How to Detect Communities in Large Networks

Yasong Jiang[1(✉)], Yuan Huang[2], Peng Li[2], Shengxiang Gao[2], Yan Zhang[1], and Yonghong Yan[1]

[1] The Key Laboratory of Speech Acoustics and Content Understanding Institute of Acoustics, Chinese Academy of Sciences, Beijing, China
{jiangyasong, zhangyan, yanyonghong}@hccl.ioa.ac.cn
[2] National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing, China
{huangyuan, gao.shengxiang}@cert.org.cn, l2685581@qq.com

Abstract. Community detection is a very popular research topic in network science nowadays. Various categories of community detection algorithms have been proposed, such as graph partitioning, hierarchical clustering, partitional clustering. Due to the high computational complexity of those algorithms, it is impossible to apply those algorithms to large networks. In order to solve the problem, Blondel introduced a new greedy approach named lovian to apply to large networks. But the remained problem lies in that the community detection result is not unstable due to the random choice of seed nodes. In this paper, we present a new modularity optimization method, LPR, for community detection, which chooses the node in order of the PageRank value rather than randomly. The experiments are executed by using medium-sized networks and large networks respectively for community detection. Comparing with lovian algorithm, the LPR method achieves better performance and higher computational efficiency, indicating the order of choosing seed nodes greatly influences the efficiency of community detection. In addition, we can get the importance values of nodes which not only is part of our algorithm, but also can be used to detect the community kernel in the network independently.

Keywords: Pagerank · Community detection · Modularity · Large network

1 Introduction

We know that real networks are not random and they usually exhibit inhomogeneity, indicating the coexistence of order and organization. The most famous character of the networks is the community structure. Community structure embodies the famous saying that "the birds of a feather flock together". In society, individuals with similar interests are more likely to become friends [6, 16]. In the Web, web pages with related topics are often hyperlinked together [5]. In the protein interaction network, communities are composed of proteins with the same specific function for chemical reactions [3, 17]. In metabolic networks, communities may correspond to functional modules such as cycles and pathways [9]. In food webs, compartments can be viewed as communities [12, 19]. Hence, the community becomes the entry point of researches of

D.-S. Huang et al. (Eds.): ICIC 2015, Part I, LNCS 9225, pp. 76–84, 2015.
DOI: 10.1007/978-3-319-22180-9_8

networks structure and functionality. Community detection is a fundamental research issue and attracts much interest over the last decade.

Community detection is to recognize the inherent structure of networks, i.e., dividing a network into several communities which have high density of edges within communities and low density between them. Nowadays, the most often used method is the modularity optimization-based community detection approach. Precise formulations of this optimization problem are known to be computationally intractable.

Several algorithms have therefore been proposed to find reasonably good partitions efficiently. The first algorithm devised to maximize modularity was a greedy method proposed by Newman [4]. It is an agglomerative hierarchical clustering method, where groups of vertices are successively joined to form larger communities such that modularity increases after the merging. A different greedy approach has been introduced by Blondel [1], for the general case of weighted graphs, which is the best algorithm that can be used in large networks. We take the two methods for comparison on different kinds of sizes of networks.

Besides, we make use of the PageRank algorithm [10] which is applied widely in community kernel detection to evaluate the importance of nodes. PageRank is a link analysis algorithm and it assigns a numerical weighting to each element of a hyperlinked set of documents, such as the World Wide Web, with the purpose of "measuring" its relative importance within the set. As the structure of the World Wide Web is very similar to the structure of network, in which we can regard the nodes as the web pages and the edges as the hyperlink.

Our new algorithm, which we refer to as a smart local moving algorithm, takes advantage of both local moving heuristic and PageRank algorithm. Furthermore, the experimental result verifies the superior performance in modularity and computational efficiency in compare with the lovian algorithm.

The remainder of this paper is organized as follows. In Sect. 2, we present our new algorithm. We analyze the result of community kernel detection and compare the performance of the lovian algorithm with the fastgreedy algorithm in Sect. 3. We first consider small and medium-sized networks, and then focus on large networks. We summarize the conclusions of our research in Sect. 4.

2 Algorithm

Before we introduce our algorithm, we should make a detailed introduction on two crucial concepts, namely PageRank and modularity.

