Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 14 (24): 3532-3537, 2014
ISSN 1812-5654 / DOI: 10.3923/jas.2014.3532.3537

A Novel Algorithm for Detecting Local Community Structure Based on Hybrid Centrality

Qiu Li-Qing, Liang Yong-Quan and Chen Zhuo-Yan
College of Information Science and Technology,
Shandong University of Science and Technology, Qingdao, Shandong, 266590, China

Abstract: Community detection has been a research topic in the complex network area. The global information of the whole network, which is required by the traditional community detection algorithms, is hard to get when the scale of the network grows. The study presents a novel algorithm for detecting local community structure based on hybrid centrality. After identifying the network nodes with hybrid centrality, our algorithm can detect local community structure starting from some important nodes. In addition, to better understand the algorithm, a subsequent processing is continued. The present algorithm is applied to some simple examples, including computer-generated and real-world networks. And the experimental results are analyzed by comparing with other traditional algorithms.

Key words: Community, community structure, hybrid centrality, novel algorithm

INTRODUCTION

In last few years, the networks have found use in many fields as a powerful tool for representing the structure of complex systems. Data from many network datasets, including the Internet and the World Wide Web in computer and information sciences, is a graph where nodes represent individuals and edges represent the relationship and interactions among individuals. In the graphs, it is important to be able to group the nodes into what is commonly known as communities. The problem of community detecting in these networks represents one of the most challenging and promising perspectives to approach, characterize and understand the general structures.

The last years have seen an increase in the number of techniques proposed to detect communities. However, each of these techniques requires knowledge of the entire structure of the graph, as we will discuss in related work. This constraint is problematic for complex networks, which for all practical purposes is too large and too dynamic to ever be known fully, or the networks which are larger than can be accommodated by the fastest algorithms. Here, a general measure of local community structure based on hybrid centrality was proposed, for networks in which we lack global knowledge. The proposed hybrid method, which leverages degree centrality and cohesion centrality, can detect important nodes in the network. The proposed algorithm requires only the local network information related to the target node and is faster compared to the traditional community detecting algorithm. Moreover, the proposed algorithm is also applicable for global community structure detecting.

A community could be loosely described as a collection of nodes within a graph that are densely connected among them while being loosely connected to the rest of the graph (Wasserman and Faust, 1994a; Flake *et al*., 2002; Radicchi *et al*., 2004). Many networks exhibit such a community structure and this motivates the present study. Traditional techniques for community detection tend to consider the global topology of networks, which aim to group nodes of networks into a number of disjoint sets. Typically, the techniques aim to optimize a criterion defined over networks partition rather than over one group. The recent and highly successful algorithms, such as spectral clustering (Von Luxburg, 2007) and Modularity clustering (Newman, 2006a), perform well within a variety of networks.

However, a common weakness in these studies is that the computation of measurements may be expensive. More importantly, while these algorithms have shown to be a useful quantity for detecting community structure, the global information of the whole network, which is required, is hard to get when the scale of the network grows. For large-scale networks, efficient algorithms of community detection are critical and require further research. This explains the increasing interest in detecting local community structure rather than global community

Corresponding Author: Qiu Li-Qing, College of Information Science and Technology,
Shandong University of Science and Technology, Qingdao, Shandong, 266590, China

structure. Costa (2004) presents a hub-based approach to community finding in complex networks. After identifying the network nodes with highest degree (the so-called hubs), the network is grouped from the hubs, accounting for the identification of the involved communities. It is worthy noting that the number of communities detected is arbitrarily pre-assigned. Clauset (2005) proposed a fast agglomerative algorithm (the so-called local modularity) that maximizes the local modularity in a greedy fashion. This algorithm is costly to compute $O(k^2d)$ for general graphs when d is the mean degree and k is the number of nodes to be explored. Bagrow and Bollt (2005) put forward the algorithm works by l-shell spreading outward from a starting vertex and computing the change in total emerging degree to some threshold. The algorithm tends to join an overall l-shell to the community, or excludes the overall l-shell outside the community, which shows not perfect.

In spite of these limitations, we would like to make quantitative statements about local community structure. For instance we might like to quantify the role of nodes in networks and the relations of important nodes are also analyzed in detail.

