A Local Greedy Search Method for Detecting Community Structure in Weighted Social Networks*

Bin Liu[1] and Tieyun Qian[2]

[1] Computer School, Wuhan University, Wuhan, 430071, China
binliu@whu.edu.cn
[2] State Key Lab of Software Engineering, Wuhan University, Wuhan, 430071, China
qty@whu.edu.cn

Abstract. In this paper, we give a new definition of community which is composed of two parts: community core and the periphery. Community core consists of highly densely connected nodes. And we propose LGSM (Local Greedy Search Method) for discovering community structures in social networks. LGSM sorts node according to weighted degree. For each node, LGSM derives a maximal weighted clique as a seed cluster. Then, LGSM adds new nodes into the seed cluster until the weighted edge density is smaller than the threshold value. After all community cores are detected, LGSM allots isolated nodes to the detected cores, and optimizes the community structure based on modularity. Our method is an integrative method, which is applicable not only to discovering overlapping communities, but also to discovering non-overlapping community. Experiments illustrate that LGSM can achieve good community structure on synthetic and real-world networks and the time complexity is O(|E|lg(|V|)).

Keywords: overlapping, community core, community structure.

1 Introduction

Nowadays, researchers have found that many real-world networks possess community structure, such as large-scale social networks, Web graphs, and biological networks. This implies that the network are naturally partitioned into groups of nodes with dense internal connections while sparse connections among groups [1-5]. For example, communities in biological networks may imply functional modules [2]; communities in a citation network might indicate related papers on a research topic [2] [6], and communities in social networks represent people with common interest or background [2] [5]. Identifying these sub-structures within a network can provide insight into the network's function and interaction among communities. In real social networks, every individual typically belongs to more than one community, such as the community of his family, the community of his joining club, the community of his co-workers.

Figure 1 is a piece of research collaboration network and its community structure which is divided into two communities symbolized by circle and square. Each edge is

* This research is supported by NSFC Projects (61070011 and 61272275).

H. Motoda et al. (Eds.): ADMA 2013, Part I, LNAI 8346, pp. 360–371, 2013.

assigned a nonnegative real value to evaluate the strength of the collaboration. And we assume the collaboration is closer, the value is greater. Node 3 is engaged in interdisciplinary research. In community C1, we find that node 6 is only connected with node 2, and the subgraph consisted of node 0, 1, 2, 3, 4 is highly densely connected. We regard the subgraph as the core of C1, and Node 6 as the community periphery. Node 3 should belong to C1, and C2 community.

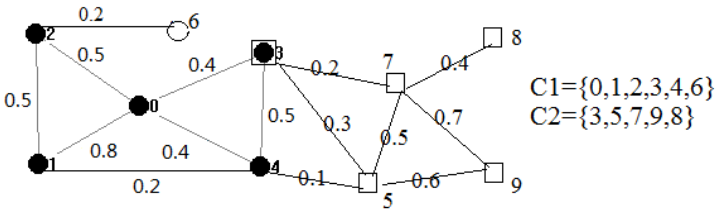

Fig. 1. A weighted network G

Our main contributions are summarized as follows.

We propose community a new definition that community equals community core and community periphery which characterizes different role of nodes in the community.

We design LGSM (local greedy search method) to find the community structure in social networks. LGSM is an integrative method, which is applicable not only to discovering overlapping communities, but also to discovering non-overlapping community.

Experimental results show that LGSM algorithm outperforms the most recent, efficient technique, towards both community accuracy and efficiency when the community structure is well known. By further experiments on synthetic networks, the results also show that LGSM method has high scalability on the graph size.

The rest of the paper is organized as follows. In section 2 we formulize the conceptions used in LGSM. In section 3, we describe the algorithms in detail. In section 4, LGSM is applied to different benchmark networks and compare its performance with several baseline methods. Section 5 introduces the related works. Finally, we summarize our conclusions and suggest future work in section 6.

2 Preliminaries

A social network can be modeled by a weighted graph $G = (V, E, \omega)$, where V is node set, E is edge set. The cardinality of V and E are |V| and |E| respectively. An edge between nodes u and v is represented by e_{uv}. Edge weight can be represented as a function ω: E→R that assigns each edge $e_{uv} \in E$ a value $\omega(e_{uv})$. In this paper, higher $\omega(e_{uv})$ value means high linking strength between u and v. In Figure 1, the weight of edge <2, 1> is 0.5.