2.1 PageRank

PageRank [10] is a probability distribution used to represent the likelihood that a person would randomly visit a particular webpage. The idea is to imagine a random web surfer visiting a page and randomly clicking links to visit other pages then randomly going to a new page and repeating the process. The original PageRank of page A can be expressed as:

$$PR(A) = \frac{1-d}{n} + d\sum_{a \in W_b} \frac{PR(a)}{L(a)} \tag{1}$$

Where n is the total number of pages in the system and d is the dampening factor that has been tried and tested in numerous studies happens to be about 0.85. W_b is the set of pages connected to page A, *PR(A)* is the *PageRank(A)* and *L(a)* is the number of outbound links on page A.

But the difference from the original PageRank in web page is that the network we study here is undirected. So, we have to change the formula to:

$$PR(i) = \frac{1-d}{N} + d\sum_{j \neq i} PR(j) \frac{w_{ij}}{\sum_{k \in adj[i]} w_{ik}} \tag{2}$$

Where N denotes the number of nodes, $adj[i]$ denotes the set of neighbors of i, and W_{ij} denotes the weight of edge ij.

2.2 Modularity

The modularity function [4] of Newman and Girvan is based on the idea that a random graph is not expected to have a cluster structure, thus the possible existence of clusters is revealed by the comparison between the actual density of edges in a subgraph and the density one would expect in the subgraph if the vertices of the graph were attached regardless of community structure. So it is written as:

$$Q = \frac{1}{2m}\sum_{ij}(A_{ij} - \frac{k_i k_j}{2m})\delta(c_i, c_j) \tag{3}$$

Where c_i denotes the community to which node i has been assigned; A_{ij} denotes whether there is an edge between nodes i and j $(A_{ij} = 1)$ or not $(A_{ij} = 0)$; $k_i = \sum_j A_{ij}$ denotes the degree of node i, and m $= 1/2\sum_{ij} A_{ij}$ denotes the total number of edges in the network. The function $\delta(c_i, c_j)$ indicates whether nodes i and j belong to the same community, which equals 1 if $c_i = c_j$ and 0 otherwise.

The gain in modularity ΔQ obtained by moving an isolated node i into a community C can be easily computed by

$$\Delta Q = [\frac{\Sigma_{in} + k_{i,in}}{2m} - (\frac{\Sigma_{tot} + k_i}{2m})^2] - [\frac{\Sigma_{in}}{2m} - (\frac{\Sigma_{tot}}{2m})^2 - (\frac{k_i}{2m})^2] \tag{4}$$

Where $\sum_{in}$ is the sum of the weights of the links inside C, $\sum_{tot}$ is the sum of the weights of the links incident to nodes in C, and $k_{i,in}$ is the sum of the weights of the links from i to nodes in C.

2.3 Algorithm Description

The main steps of the algorithm named LPR(Local PageRank) are shown in Algorithm 1:

Input: $G = (V, E)$: The edges of the network
Output: c: Final assignment of nodes to communities;

// assign the initial communities to nodes
$c \leftarrow Initial(G)$

//compute the PageRank of each node

$pageRankValue \leftarrow PageRank(G)$
//sort the nodes in reversed order by pageRankValue
$nodesInOrder \leftarrow Sort(pageRankValue, G)$

// Run the local moving heuristic.
$c \leftarrow LocalMove(G, nodesInOrder)$

// Construct a new network.
$G_{new} \leftarrow GetNewNetwork(G, c)$

if $Modularity(G) < Modularity(G_new)$ **then**
  $c_{new} \leftarrow LR(G_{new})$
end

Algorithm 1. The main steps of the LPR algorithm

Our algorithm consists of three phases that are repeated iteratively. Assume that we start with a weighted network of *N* nodes. Initially, we assign a different community to each node of the network, so each node in a network is assigned to its own singleton community.

Firstly, we calculate the PageRank value in which assigning initial PageRank value of each node with one, and recalculating each value according to the Eq. 6 until each PageRank value does not change any more. The PageRank value is not only used as the measurement to evaluate the importance of each node, but also used to determine the order in which nodes are chosen in the second phase.

In the second phase, we take out the node *i* in the reversed order according to the PageRank value calculated in first phase. Then, considering the neighbours *j* of node *i*, we evaluate the gain of modularity that would take place by removing *i* from its current community and by placing it in the community of *j*. If the maximal gain is positive, the node *i* is then placed in the community for which this gain, otherwise, node *i* stays in its original community. This process is applied repeatedly and sequentially for all nodes until a local maxima of the modularity is attained and the phase is then complete.