ALGORITHM

As emphasized by the several investigations targeting complex networks, some important nodes play determinantal role in defining the connectivity patterns. Therefore, the consideration of important nodes as starting points for community detection represents a particular promising perspective from which to approach the community detection. Centrality analysis provides answers with measures that define the importance of nodes. There are many classical and commonly used methods: Degree centrality, closeness centrality, between centrality and eigenvector centrality (Wasserman and Faust, 1994b). These centrality measures capture the importance of nodes in different perspectives. However, with large-scale networks, the computation of centrality measures would be expensive except for degree centrality.

We propose a simple and powerful local community detection algorithm. Some important nodes are first found according to their degree centrality and cohesion centrality and then started from important nodes an alternative breadth-first search is conducted to get the local community structure of the nodes.

The proposed algorithm is described roughly as follows:

- Select some important nodes
- Detect local community structure starting from some important nodes

In addition, to better understand the algorithm, a subsequent processing is continued.

Notations: The study focus on a simplest form of networks, i.e., undirected networks with boolean edge weights. The notations that will be used frequently throughout the study are summarized in Table 1.

Selecting important nodes: For degree centrality, the importance of a node is determined by the number of nodes adjacent to it. The larger the degree of one node, the more important the node is. Node degrees in most networks follow a power law distribution, i.e., a very small number of nodes have an extremely large number of connections. Those high-degree nodes naturally have more impact to reach a large population than the remaining nodes within the same network. Moreover, degree centrality, as one of the most simple centrality measurement, requires not much computation, which shows more suitable to large-scaled networks. However, degree centrality is inefficient under such scenarios, i.e., some important nodes don't have high degree centrality.

In Fig. 1, the network represents the extreme fulfillment of the idea of a community. Each community has the maximum number of internal links possible while having close to the minimum number of external links. In addition, the network contains single node (i.e., node 6) situated between the communities. If we use degree centrality as the measurement, node 6 has

Table 1: Notations

Symbol	Meaning
V	Set of nodes in the network
E	Set of edges in the network
n	No. of nodes (n = |V|)
m	No. of edges (m = |E|)
v	A node v
e(v, t)	An edge between nodes v and t
N_v	Neighborhood (i.e., neighboring nodes) of node v
d_v	Degree of node v ($d_v = |N_v|$)
e_v	No. of edges between v and its neighbors

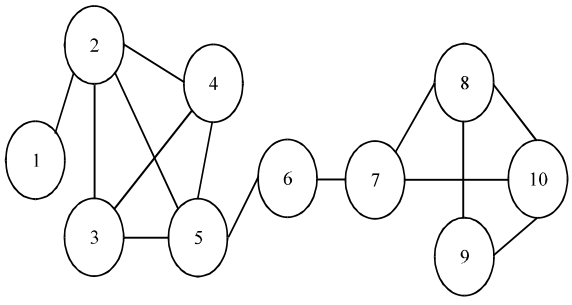

Fig. 1: A simple network of 10 nodes and 14 edges. The network can be grouped into three communities: {1, 2, 3, 4, 5}, {7, 8, 9, 10} and {6}. Node 6 is a single node, which joints the two communities

the same degree with node 9 and their degree is 2. However, node 6 plays a more important role actually, because it is on the boundary of two different communities.

The non-homogenous topology of complex networks determines that the importance of nodes is different. The importance of the nodes primarily depends on their position in the networks. For example, "Peripheral nodes" and "Non-peripheral nodes" have different importance obviously, which can be seen from Fig. 1, i.e., node 1 as the peripheral node would not have the same important role as other non-peripheral nodes. Similarly, "Central nodes" and "Non-central-nodes" are clearly different from the degree of importance. Secondly, the importance of the nodes depends on their role in the network. As we can see from Fig. 1, although node 6 and 9 have the equal degree, node 6 joints the two communities which means that the deletion of node 6 will make more affect than the deletion of node 9. In other words, we can see that node 6 is more important than node 9 from the idea of role. Thus when we select important nodes, the role of nodes should be considered. But how should we measure the role of the nods? Here, we use the definition of cohesion centrality to measure the role of the nodes.

