity with the local community. Through the two self-adaptive phases of community detection, the method iteratively identifies the surrounding boundary of the current local community and updates the neighborhood of the local community for the next round of node clustering. Consequently, this mechanism guarantees that our method can add suitable nodes into the local community in the condition of no parameters predefined and finally obtain an accuracy and meaningful community from any given node. The main contributions are summarized as follows.

(1) In the agglomeration of the local community, for each neighborhood node, we comprehensively consider its link similarity with the local community and with other adjacent nodes, rather than just depending on its link relationship with the local community like most existing methods. This mechanism ensures that each expansion of the local community is reasonable and achieves high accuracy.

(2) By combining the advantages of boundary detection and link-similarity-based modularity optimization, we can automatically obtain the precise coverage of the local community in networks and make the quality of the discovering community optimal. So our algorithm needs not to preset some certain thresholds to stop the clustering and overcomes the resolution limitation that other modularity-based algorithms suffer from.

(3) Our algorithm has good robustness and stability. The community results do not depend on the position and topological features of the given nodes. Experimental results show that our algorithm is effective and efficient in both computer-generated networks and real-world networks, regardless of what kind of source nodes is given.

This paper is organized as follows. We review the existing local community detection methods and analyze the problems of these approaches in Section 2. In Section 3, we formalize the notions used in our method. Then the algorithm is described in detail in Section 4. Section 5 presents our experiment results and Section 6 concludes the paper.

2. Related Work

Here we have reviewed and compared several typical approaches to study the local community structure in networks. From the given node, these algorithms [9–11] proposed their different corresponding metrics to cluster the adjacent nodes. These metrics determine which node can be clustered into the community and finally form the local community structure. Clauset [9] defined local community modularity R :

$$R = \frac{B_{in}}{B_{in} + B_{out}}, \quad (1)$$

where B_{in} is the number of edges that connect community boundary nodes and nodes in the community, while B_{out} is the number of edges that connect boundary nodes and nodes outside the community. The algorithm iteratively clusters the neighboring node yielding the largest increase of R , until the community has reached a predefined size.

Luo et al. proposed a new modularity M for local community evaluation [10]. M is the ratio of internal edges E_{in} and external edges E_{out} ,

$$M = \frac{E_{in}}{E_{out}}, \quad (2)$$

where modularity M indicates the ratio of internal edges E_{in} and external edges E_{out} . Nodes will be added or removed from C if and only if it can cause an increase in M . This algorithm turns out to result in high recall but low accuracy.

In paper [11], a fitness function is defined,

$$f_C = \frac{k_{in}^C}{(k_{in}^C + k_{ex}^C)^\alpha}, \quad (3)$$

where k_{in}^C and k_{ex}^C are the total internal and external degrees of the nodes of module C and α is the resolution parameter, which is used for controlling the size of the communities.

Recently, several novel metrics for local community structure have also been presented in [12–15]. Combined with the advantages of these quality metrics, the existing algorithms have made great improvement in finding a more precise and complete community. Furthermore, some other methods have been proposed by using techniques such as maximum cliques extension [16] and clustering with selection probability [17] or are based on density drop of subgraphs [18]. But all of these methods have not really solved the problem of when to stop the clustering iteration, so they have to preset some manual parameters to control the expansion of the local community. These parameters are hard to be accurately acquired and may greatly affect the final results of community detection. More seriously, the performances of the approaches are sensitive to the given node and may be greatly confined by its location or topology. When starting from different nodes, even when they actually belong to the same community, the detecting results of local communities may be very different. To deal with the problems above, we introduce link similarity to measure the connecting tightness between the indeterminate neighborhood node and the local community, reveal the natural community structure based on boundary identification, and finally control the community size and scale via local optimization of the similarity-modularity measure.

3. Preliminary

Considering a network $G(V, E)$ composed of the node set V and the edge set E , $\omega(i, j)$ represents the weight of the edge connecting nodes i and j . Given a node s , our work is to discover the entire local community that s belongs to. In this section, we formalize some notions of the local community and introduce a structural similarity-based metric.

Definition 1 (neighborhood). The structure neighborhood [12, 19] of node i is the set $N(i) = \{v \in V \mid \exists e_{iv} \in E\}$, which contains node i and its adjacent nodes which have links with i .