In the last phase of the algorithm, we rebuild a new network whose nodes are now the communities found during the second phase. To do so, the weights of the links between the new nodes are given by the sum of the weight of the links between nodes in the corresponding two communities. Links between nodes of the same community lead to self-loops for this community in the new network. After the last phase is completed, it is then possible to reapply the first two phases of the algorithm to the resulting weighted network and to iterate.

3 Experiments and Results

In this section, we study the performance of our LPR algorithm in contrast to the lovian algorithm and the fastgreedy algorithm. To quantitatively evaluate our algorithms, we take the modularity Q as the measurement and compare the computational time. Empirically, higher values of the Q function have been shown to correlate well with better graph clusterings [18]. In addition, we apply the LPR algorithm to the karate club network.

3.1 Data Set

We have selected ten small and medium-sized networks and three large networks commonly used in community detection, originating from a number of different domains. Although the real system is more complex, most directed networks can be transformed to undirected networks. Therefore all networks considered are undirected, shown in Tables 1 and 2.

Table 1. Number of nodes and edges of ten small and medium-sized networks.

Network	Nodes	Edges
Karate club [21]	34	78
Les Miserables [11]	77	254
Football [7]	115	613
Jazz [8]	198	2,742
Ego-Facebook [15]	4,039	88,234
Ca-GrQc [13]	5,242	14,496
PGP [2]	10,680	24316
Ca-AstroPh [2]	18,772	198,110
Condmat2003 [16]	27,519	116,181
Email-enron [14]	36,692	183,831

Table 2. Number of nodes and edges of three large networks

Network	Nodes	Edges
com-DBLP [20]	317,080	1,049,866
com-amazon [20]	334,863	925,872
com-LiveJournal [20]	3,997,962	34,681,189

3.2 Result Analyses

Result of Community Kernel Detection. In this section, we adopt the PageRank algorithm independently which is the first step in our algorithm to detect the community kernel in the karate club network. We show the original karate club network in Fig. 1

and the PageRank values in Fig. 2. We could get some information from Fig. 2 that the PageRank values of node 0 and node 33 are the greatest and obviously much higher than other's which is consistent with the fact that at some point, a conflict between the club president (indicated by node 33) and the instructor (indicated by node 0) led to the fission of the club into two separate groups, supporting the instructor and the president respectively.

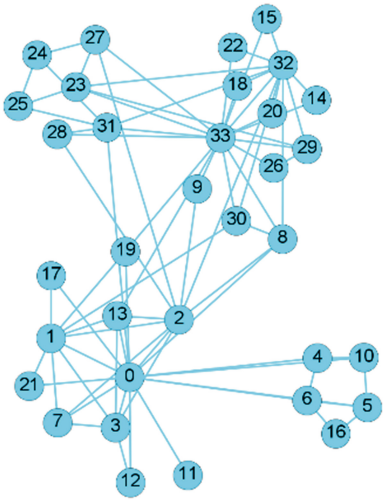

Fig. 1. The original network of karate club

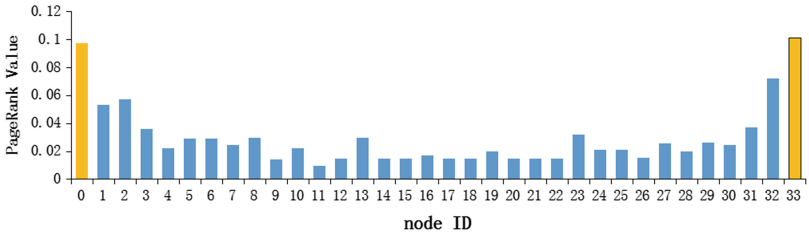

Fig. 2. The PageRank values of karate club

Quantitative Performance. We can get modularity results when applying all algorithms to each network, shown in the Tables 3 and 4. It indicates that the modularity of our algorithm is always higher than the others two in all network data source. Especially comparing with the fastgreedy algorithm, our algorithm is obvious higher, but our algorithm gets slightly higher modularity when compared to lovian algorithm. However, as we can see from the Table 5, our algorithm has an advantage in computational time when applying in large networks, and especially in the com-LiveJournal network, our algorithm gets a 25 % reduction in comparison with the lovian algorithm.