Let us first define the following normalized degree centrality:

$$C_D(v) = d_v/(n-1) \quad (1)$$

Then, the definition of cohesion centrality was given to describe the role of the nodes. From Fig. 1, it is observed that the role of the nodes is determined by the number of neighbors, i.e., if the node has more neighbors, then the deletion of the node will make less affect to the network. Therefore, we consider that cohesion centrality should tend to reflect the local connection property of the node. It follows from the following equation:

$$C_c(v) = \frac{d_v(d_v-1)}{2e_i} \quad (2)$$

Obviously, the span of e_i is between 0 and $d_v(d_v-1)/2$. Therefore, the value of $C_c(v)$ satisfies the conditions:

$$C_c(v) \geq 1 \quad (3)$$

We find that the larger cohesion centrality of one node, the more important the node is. This is because the deletion of the node with larger values will make more affection on the network. Therefore, the cohesion centrality is the positive evaluation measurement for the node.

As we have described, the importance of the nodes depends on their position as well as their role in the network. We define a new hybrid centrality, which uses degree centrality as the measurement of the former and cohesion centrality as the measurement of the latter. Let us define a parameter α to integrate the two measurements:

$$\text{Importance}(v) = \alpha.C_D(v)+(1-\alpha).C_c(v) \quad (4)$$

where, α satisfies $0 \leq \alpha \leq 1$.

The algorithm of selecting important nodes is based on a single parameter α, which controls the tendency of computation. When $\alpha = 0$, the role of node v is only considered. As α increases in size, the algorithm will tend to consider the position of node v until $\alpha = 1$. How to determine the parameter α is beyond the scope of this srudy. To simply the experiments, we set α as 0.5 in the following example.

Detecting local community structure: If we consider some nodes to constitute a local community, the simplest measure of the quality of such a grouping of the network is simply the fraction of known adjacencies that are neighbors of the node. For selected node v, the relation of it and its neighbor t can be defined to satisfy following three conditions:

- If the degree of t is smaller than the degree of v and the degrees of other t's neighbors are smaller than the degree of v, then v is considered as the most important node to t, i.e., t is merged to the community of v
- If the degree of t is larger than the degree of v and the degrees of other v's neighbors are smaller than the degree of t, then t is considered as the most important node to v, i.e., v is merged to the community of t
- If above two conditions are not satisfied, then v and t are not in the same community

Suppose that in the network, we have perfect knowledge of the connectivity of some set of nodes, i.e., the most important node v has been selected according to Eq. 4, which we denote as community c. This necessarily implies the existence of a set of nodes u about which we know only their adjacent to c. Further, let us assume that the only way we may gain additional knowledge about the network is visiting some neighboring node $v_i \in u$, which yields a list of its adjacencies. We place the neighboring nodes in the communities, $v_i = c_i$. Then, we can define whether node v_i belongs to community c, through

comparing the degree of node v and node v_i according to above conditions. At each step, the algorithm updates the starting node when some conditions are satisfied. The process continues until the whole community is discovered.

The algorithm is agglomerative indeed, which maximizes the degree of nodes in a greed fashion. The algorithm only takes time polynomial in n and that infers local community structure by using the node-at-a-time discovery process which is directly analogous to the manner in which spider program harvests the hyperlink structure of the Internet.

Algorithm 1 is presented for more exact pseudocode.

Algorithm 1: Local algorithm to determine a starting node's community

Add v to c//v is the starting node
Add all neighbors of v to u
Add $v_i \in u$ to c_i
while $u \neq \phi$ do
  for each $v_i \in u$ do
  if $d_{vi} < d_v$ and $d_{Nv} < d_v$
    add v_i to c
    update v//$v \leftarrow v_i$
  if $d_{vi} < d_v$ and $d_{Nv} < d_{vi}$
   add v to c//c_i is the community of v_i
   update v//$v \leftarrow v_i$
  end for
end while

Some improvement: In Fig. 1, node 5 is one of the most important nodes according to Eq. 4, which can be seen as the starting node of Algorithm 1. Initially, we place node 5 in the community, {5} = c and place its neighbors in u, {2, 3, 4, 6} = u. At each step, the algorithm adds to c the neighboring node that results in the largest increase in c. After node 2, node 3 and 4 are merged into c, node 6 is analyzed as follows: The degree of node 6 is smaller than the degree of node 5 and the degrees of other node 6's neighbors are smaller than the degree of node 5, thus node 6 should be merged into c. The analogous steps continue, which results in the network only has one community, i.e., {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}. However, this is not our desirable result because the network should be grouped into three communities, i.e., {1, 2, 3, 4, 5}, {7, 8, 9, 10} and {6}.