2.1 Weighted Degree

The weighted degree of node v, $wd(v)$, is defined as the sum of the weights of its incident edges [9], represented in (1). $wd(0)$ is 2.1 in Figure 1.

$$wd(v) = \sum_{<u,v>\in E} \omega(e_{uv}) \tag{1}$$

2.2 Weighted Edge Density of Subgraph

A graph $G' = (V', E', \omega)$ is a subgraph of the graph G if $V' \subseteq V$ and $E' \subseteq E$. The cardinality of V' is $|V'|$. The weighted edge density of G' is calculated by (2) [9]. For example, C is a subgraph containing node 0, 1, 2, 3, 4, $WED(C)$ equals 0.33.

$$WED(G') = \frac{2\times\sum_{<u,v>\in E'}\omega(e_{uv})}{|V'|\times(|V'|-1)}. \tag{2}$$

2.3 Local Subset, Boundary Subset and Peripheral Subset

For a given subgraph G' of graph G, nodes in G can be partitioned into three parts by G': Local Subset L, Boundary Subset B and Peripheral Subset U, defined in (3).

$$\begin{aligned} L &= \{v \mid v \in G'\} \\ B &= \{v \mid (\exists u)(\exists v)(u \in G') \wedge (v \in G) \wedge (v \notin G') \wedge (e_{uv} \in E)\} \\ U &= G - L - B \end{aligned} \tag{3}$$

For subgraph C, L= {0, 1, 2, 3, 4}, B= {5, 6, 7} and U= {8, 9}

2.4 Internal Weighted Degree

The internal weighted degree of node v to G', $k_v^{in}(G_{sub})$, is defined as the sum of weights of edges between v and the nodes in G', as shown in (4). For subgraph C, and node 3, 5, $k_3^{in}(C) = 0.9$, and $k_5^{in}(C) = 0.4$

$$k_v^{in}(G') = \sum_{u\in G'} \omega(e_{uv}) \tag{4}$$

2.5 Community Core

The community core is a subgraph whose weighted edge density is greater than a given threshold. For a threshold α and a subgraph $C = (V', E', \omega)$, C is a community core if it satisfies (5).

$$WED(C) \geq \alpha \wedge WED(C\{v\}) < \alpha \qquad s.t. < uv > \in E \wedge u \in C \wedge v \notin C \tag{5}$$

If we set α 0.3, $WED(C)$ is 0.33 which is greater than α. And if we add node 5 into C, $WED(C \cup \{5\})$ is 0.25. The WED is smaller than α. So $C \cup \{5\}$ is not a community core. It is NP-hard problem to find all subgraphs with weighted edge density greater than α. LGSM will adopt a heuristic search strategy to find them.

3 LGSM Method

3.1 Obtaining Community Core

LGSM chooses the seed nodes from the social network and uses local search strategy to mine community cores from those seed nodes. Seeds are very important for LGSM. A clique has been shown to be a better alternative over an individual node as a seed [6], [7].

Firstly, LGSM employs weighted degree to sort nodes. After choosing a node v, LGSM derives the max weighted clique from v and its neighbors as seed subgraph. Then all nodes in the remainder network are split into three subsets: L, B and U.

The second step is to expand the seed subgraph to obtain a community core with its weighted edge density greater than a given α. Here we adopt two heuristic search rules, (6) and (7), to expand the seed cluster by selecting the appropriate node v from B and adding it into the L.

$$k_v^{in}(L) \geq \forall k_u^{in}(L) \qquad v,u \in B \tag{6}$$

$$(1+\beta)k_v^{in}(L) \geq k_v^{ex}(L) \tag{7}$$

Rule (6) makes the edge density of community core may be greater than α. Rule (7) makes some nodes to become overlapping nodes if β is greater than 0. If β equals 0, LGSM can mine non-overlapping community structure.

When a community core is found and cannot be enlarged any more. LGSM will choose another seed node and repeats above procedure until all community cores are discovered. In LGSM, the seed node cannot be regarded as the overlapping node.