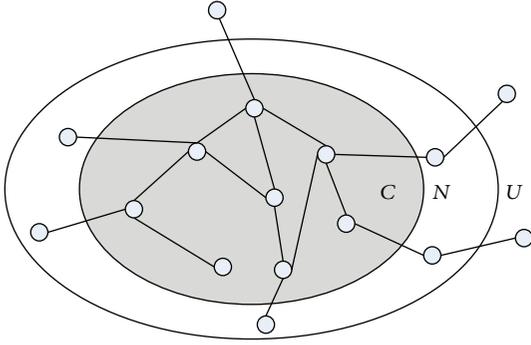


FIGURE 1: An illustration of an abstract network containing the local community C , its neighborhood $N(C)$, and the outside node set U .

By definition, $N(i)$ is comprised of i and the adjacent nodes which directly connect with node i . For the local community C (shown in Figure 1), $N(C) = \{v \in V - C \mid \exists i \in C, \text{ s.t. } e_{iv} \in E\}$ denotes the neighborhood of community C , $C \subseteq N(C)$, and $U = V - N(C)$ is the outside node set in networks. In order to quantify the connecting relationship between nodes, here we adopt a structural similarity measure which is extended from the cosine similarity [20]. That metric effectively denotes the link similarity of any two adjacent nodes in the network and reflects the local structural similarity between them.

Definition 2 (link similarity). Given a weighted network $G(V, E)$, one can get the neighborhood structure of the node i and node j to evaluate their similarity of link density and link formation [14]. The link similarity between two nodes i and j is defined as

$$L(i, j) = \frac{\sum_{k \in N(i) \cap N(j)} \omega(i, k) \cdot \omega(j, k)}{\sqrt{\sum_{k \in N(i)} \omega^2(i, k)} \cdot \sqrt{\sum_{k \in N(j)} \omega^2(j, k)}}. \quad (4)$$

When considering an unweighted graph, the weight $\omega(i, j)$ of an edge $e_{ij} \in E$ can be set to 1. It is easy to understand that the adjacent nodes sharing more common links have larger local similarity and connection tightness and are more likely to be clustered into the local community.

Generally, a network community is regarded as a group of nodes that are more densely connected inside the group than the outside of the network. The topology and organization of a community indicate its structural and functional characteristics that distinguish it from other communities in networks. Based on the above analysis, we can conclude that link similarity can be utilized as a criterion to determine whether the candidate node should be added in the community or not. Moreover, our experimental results also proved the effectiveness of that metric.

Definition 3 (boundary). By employing the structural similarity, one introduces the node set $B(C)$ to denote the boundary of local community C [4, 9]. For a node j , it is a neighborhood node for C ; that is, $j \in N(C)$. The criterion for the boundary $B(C)$ accepting j is $\exists L(j, C) < L(j, k)$, $k \notin C$.

The definition means the nodes inside $B(C)$ have less structural similarity with C than with nodes or subgraph outside C . As shown in Figure 1, the community structure usually characterizes that there are more interactions among its members than the remainder of the network; the boundary of the community is naturally formed by the border nodes which are far from the core part of their located community. Through detecting of the boundary nodes, we can determine the borders of the local community and make the local community detection under effective control avoid introducing the outliers during the process of the expansion by clustering its adjacent nodes.

4. The Two-Phase Local Community Detection Based on Boundary Identification

In this section, we describe our local community detection algorithm consisting of two phases: the agglomeration phase and the boundary identification phase. After identifying an accurate agglomeration, it is necessary for the local method to determine when to stop clustering nodes appropriately [19]. So our algorithm is developed by integrating the two phases to enhance the neighborhood node clustering in each iteration. Moreover, a link-similarity-based modularity is adopted to evaluate the quality of the local community and help to identify the border nodes.