What results in such undesirable result? Since Algorithm 1 tends to merge the important nodes from different communities into one community. Algorithm 1 works in a greed fashion, which seems lack of reason. In case it is desired to merge important nodes, which is an application-dependent decision, the following post-processing can be performed. Before merging the neighboring node v_i into c, we should add one new step, i.e., judging whether node v_i shares most mutual neighbors with node v. In case the number of mutual neighbors is larger than a pre-specified threshold value β, node v_i is merged into c. The problem of how to automatically find the best β when there is no ground truth is beyond the scope of this study. To simply the experiments, we will set β as 0.5 in the following example applications.

EXPERIMENTAL STUDIES

Several different networks to study the performance of our generalized novel algorithm for detecting local community structure were used.

An idealized network: We started with the first synthetic dataset, which is shown as Fig. 1, to illustrate the process of the algorithm in detail. The network contains 10 nodes which roughly form 3 communities: c_1 = {1, 2, 3, 4, 5}, c_2 = {6} and c_3 = {7, 8, 9, 10}.

In the previous section, we have shown that generalized local community detection algorithm starting from some important nodes. Therefore, we first select the most important node according to hybrid centrality, which is put forward in Eq. 4. Hybrid centrality value of each node is shown in Table 2.

The second column illustrates normalized degree centrality of each node. The third column shows cohesion centrality of each node. And the last column is hybrid centrality value of each node.

Once we have the important value of each node, we can select the most important nodes and hence to find node 2 and 5 as the starting nodes, which have the largest value 0.5824. Rankly, we selected node 2 as the starting important node. Initially, we placeed node 2 in the community, {2} = c_1 and placed its neighbors in u, {1, 3, 4, 5} = u. At each step, the algorithm adds to c_1 the neighboring node that results in the largest increase in c_1. After node 1, node 3, node 4 and node 5 are merged into c_1, node 6 is analyzed as follows. The degree of node 6 is smaller than the degree of node 5 and the degrees of other node 6's neighbors are smaller than the degree of node 5. However, node 5 and node 6 don't share any mutual neighbors. Therefore, node 6 isn't the member of community c_1 and we denote node 6's community as c_2.

Table 2: Hybrid centrality value of each node

Node	$C_D(v)$	$C_c(v)$	Importance (v)
1	0.0769	0	0.0386
2	0.3077	0.8570	0.5824
3	0.2308	0.5000	0.3654
4	0.2308	0.5000	0.3654
5	0.3077	0.8571	0.5824
6	0.1538	0.5000	0.3269
7	0.2308	0.7500	0.4904
8	0.2308	0.7500	0.4904
9	0.1538	0.3333	0.2436
10	0.2308	0.6000	0.4154

The analogous steps continue on until we get three communities (i.e., {1, 2, 3, 4, 5}, {7, 8, 9, 10} and {6}). The result is desirable, which corresponds perfectly to the division observed in real life. In particular, if we start from another important node (i.e., node 5) and we will get the same result, which shows that our generalized algorithm will lead to identical results when the starting nodes are varied.

Through the experiment, we can better understand how to interpret the performance of our algorithm. In addition, it is useful to note that the algorithm dose not require any global information.

Real-world networks: The proposed algorithm performs extremely well on idealized networks, but how does it perform on real-world networks? Here we first analyze the Zachary Karate Club, which is perhaps the most famous network in terms of community structure (Zachary, 1977). The club suffered from infighting and eventually split in half, providing actual evidence of the community structure.

Figure 2 shows the division of this network into two groups found using our new algorithm. All of the nodes which are on the boundary of the communities are grouped correctly. All in all, we get identical results which correspond perfectly to the division observed in real life.