Here we introduce Edge Density Lemma to prove that the searching method is effective.

Lemma 1. For a given subgraph $G' = (V', E', \omega)$ and its boundary subset B, $v \in B$, if $k_v^{in}(G') \geq |V'| \times WED(G')$, then $WED(G' \cup \{v\}) \geq WED(G')$

.

The pseudo code of LGSM is shown in PROGRAM LGSM. When LGSM chooses the node having maximal internal weighted degree from subset B and adds it into subset L during local search, LGSM compares current $WED(C)$ with α. The result includes three cases: 1) If $WED(C)$ is greater than α. LGSM repeats the above procedure until $WED(C)$ is smaller than threshold α (from 4 to 10 lines). 2) If $WED(C \cup \{v\})$ is smaller than α but greater than $WED(C)$. LGSM repeats the above procedure until $WED(C)$ reaches maximum value (from 12 to 17 lines). If the maximum value of $WED(C)$ is

greater than α, LGSM will repeat the procedure until $WED(C)$ is smaller than α (from 18 to 25 lines). 3) If $WED(C \cup \{v\})$ is smaller than α and also smaller than $WED(C)$. LGSM will stop searching and choose next seed to repeat above procedure.

```
Program LGSM
input:
   Seed Node v
   α: the threshold of weight edge density
   β: overlapping parameter
Output:
  Community Core C
(1)begin
(2)   C= MaxWeightClique(v);
(3)   Initialize B, U;
(4)     if (WED(C)≥α) then
(5)       v=max( k_v^in (C) ;
(6)       while (WED(C∪{v})≥α) do
(7)         C = C∪{u};
(8)         update B and U;
(9)         v=max( k_v^in (C) ;
(10)      end
(11)    else
(12)      v=max( k_v^in (C) ;
(13)      while (WED(C∪{v})≥WED(C)) do
(14)        C = C∪{v};
(15)        update B and U;
(16)        v=max( k_v^in (C) ;
(17)      end
(18)      if (WED(C)≥α) then
(19)        v=max( k_v^in (C) ;
(20)        while (WED(C∪{v})≥α) do
(21)          C = C∪{v};
(22)          update B and U;
(23)          v=max( k_v^in (C) ;
(24)        end
(25)      end
(26)    end
(27)  return C
(28)end
```

Figure 2 illustrates the process of obtaining a community core. In step 1, LGSM finds three cliques: C1= {0, 1, 2}, C2= {0, 3, 4}, C3= {0, 1, 4}. But the weight of C1 is 1.8 greater than C2, and C3. So LGSM applies C1 as the seed subgraph.

In step 4, although node 5 satisfies (6) and (7), the edge density is 0.24 after adding node 5. LGSM cannot add it into the community core and stop search.

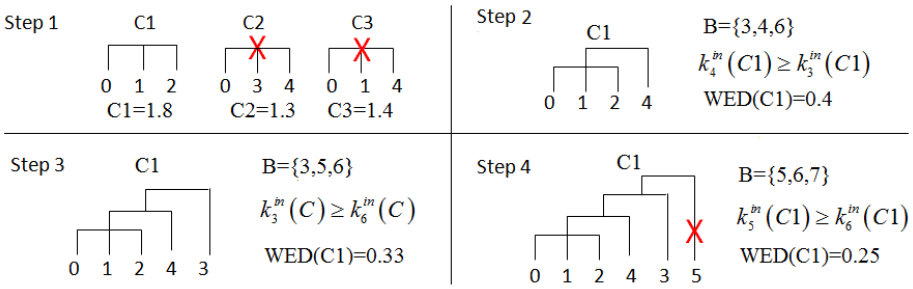

Fig. 2. Obtaining a community core from node 0 in Figure1 with α =0.3, β =0.2

3.2 Allotting Isolated Nodes

After all communities are discovered, each isolated node, which does not belong to any community core, is allotted to the core which the isolated node has the maximal internal weighted degree to. We get the community structure of the network which is called the initial partition.