In the agglomeration phase, we consider the local community C as a supernode. For each neighborhood node of C , we, respectively, calculate its link similarity with every adjacent node. Assuming i is a neighborhood node, find the node having the maximum link similarity with i . If that node just represents the local community C , add i into C ; if not, put i into the boundary set. Through that agglomeration process, we build the current local community and construct its boundary. The example procedure is illustrated in Figure 2. In the boundary identification phase, we confirm the boundary nodes of the local community to control its exact size and scale. For a node $j \in B(C)$ that has the maximum link similarity with a node p outside C , we merge the two nodes as a module and calculate the modularity gain $\Delta Q_S(C)$, which results from adding the module to the local community, to judge whether to add that module into C . After all the border nodes are fixed and no neighborhood nodes can be agglomerated, we get the final resulting local community. The algorithm is as shown in Algorithm 1.

It is important to note that, in the boundary identification phase, a new metric is necessary for our algorithm to measure the goodness of the discovered local community. Many quality functions have been proposed, such as fitness [11] and modularity [21]. Here we use the link-similarity-based modularity function Q_S proposed in [22]. It is extended from the normal modularity Q [21] and has a better ability to deal with hubs and outliers. Given two adjacent modules C and M , the modularity gain $\Delta Q_S(C)$ can be defined as follows:

$$\begin{aligned} \Delta Q_S(C) &= Q_S^{CUM} - Q_S^C - Q_S^M \\ &= \frac{2IS(C, M)}{TS} - \frac{2DS(C) \cdot DS(M)}{TS^2}, \end{aligned} \quad (5)$$

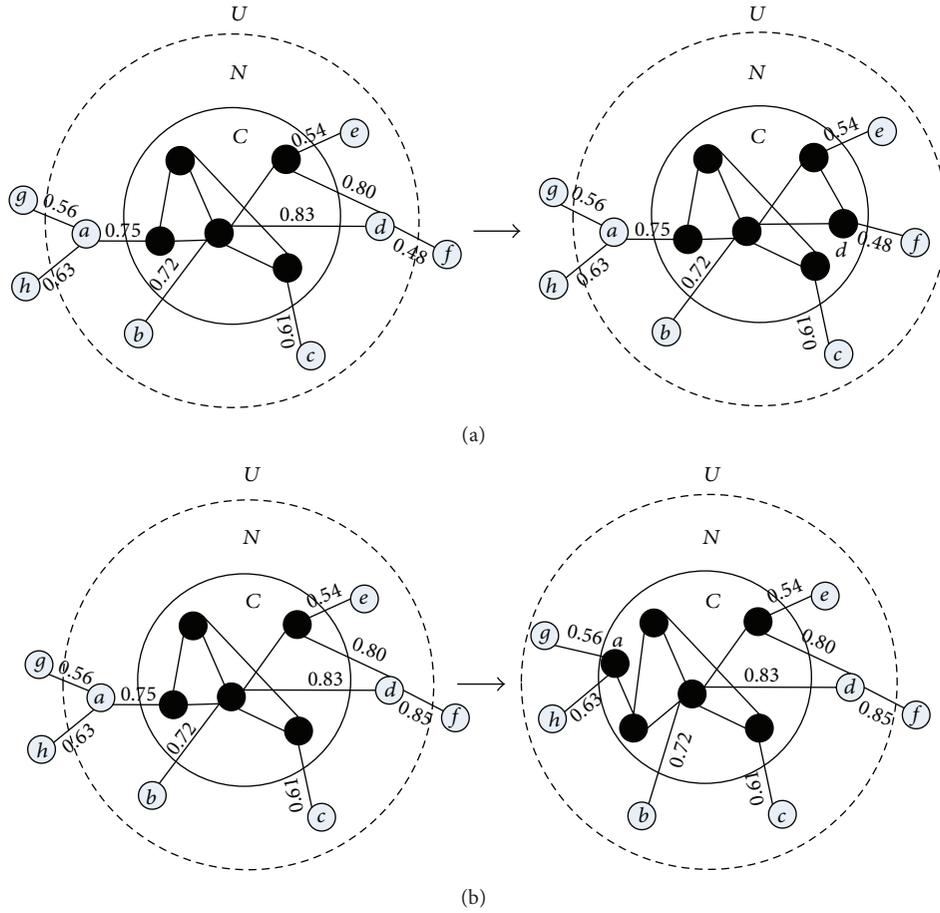


FIGURE 2: Procedure of the local community detection in an example network based on boundary identification. (a) An example for clustering the neighborhood node into the local community C , when node $d = \arg \max_{i \in N(C)} L(i, C) \wedge L(d, C) = \max_{j \in N(d)} L(d, j)$; (b) an example for clustering node a with the secondary maximum link similarity into the local community C , when node $d = \arg \max_{i \in N(C)} L(i, C) \wedge L(d, C) < \max_{j \in N(d)} L(d, j)$.