But the algorithm reveals much more about the network than this. Now we draw attention to the runtime of the algorithm and we find some good performance characteristics. We compare our algorithm with some baseline algorithms (i.e., betweenness partitioning (Newman and Girvan, 2004) and eigenvector partitioning (Newman, 2006b). Under the same computing environment, the time of our algorithm need only 0.01 sec,

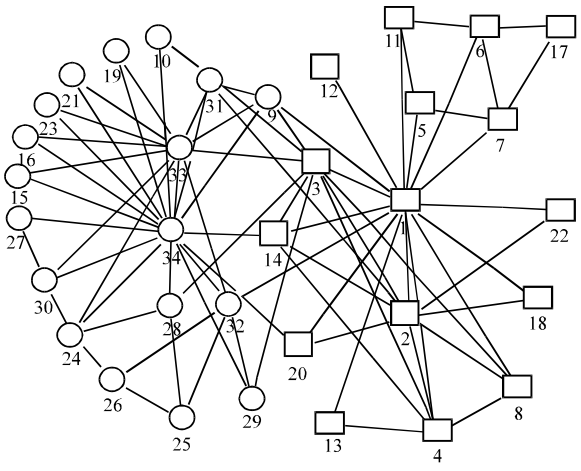

Fig. 2: Zachary Karate Club. The different shapes indicate the membership of the two clusters

which is similar with baseline algorithms. However, how does our algorithm perform on large-scaled networks? Here we analyze the Jazz network (Gleiser and Danon, 2003), which is another famous network in community detection. The Jazz network is more complex than the Zachary Karate Club, which contains 198 nodes and 2742 edges. The runtime of our algorithm is much less than any other baseline algorithms. Our algorithm needs only 0.03 second, while two baseline algorithms need 35.61 and 0.45 sec, respectively. This is because our algorithm works in a greedy fashion and the algorithm only takes time polynomial in n.

CONCLUSION

Community detection is a challenging research problem with broad applications. In this study we have described a general measure of local community structure based on hybrid centrality. We first select some important nodes according to hybrid centrality, which leverages degree centrality and cohesion centrality. Then we detect local community structure starting from the important nodes by using some strategies. The strategies can detect local community structure by determining the relations of some node and its neighbors. In addition, to better the algorithm, a subsequent processing is continued. We have demonstrated the method with applications to some simple examples, including computer-generated and real-world networks. The method's strength is its efficiency which leads to nearly identical results and less runtime compared to some baseline algorithms.

ACKNOWLEDGMENT

This study is supported by National Natural Science Foundation of China under Grant No. 61203305 and Natural Science Foundation of Shandong Province of China under grant ZR2010FQ021 and ZR2012FM003.

REFERENCES

Bagrow, J.P. and E.M. Bollt, 2005. Local method for detecting communities. Phys. Rev. E, Vol. 72. 10.1103/PhysRevE.72.046108

Clauset, A., 2005. Finding local community structure in networks. Phys. Rev. E, Vol. 72. 10.1103/Phys RevE.72.026132

Costa, L.D.F., 2004. Hub-based community finding. pp: 1-4, http://arxiv.org/pdf/cond-mat/0405022.pdf

Flake, G.W., S. Lawrence, C.L. Giles and F.M. Coetzee, 2002. Self-organization and identification of web communities. Computer, 35: 66-70.

Gleiser, A. and P.M.L. Danon, 2003. Community structure in jazz. Adv. Complex Syst., 6: 565-574.

Newman, M.E.J. and M. Girvan, 2004. Finding and evaluating community structure in networks. Phys. Rev. E, Vol. 69. 10.1103/PhysRevE.69.026113

Newman, M.E.J., 2006a. Modularity and community structure in networks. Proc. Natl. Acad. Sci. USA., 103: 8577-8582.

Newman, M.E.J., 2006b. Finding community structure in networks using the eigenvectors of matrices. Phys. Rev., Vol, E 74.

Radicchi, F., C. Castellano, F. Cecconi, V. Loreto and D. Parisi, 2004. Defining and identifying communities in networks. Proc. Natl. Acad. Sci. USA., 101: 2658-2663.

Von Luxburg, U., 2007. A tutorial on spectral clustering. Stat. Comput., 17: 395-416.

Wasserman, S. and K. Faust, 1994a. Social Network Analysis: Methods and Applications. Cambridge University Press, Cambridge, UK., ISBN-13: 9780521387071, pp: 34-68.

Wasserman, S. and K. Faust, 1994b. Social Network Analysis: Methods and applications. Cambridge University Press, Cambridge, UK., ISBN-13: 9780521387071, Pages: 857.

Zachary, W.W., 1977. An information flow model for conflict and fission in small groups. J. Anthropol. Res., 33: 452-473.