3.3 Optimizing Community Structure

LGSM intends to split large community into small communities because LGSM only chooses the node matching (6) and (7) during the expanding process. We select the modularity [2] [5] [7], Q, as the object function to optimize community structure because of its practical effectiveness and efficiency. Suppose the initial partition contains s communities, and C_i and C_j (1<=i, j<=s) are two communities, modularity can be calculated by defining modularity matrix e with s dimension. The diagonal elements e_{ii} equal to the sum of weight of edges which fall within C_i. And e_{ij} is one-half of the weight sums of the edges between nodes in C_i and nodes in C_j. Let $a_i = \sum_{j=1}^{s} e_{ij}$,be the fraction of edges attached to nodes in C_i. Q can be calculated by (8).

$$Q = \sum_{i=1}^{i=s} (e_{ii} - a_i^2) \tag{8}$$

If there are no edges between C_i and C_j, merging them can never increase Q. We need only consider those pair communities having edges between them. The gain in Q upon merging two communities is given by (9).

$$\Delta Q = e_{ij} + e_{ji} - 2a_i a_j = 2(e_{ij} - a_i a_j) \tag{9}$$

LGSM merges two communities which can achieve the maximal gain of Q until Q reaches its maximal value.

4 Experiments

In this section, we evaluate LGSM by using synthetic benchmark datasets and four real-world datasets. In overlapping community structure, we compare LGSM with EAGEL [8], Game [9], and CFinder [10] methods. In non-overlapping community structure, we compare with our method with several representative community detection methods: DA [14], SM [3], and FM [4].

In experiment comparison, Normalized Mutual Information (NMI) is adopted to evaluate the quality of clusters generated by different algorithms [5] [7], which is currently widely used in measuring the performance of community detection algorithms. Given two community structures A and B of the same network G, A is the real and B is the detected. Suppose N is the confusion matrix whose element N_{ij} is the number of nodes in both C_i of A and C_j of B, then NMI(A, B) is defined in (10), where $N_{i.}$ is the sum over row *i* of N and $N_{.j}$ is the sum over column *j* of N.

High value of the NMI(A, B) indicates that the detected partition has high similarity with the real one. The two partitions are exactly equivalent if NMI(A,B)=1 while the two partitions are definitely different if NMI(A,B) = 0.

$$\mathrm{NMI}(\mathrm{A},\mathrm{B}) = \frac{-2\sum_{i=1}^{k}\sum_{j=1}^{k} N_{ij}\log(N_{ij}\,|V|/N_{i.}N_{.j})}{\sum_{i=1}^{k} N_{i.}\log(N_{i.}/|V|) + \sum_{j=1}^{k} N_{.j}\log(N_{.j}/|V|)} \tag{10}$$

LGSM has two parameters α and β. For a given social network G, and a community core C, $WED(C)$ should be greater than $WED(G)$ because G is a sparse network. In the following experiments, α is set 5 times of $WED(G)$. The parameter of β controls the numbers of overlapping nodes. β is given a default value 0.2.

4.1 Experiments on Synthetic Networks with Overlapping Community Structure

Lancichinetti-Fortunato-Radicchi (LFR) algorithm is used to generate benchmark graphs [5][6] [7]with overlapping community structure. Some important parameters of the benchmark networks are listed in Table 1.

Two type weighted networks are generated with the number of node |V|=1000 and |V|=5000 [6]. By varying the parameters of the networks, we can analyze the behavior of the algorithms in detail. The mixing parameter μ is taken from the range {0.1, 0.3}. The average degree is <k> = 10, while the maximum degree is maxk = 50.

Table 1. Important Parameters of LFR algorithm

Parameters	Meaning
|V|	number of nodes
<k>	average degree of the nodes
maxk	maximum degree
μ	mixing parameter, each node shares a fraction<k> of its edges with nodes in other communities
minc	minimum for the community sizes
maxc	maximum for the community sizes
O_N	fraction of overlapping nodes of the whole network
O_m	number of memberships of the overlapping nodes

And community sizes vary between minc = 20 and maxc = 100. We set ON to be 10% of the total number of nodes, Om to vary from 2 to 8 indicating the diversity of overlapping nodes. For each network, we generated 10 instantiations. We set α= 0.15 and apply LGSM to find the overlapping community structure.