where $IS(C, M) = \sum_{u \in C, v \in M} L(u, v)$ is the total similarity of the links between two modules C and M , $DS(C) = \sum_{u, v \in C} L(u, v)$ is the total similarity of nodes within community C , and TS represents the total similarity between any two nodes in the network G .

The link-similarity-based modularity described above is an effective metric to evaluate the quality of networks partition, and we use the modularity gain $\Delta Q_S(C)$ to determine whether the boundary node belongs to the local community. If $\Delta Q_S(C) \leq 0$, that means the node j is the border node of C ; otherwise, the module M should be merged into the local community and update $N(C)$ and $B(C)$ for the next new round of agglomeration iteration, until there is no node left that satisfy the above conditions. It is worth noticing that the main task of our method is finding a higher modularity solution under the principle of the boundary identification and optimal similarity-modularity, rather than to search a partition greedily maximizing the modularity Q_S .

These boundary nodes around C are at the junction of several adjacent communities, and their special locations make it more difficult to judge whether it is contained in the

local community. Therefore, during the expansion of the local community, once the link conditions of C are changed, we should update its neighborhood and boundary and reevaluate the link similarity between them for the further agglomeration and boundary identification. That heuristic mechanism can ensure that our result is a strong and stable community by performing that two-phase community detection.

The running time of our algorithm is mainly consumed in calculating the link similarity and finding the neighborhood node with the largest similarity in the process of forming the local community. The calculation of $\Delta Q_S(C)$ is a little time consuming and the complexity is $O(|C|)$. So the computational complexity for the local community detection method is $O(k \log |C|)$, where k is the average degree of nodes inferred and $|C|$ is the number of nodes in the local community.

5. Experiments and Results

To evaluate the performances of the local community detection methods, we evaluate our results by precision, recall, and F -measure, which are widely adopted by other community

Local Community Detection Algorithm based on Boundary Identification

Input: Network $G(V, E)$, and a given node s

Output: The local community C that s belongs to.

```

(1) Initialize  $C = s, B = \emptyset$ ;
(2) Calculate  $N(C), U$ ;
(3) while true do
(4) //Phase 1: agglomerate local community
(5) for each  $i \in N(C)$  do
(6)   Calculate  $L(i, C)$  by (4);
(7)   if  $L(i, C) > \max_{k \in N(i)} L(i, k)$ 
(8)      $C \leftarrow C \cup \{i\}$ ; //merge  $i$  into  $C$ 
(9)     Update  $N(C), B$ ;
(10)  else
(11)     $B \leftarrow B \cup \{i\}$ ; //set  $i$  to be a boundary node
(12)  end if
(13) end for
(14) //Phase 2: Identify the boundary of the local community
(15) for  $i \in B$  do
(16)    $p \leftarrow \arg \max_{k \in N(i)} L(i, k)$ ; //  $p \notin C \wedge p \in N(i)$ 
(17)    $M \leftarrow \{p\} \cap \{i\}$ ; //merge  $i, p$  into a module node
(18)    $\Delta Q_s(C) \leftarrow Q_s^{C \cup M} - Q_s^C - Q_s^M$ ;
(19)   if  $\Delta Q_s(C) > 0$ 
(20)      $C \leftarrow C \cup M$ ; //merge  $M$  into  $C$ 
(21)     update  $N(C), B$ ;
(22)   end if
(23) end for
(24) end while
(25) return  $C$ 

```

ALGORITHM 1

detection methods [10, 11]. The precision and recall are calculated as follows:

$$\text{precision} = \frac{|F_d \cap F_r|}{|F_d|}, \quad (6)$$

$$\text{recall} = \frac{|F_d \cap F_r|}{|F_r|},$$

where F_r indicates the node set forming the real local community originating from a given node and F_d represent the node set which is the result of the local community detection. F -measure combines precision and recall:

$$F = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \quad (7)$$

In this section, the experiment uses a series of computer-generated benchmark networks and some real-world networks to validate the algorithms' performance. For all the networks, we compare the experimental results with other three typical algorithms (Clauset [9], Luo et al. [10], Chen et al. [12]), respectively.