Table 3. Results for 10 small and medium-sized networks.

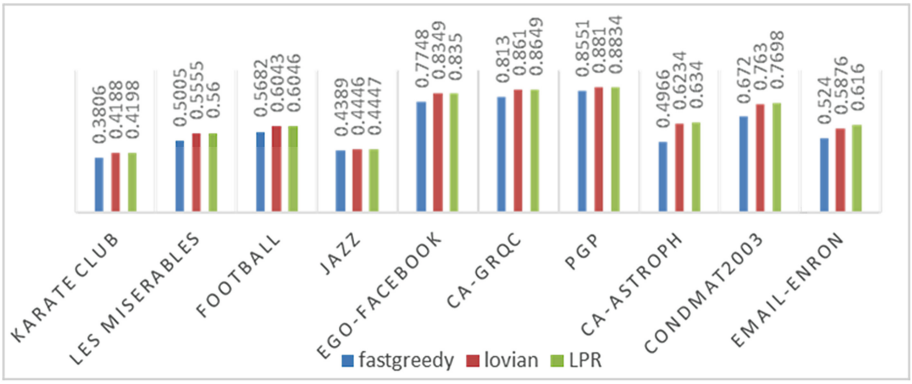

Table 4. Results for 3 large networks. For the com-LiveJournal network, the result of fastgreedy is not available because the computational complexity of those algorithm is too high to get a result in reasonabletime.

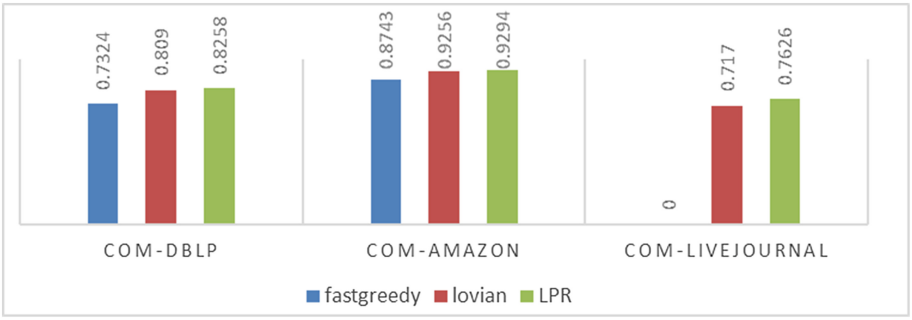

Table 5. The time each algorithm takes in three large networks.

Network	LPR	fastgreedy	lovian
com-DBLP	20.33(s)	4896.62(s)	20.85(s)
com-amazon	10.66(s)	1545.24(s)	11.73(s)
com-LiveJournal	4870(s)	–	6520(s)

Application Case Study. From our experiment, we know that the modularity of the karate club network is 0.4198, which is the best compared to the other algorithms'. We show the original karate club network in Fig. 1 and the division results in Fig. 3. The division result of our algorithm is the same as the famous lovian algorithm's that splits the network into four parts.

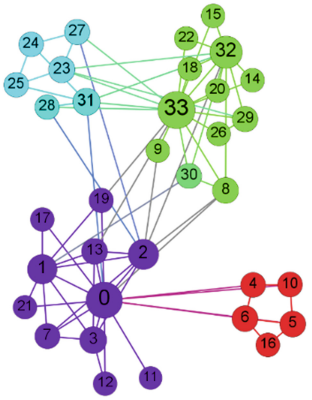

Fig. 3. The application of our algorithm to the karate club network

4 Conclusion

In this paper, we have introduced the LPR algorithm for modularity-based community detection. Our algorithm is intended primarily for community detection in large networks, and is combined with the PageRank algorithm to evaluate the importance of the nodes. Compared with five other algorithms, our algorithm gets a better result in the modularity and performs well in the division result of karate club network.

In future work, we would like to investigate the effect of other methods of community detection using the seed set expansion. Also, it is very interesting to detect community using the distributed computation method when dealing with super large networks.

Acknowledgement. This work is partially supported by the National Natural Science Foundation of China (Nos. 11161140319, 91120001, 61271426), the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant Nos. XDA06030100, XDA06030500), the National 863 Program (No. 2012AA012503) and the CAS Priority Deployment Project (No. KGZD-EW-103-2).