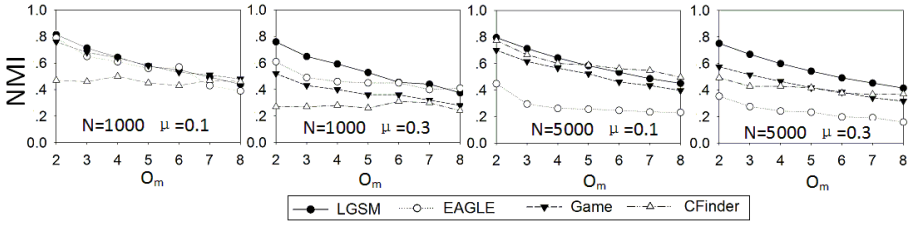

Fig. 3. Comparison with EAGLE, Game, and CFinder on computer-generated networks. Each point corresponds to an average over 100 graph realizations.

Figure 3 shows the comparison result. LGSM is better than EAGLE [8], Game [9], and CFinder [10] in both type networks. As the Om value increases, a node belongs to more communities. Then the community structure becomes fuzzy. This makes that the algorithm to detect community structure accurately is becoming more and more difficult. The accuracy of the methods is reducing with the Om value increasing except CFinder, which is not stable in |V|=1000.

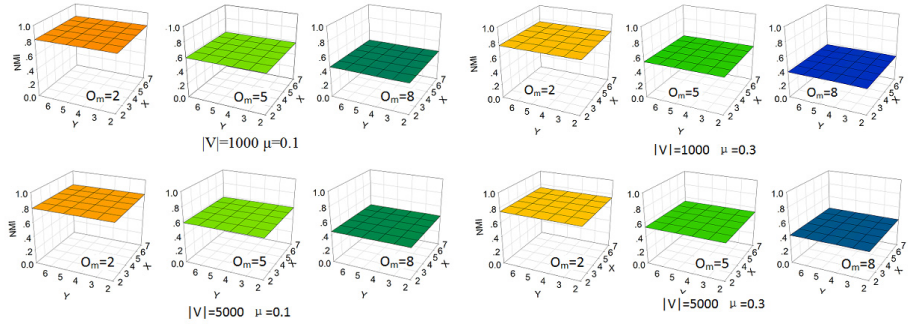

Fig. 4. The influence of α and β on prediction accuracy of LGSM. The x-axis varies from 2 to 7 which is the result of α divided by 0.05.

Then we analyze the influence of α and β on prediction accuracy of the LGSM. The value of α is set from 2 times of $WED(G)$, to 7 times of $WED(G)$. And β is set 0.1 to 0.35. The experiment results are shown in Figure 4.From Figure 4, when μ is fixed, the value of α and β has little effect on the prediction accuracy of LGSM. The prediction accuracy of LGSM is affected by μ more.

4.2 Experiments on Synthetic Networks with Non-overlapping Community Structure

In this section we analyze the performance of LGSM in detecting non-overlapping community structure. If the value of αis set 0, a node can only belong to one community. The community structure is non-overlapping.

In order to compare with existing algorithms better, LFR algorithm is applied to generate two weighted undirected networks with the number of nodes |V|=1000 and |V|=5000. For each network, three individual networks are generated with 15, 20 and 25 as average node degree, with μ varying from 0.1 to 0.6 with a span of 0.1.

DA [14], SM [3], and FM [4] community detection algorithms are chosen as the baseline methods. DA is a global optimization method which employs modularity as objection function. SM is a classical spectral clustering method to detect community structure. FM is a local optimization method which also uses modularity as objection function. All these algorithms are free of parameters. The comparison results are shown is Figure 5. It can be observed that the accuracy of DA and LGSM is almost same, and they are better than SM, and FM.