(1) Synthetic benchmark networks: by using the Lancichinetti-Fortunato-Radicchi (LFR) benchmark networks [23], we can generate 9 networks with different mixing parameter μ ranging from 0.1 to 0.9 with a span of 0.1 and randomly choose 100 nodes from each network to evaluate the performance of our algorithm. These networks' important properties are presented as follows: the node number is $n = 10000$, the average edge number is $m = 151734$, and the average node

degree is $k = 15$. The community structures of these LFR benchmark networks are all known.

The mixture parameter μ denotes the average ratio between external and total degree of each node in the network. Generally, the lower the mixture parameter of a network is, the stronger community structure it has. The LFR networks with various mixture parameters are necessary and valuable to evaluate those methods. Figure 3, respectively, shows the comparison results of three metrics for different algorithms, because a well-performed algorithm should achieve high precision, recall, and F -measure at the same time. From the comprehensive comparison of all the metrics, it can be concluded that our algorithm achieves the best performance.

Particularly, when the mixture parameter μ is less than 0.5, the average F -measure of our algorithm is almost 0.90. This means our approach can discover exactly the full local community structure from the given node. But when the community structure of the LFR networks becomes weaker ($\mu \geq 0.5$), all the algorithms suffer varying degrees of performance degradation and become ineffective to detect community structure. In general, as the community structure of LFR networks tends to be weaker and weaker, our algorithm shows much better robustness than the other three algorithms.

(2) Real-world networks: real networks are always more irregular and various than synthetic networks and have more complex community structures. Here we choose five typical real-world networks: Football [24], Polblogs [25], US airport [26], US power grid [27], and Internet [28]. These networks'