References

1. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech. Theor. Exp. **2008**, 10008 (2008). http://iopscience.iop.org/11742-15468/12008/10010/P10008
2. Boguñá, M., Pastor-Satorras, R., Díaz-Guilera, A., Arenas, A.: Models of social networks based on social distance attachment. Phys. Rev. E **70**, 056122 (2004). http://journals.aps.org/pre/abstract/056110.051103/PhysRevE.056170.056122
3. Chen, J., Yuan, B.: Detecting functional modules in the yeast protein–protein interaction network. Bioinformatics **22**, 2283–2290 (2006). http://bioinformatics.oxfordjournals.org/content/2222/2218/2283.short

4. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Phys. Rev. E **70**, 066111 (2004). http://journals.aps.org/pre/abstract/066110.061103/PhysRevE.066170.066111
5. Flake, G.W., Lawrence, S., Giles, C.L., Coetzee, F.M.: Self-organization and identification of web communities. Computer **35**, 66–70 (2002). http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=989932
6. Freeman, L.: The Development Of Social Network Analysis: A Study in the Sociology of Science. Empirical Press, Vancouver (2004). http://www.researchgate.net/profile/Linton_Freeman/publication/239228599_The_Development_of_Social_Network_Analysis/links/54415c650cf2e6f0c0f616a8.pdf
7. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proc. Nat. Acad. Sci. **99**(12), 7821–7826 (2002). http://www.pnas.org/content/7899/7812/7821.short
8. Gleiser, P.M., Danon, L.: Community structure in jazz. Advances in complex systems **6**(4), 565–573 (2003). http://www.worldscientific.com/doi/abs/510.1142/S0219525903001067
9. Guimera, R., Amaral, L.A.N.: Functional cartography of complex metabolic networks. Nature **433**, 895–900 (2005). http://www.nature.com/articles/nature03288
10. Haveliwala, T.: Efficient computation of PageRank (1999)
11. Knuth, D.E.: The Stanford GraphBase: a platform for combinatorial computing. Addison-Wesley, Reading (1993). http://tex.loria.fr/sgb/abstract.pdf
12. Krawczyk, M.J.: Differential equations as a tool for community identification. Phys. Rev. E **77**, 065701 (2008). http://journals.aps.org/pre/abstract/065710.061103/PhysRevE.065777.065701
13. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graph evolution: densification and shrinking diameters. ACM Trans. Knowl. Disc. Data (TKDD) **1**(1), 2 (2007). http://dl.acm.org/citation.cfm?id=1217301
14. Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Community structure in large networks: natural cluster sizes and the absence of large well-defined clusters. Internet Math. **6**(1), 29–123 (2009). http://www.tandfonline.com/doi/abs/110.1080/15427951.15422009.10129177
15. Leskovec, J., McAuley, J.J.: Learning to discover social circles in ego networks. Advances in neural information processing systems **25**, 539–547 (2012). http://papers.nips.cc/paper/4532-learning-to-discover-social-circles-in-ego-networks
16. Newman, M.E.J.: The structure of scientific collaboration networks. Proc. Natl. Acad. Sci. **98**(2), 404–409 (2001). http://www.pnas.org/content/498/402/404.short
17. Rives, A.W., Galitski, T.: Modular organization of cellular networks. Proc. Nat. Acad. Sci. **100**(3), 1128–1133 (2003). http://www.pnas.org/content/1100/1123/1128.short
18. White, S., Smyth, P.: A Spectral Clustering Approach To Finding Communities in Graph. SDM. SIAM **5**, 76–84 (2005). http://epubs.siam.org/doi/abs/10.1137/1131.9781611972757.9781611972725
19. Williams, R.J., Martinez, N.D.: Simple rules yield complex food webs. Nature **404**, 180–183 (2000). http://www.nature.com/articles/35004572
20. Yang, J., Leskovec, J.: Defining and evaluating network communities based on ground-truth. In: Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics, p. 3. ACM (2012). http://dl.acm.org/citation.cfm?id=2350193
21. Zachary, W.W.: An information flow model for conflict and fission in small groups. J. Anthropol. Res. **33**(4), 452–473 (1977)