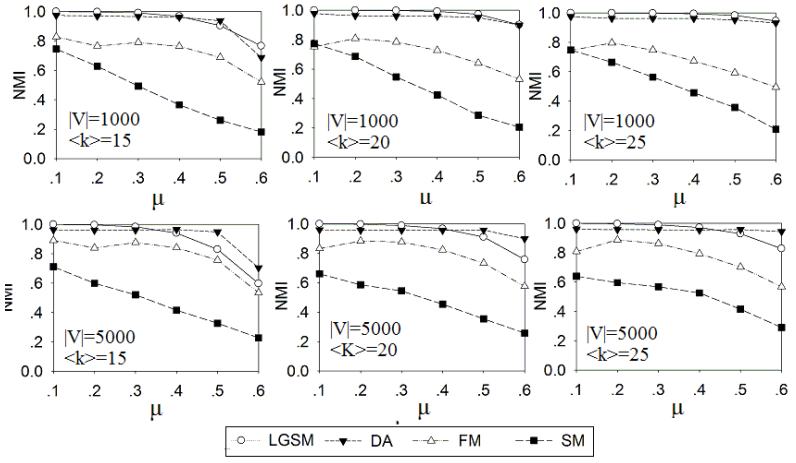

Fig. 5. Comparison with FM, DA, and SM algorithms on computer-generated networks and each point corresponds to an average over 100 graph realizations

4.3 Experiments on Real-World Networks with Ground Partition

In this section, LGSM is applied on four real-world networks: “Karate Club Network”, “Dolphin Social Network”, “American College Football”, and “Books about

US Politics" [2] [3], which community structure is all known. Table 1 shows the comparison result of LGSM, and existing algorithms such as FM, DM, and SA. As can be seen, DA gets best results in most of cases. For "Karate Club", "American Football network", and "Books about US Politics", the community structure detected by LGSM is as good as that detected by DA.

Table 2. Comparison LGSM, with representative community detection algorithms and each cell is is a NMI value corresponding to the deteced community structure and the ground truth

Algorithm	Karate Club	American College Football	Dolphins	Books about US Politics
LGSM	0.71	0. 87	0.58	0.49
FM	0.69	0.71	0.53	0.53
DA	0.69	0.88	0.62	0.56
SM	0.69	0.70	0.48	0.49

4.4 Running Time Complexity

Finally, we analyze the time complexity of our algorithm LGSM. The complexity of computing degree is $O(|V|<k>)$ where <k> is the average neighbor nodes of each node. LGSM applies heap sort to rank nodes, so the time complexity of heap sorting is $O(|V|\lg(|V|))$.

Since the network contains |V| nodes, |V| is the maximal number of seed node. When a seed is identified, LGSM derive a clique from its one-order-neighbor-subgraph. Although the time complexity of finding a clique is high, but LGSM only find the clique from one-order-neighbor-subgraph. The real time cost actually low and ignored here.

The next step is to build the boundary subset and choose the nodes according to their internal weight degree. Since the seed node has $<K>$ neighbors and each neighbor node has $<k>$ neighbors, the boundary subset contains $<k>^2$ nodes at most. The complexity of choosing node is $O(<k>^2)$.

After LGSM updating subset B, the complexity is $O(<k>^2)$. Because the average diameter of a social network is $\lg(|V|)$ [1], the average iterative step of LGSM is $\lg(|V|)$. The total complexity is $O(2|V|<k>^2\lg(|V|)+|V|\lg(|V|))$. Since |V|*<k>=2|E| and the network is a sparse network, the complexity approximates $O(|E|\lg(|V|))$.

Table 3. Time Complexity

Algorithm	Time Complexity
LGSM	$O(\|E\|\lg(\|V\|))$
FM	$O(\|V\|^2)$ or $O(\|V\|(\|V\|+\|E\|))$
DA	$O(\|V\|^2)$ or $O(\|V\|(\|V\|+\|E\|))$
SM	$O(\|V\|^3)$

To illustrate the running time of the proposed algorithms, we generate five networks with the number of nodes |V| ranging from 1,000 to 5,000 with <k> as 20. Figure 8 shows the running time. We observe that LGSM is faster than DA, FM, and SM. Furthermore, we generate larger synthetic networks with the number of nodes |V| ranging from 10,000 to 50,000 with <k> as 15, 20 respectively. We can find that LGSM can process the network of 50,000 nodes within 800 seconds.