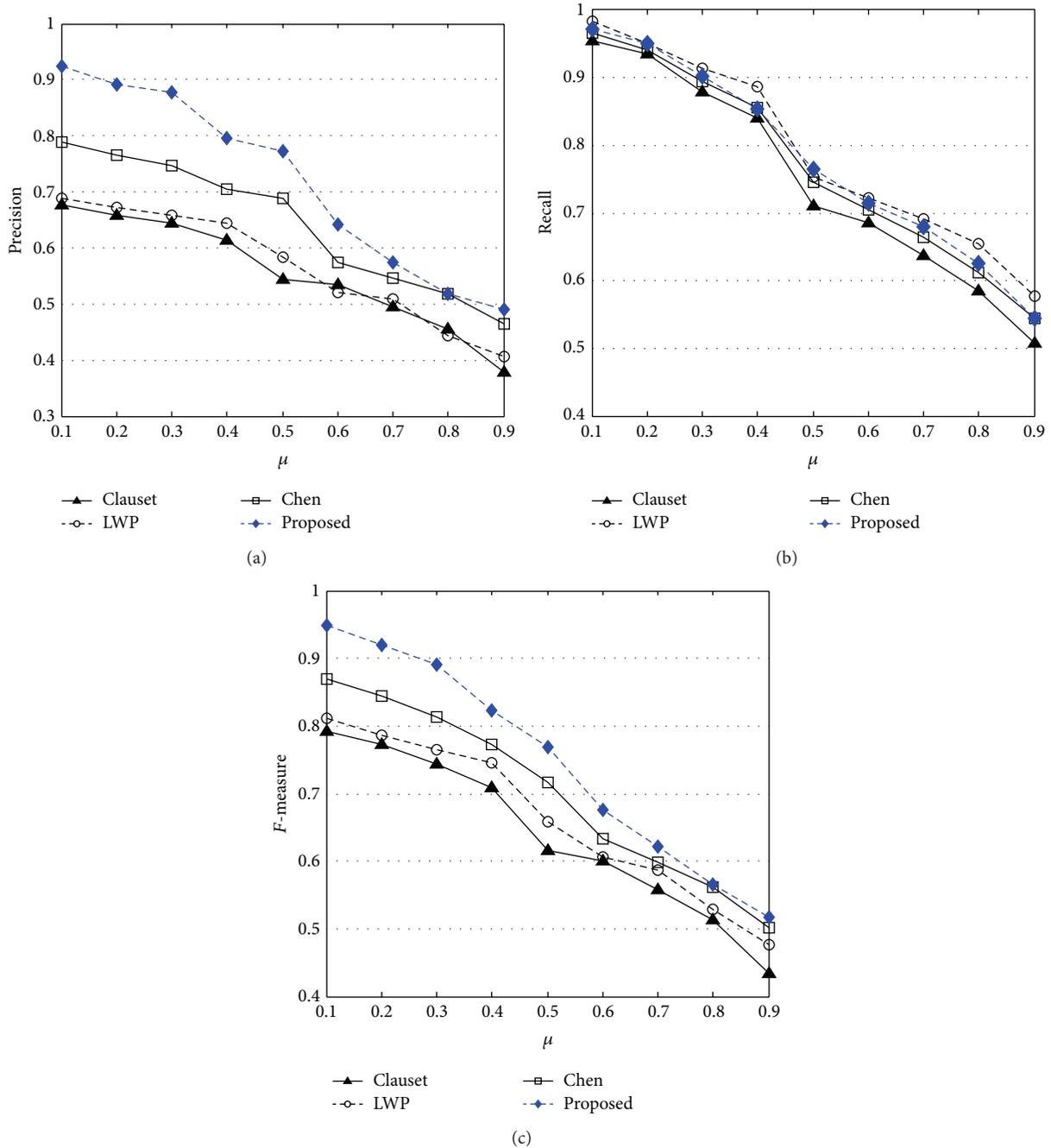


FIGURE 3: Comparison of (a) precision, (b) recall, and (c) F -measure for different methods on the LFR networks.

basic properties (about nodes, edges, etc.) are presented in Table 1.

The Louvain method (LM) [8] is one of the most popular algorithms of global community detection in terms of accuracy and computational costs and provides excellent performance even when the networks to process are very large. For the real networks with unknown community structure, here we use LM to detect the global community structure of real-world networks and find the local community that the starting node belongs to as the real local

community structures. For each network, we randomly select 100 nodes to find the local community with four algorithms, respectively, and compare the results with the real community to test the stability and accuracy of our algorithm.

We next focus on results obtained on real-world network datasets. Here take Football and Polblogs networks as examples; we compare the results of different algorithms with three evaluation metrics, respectively. In Figure 4, it shows that the performance of our algorithm is quite well and balanced in precision, recall, and F -measure. Meanwhile,

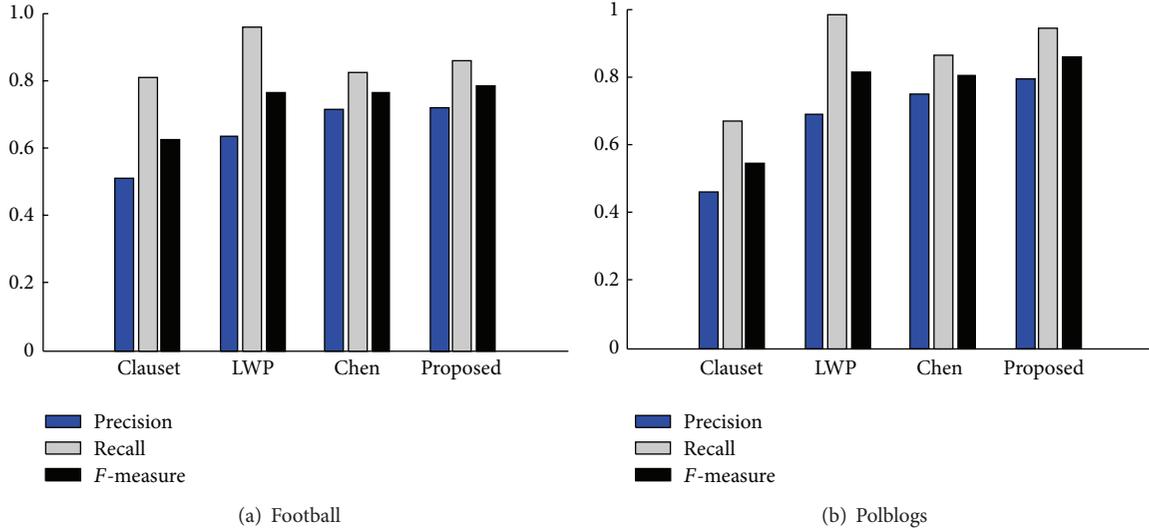
FIGURE 4: Comparison of precision, recall, and F -measure for different methods on Football network.