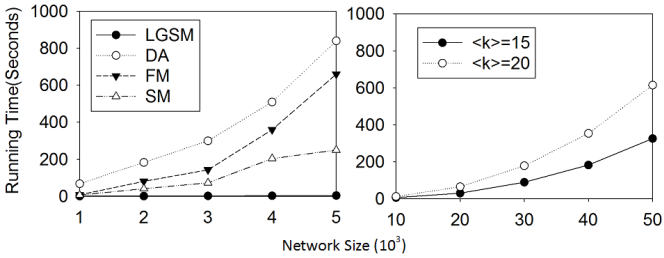

Fig. 6. Runing time on synthetic networks

5 Related Works

Community detection has become a challenge task which has received a great deal of attention in recent years [5] [6] [7]. And many algorithms are put forward, which can be divided into two types: non-overlapping, and overlapping.

Non-overlapping methods. Agglomerative methods merge nodes into a cluster according to some criterion such as node similarity [11], while divisive methods remove edges from the network until the network is split into clusters based on edge or node properties such as betweenness [2] [3]. These methods need to set a condition to stop them [3].

Modularity, denoted as Q, is a benefit function that measures the quality of a particular division of a network into communities which is put forward by Girvan and Newman [3] [7]. And many optimization approaches are proposed to discover communities in a network [12] [13] [14]. Modularity-based methods suffer from time complexity [3] [7].

Overlapping methods. Chen proposed a game-theoretic framework to find overlapping community structure, in which a community is associated with a Nash local equilibrium [9]. Palla designed CFinder method based on the clique percolation [10]. CFinder begins by identifying all cliques of size k in a network. And all k-cliques, sharing k-1 nodes, are regarded as overlapping communities. However it also fails to terminate in many large social networks. EAGLE uses the agglomerative framework to produce a dendrogram [8]. First, all maximal cliques are found and made to be the initial communities. Then, the pair of communities with maximum similarity is merged. The optimal cut on the dendrogram is determined by the extended modularity [7]. And also its time complexity is high.

6 Conclusion and Future Work

In this paper, we partition community into two parts: community core and community periphery according to different roles which the nodes play in a community. And we propose a method, LGSM, to detect community structure in weighted social networks. This method is not only suitable for overlapping community detection but also for non-overlapping community detection. Experiments on synthetic networks and real-world networks show that LGSM can get better performance and has lower time complexity than the benchmark community detection algorithms. Our future work will apply LGSM to investigate the local communities in large-scale online networks and to use our method to analyze complex networks in various applications.

References

1. Newman, M.: Communities, modules and large-scale structure in networks. Nature Physics 8, 25–31 (2012)
2. Newman, M.: Modularity and community structure in networks. PNAS 103(23), 8577–8582 (2006)
3. Girvan, M., Newman, M.: Community structure in social and biological networks. PNAS 99(12), 7821–7826 (2002)
4. Newman, M.: Fast algorithm for detecting community structure in networks. Phys. Rev. E 69, 066133 (2004)
5. Lancichinetti, A., Fortunato, S.: Community detection algorithms: A comparative analysis. Phys. Rev. E 80(5), 056117 (2009)
6. Xie, J., Kelley, S., Szymanski, B.K.: Overlapping Community Detection in Networks: The State of the Art and Comparative Study. ACM Computing Surveys 4 (2013)
7. Fortunato, S.: Community detection in graphs. arXiv:0906.0612 (2009)
8. Shen, H., Cheng, X., Cai, K., Hu, M.-B.: Detect overlapping and hierarchical community structure. Physica A 388, 1706 (2009)
9. Chen, W., Liu, Z., Sun, X., Wang, Y.: A game-theoretic framework to identify overlapping communities in social networks. Data Min. Knowl. Discov. 21, 224–240 (2010)
10. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature 435, 814–818 (2005)
11. Jianbin, H., Heli, S., Jiawei, H., Hongbo, D., Yizhou, S., Yaguang, L.: SHRINK: A Structural Clustering Algorithm for Detecting Hierarchical Communities in Networks. In: CIKM, pp. 219–228 (2009)
12. Clauset, A., Newman, M., Moore, C.: Finding community structure in very large networks. Phys. Rev. E 70, 066111 (2004)
13. Medus, A., Acuna, G., Dorso, C.O.: Detection of community structures in networks via global optimization. Physical A 358, 593–604 (2005)
14. Duch, J., Alex Arenas, A.: Community detection in complex networks using extremal optimization. Phys. Rev. E 72, 027104 (2005)