TABLE 1: The features of the real-world networks for performance evaluation.

| Network | Nodes | Edges | Directed | Community structure |
|---------------|-------|--------|----------|---------------------|
| Football | 115 | 616 | no | known |
| Polblogs | 1490 | 19025 | yes | known |
| US Airport | 1574 | 28236 | yes | unknown |
| US power grid | 4941 | 6594 | no | unknown |
| Internet | 34761 | 171403 | no | unknown |

we can summarize that the existing methods are hardly available for optimal performance in three evaluation metrics together. As shown in Figure 4, LWP and Clauset get a higher recall but a much lower precision, which eventually leads to an unsatisfactory F -measure value. For the Chen method, the three metrics are also less than satisfactory, since these methods are sensitive to the source node's position. When starting from a boundary node s of the local community, the expansion of community C might fall into a different neighbor community, which has some nodes connecting to s . As the algorithm proceeds, the discovered local community would be far from the real local community structure and results in the decrease of the recall and precision.

When detecting the local community structure in the 5 real-world networks, the boundary identification phase in our algorithm can avoid introducing outliers into the local community. The two-phase process can help to discover a more stable and complete community structure as soon as possible. Since F -measure combines the advantages of precision and recall, Table 2 shows the F -measure of all comparison methods in those networks. Compared to the other three algorithms, the average values of F -measure on our method have been improved in different degrees, respectively. For Football and US power grid networks, the improvement of F -measure is relatively slighter. For Polblogs,

TABLE 2: The average evaluation values of F -measure in real-world networks.

| Network | Clauset | LWP | Chen | Proposed |
|---------------|---------|--------|--------|----------|
| Football | 0.6267 | 0.7623 | 0.7633 | 0.7815 |
| Polblogs | 0.5433 | 0.8111 | 0.8025 | 0.8603 |
| US Airport | 0.4950 | 0.7833 | 0.8257 | 0.9145 |
| US power grid | 0.6882 | 0.7452 | 0.7584 | 0.8160 |
| Internet | 0.6319 | 0.8573 | 0.8726 | 0.9314 |

US Airport, and Internet networks, the average values of F -measure on our algorithm demonstrated a much higher increase. The local community detection method based on boundary identification demonstrated much better performance. Even in the complex large networks, it also keeps excellent accuracy and stability. Therefore, our method is more suitable for mining the meaningful local community structure in the networks.

6. Conclusions

In this paper, we proposed a novel local community detection method based on boundary identification. Compared with other existing algorithms, our algorithm is comparatively simpler and more effective to discover the full structure of the local community, because our approach is capable of locating the boundary of the local community by the two-phase iterations, it precisely controls the expansion of the local community, and needs not to require some extra parameters. Experimental results on the synthetic and real-world datasets all show that our algorithm is an effective method to explore local community structures. Moreover, the method works stably and effectively in most networks and can be used freely to acquire communities from the random given node.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Science and Technology Major Project of the Ministry of Science and Technology of China under Grant no. 2012ZX03006002 and the National High Technology Development 863 Program of China under Grant no. 2011AA010604.

References

- [1] B. Shen, Z. Wang, and X. Liu, "Sampled-data synchronization control of dynamical networks with stochastic sampling," *IEEE Transactions on Automatic Control*, vol. 57, no. 10, pp. 2644–2650, 2012.
- [2] A. Mehrsaia, H. R. Karimib, and K. Thobena, "Integration of supply networks for customization with modularity in cloud and make-to-upgrade strategy," *Systems Science & Control Engineering*, vol. 1, no. 1, pp. 28–42, 2013.
- [3] N. Sharma and K. Singh, "Neural network and support vector machine predictive control of *tert*-amyl methyl ether reactive distillation column," *Systems Science & Control Engineering*, vol. 2, no. 1, pp. 512–526, 2014.
- [4] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3–5, pp. 75–174, 2010.
- [5] S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos, "Community detection in social media," *Data Mining and Knowledge Discovery*, vol. 24, no. 3, pp. 515–554, 2012.
- [6] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, no. 4, pp. 1118–1123, 2008.
- [7] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato, "Finding statistically significant communities in networks," *PLoS ONE*, vol. 6, no. 4, Article ID e18961, 2011.
- [8] V. D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, Article ID P10008, 2008.
- [9] A. Clauset, "Finding local community structure in networks," *Physical Review E*, vol. 72, no. 2, Article ID 026132, 2005.
- [10] F. Luo, J. Z. Wang, and E. Promislow, "Exploring local community structures in large networks," *Web Intelligence and Agent Systems*, vol. 6, no. 4, pp. 387–400, 2008.
- [11] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New Journal of Physics*, vol. 11, Article ID 033015, 2009.
- [12] Q. Chen, T. Wu, and M. Fang, "Detecting local community structures in complex networks based on local degree central nodes," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 3, pp. 529–537, 2013.
- [13] X. Zhang, L. Wang, Y. Li, and W. Liang, "Extracting local community structure from local cores," in *Database Systems for Advanced Applications: Proceedings of the 16th International Conference, DASFAA 2011, International Workshops: GDB, SIM3, FlashDB, SNSMW, DaMEN, DQIS, Hong Kong, China, April 22–25, 2011*, vol. 6637 of *Lecture Notes in Computer Science*, pp. 287–298, Springer, Berlin, Germany, 2011.
- [14] Y. J. Wu, H. Huang, Z. F. Hao et al., "Local community detection using link similarity," *Journal of Computer Science and Technology*, vol. 27, no. 6, pp. 1261–1268, 2012.
- [15] X. Zhang, L. Wang, Y. Li, and W. Liang, "Extracting local community structure from local cores," in *Proceedings of the 6th International Conference Database Systems on Advanced Applications*, pp. 287–298, April 2011.
- [16] M. Fanrong, Z. Mu, Z. Yong, and Z. Ranran, "Local community detection in complex networks based on maximum cliques extension," *Mathematical Problems in Engineering*, vol. 2014, Article ID 653670, 12 pages, 2014.
- [17] S. Xia, R. Zhou, Y. Zhou, and M. Zhu, "An improved local community detection algorithm using selection probability," *Mathematical Problems in Engineering*, vol. 2014, Article ID 406485, 10 pages, 2014.
- [18] X. Qi, W. Tang, Y. Wu, G. Guo, E. Fuller, and C. Zhang, "Optimal local community detection in social networks based on density drop of subgraphs," *Pattern Recognition Letters*, vol. 36, pp. 46–53, 2014.
- [19] J. P. Bagrow, "Evaluating local community methods in networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 5, Article ID P05001, 2008.
- [20] F. Fouss, A. Pirotte, J. M. Renders, and M. Saeuens, "Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 3, pp. 355–369, 2007.
- [21] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, no. 23, pp. 8577–8582, 2006.
- [22] Z. Feng, X. Xu, N. Yuruk et al., "A novel similarity-based modularity function for graph partitioning," in *Data Warehousing and Knowledge Discovery*, pp. 385–396, Springer, Berlin, Germany, 2007.
- [23] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical Review E*, vol. 78, no. 4, Article ID 46110, 2008.
- [24] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, 2002.
- [25] L. Adamic A and N. Glance, "The political blogosphere and the 2004 US election: divided they blog," in *Proceedings of the 3rd International Workshop on the Weblogging Ecosystem*, pp. 36–43, ACM, New York, NY, USA, 2005.
- [26] USA power grid network dataset-KONECT, 2013, <http://konect.uni-koblenz.de/networks/opsahl-powergrid>.
- [27] "USA airports network dataset-KONECT," 2013, <http://konect.uni-koblenz.de/networks/opsahl-usairport>.
- [28] B. Zhang, R. Liu, D. Massey, and L. Zhang, "Collecting the Internet AS-level topology," *SIGCOMM Computer Communication Review*, vol. 35, no. 1, pp. 53–61, 2005.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

