

A Comparative Study on Community Detection Methods in Complex Networks

Zhongying Zhao ^{a,b,*}, Shaoqiang Zheng ^a, Chao Li ^{a,**}, Jinqing Sun ^a, Liang Chang ^c, Francisco Chiclana ^d

^a *College of Computer Science and Engineering, Shandong Province Key laboratory of Wisdom Mine Information Technology, Shandong University of Science and Technology, Qingdao, China*

^b *Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China*

^c *Guangxi Key Lab of Trusted Software, Guilin University of Electronic Technology, Guilin, China*

^d *Center for Computational Intelligence, Faculty of Technology, De Montfort University, Leicester, UK*

Abstract. Community detection aims to discover cohesive groups in which people connect with each other closely in social networks. A variety of methods have been proposed to detect communities in social networks. However, there is still few work to make a comparative study on those methods. In this paper, we first introduce and compare several representative methods on community detection. Then we implement those methods with python and make a comparative analysis on different real world social networking data sets. The experimental results have shown that GN algorithm is suitable for small networks, while LPA algorithm has a better scalability. FU algorithm is of the best stability. This work could help researchers to understand the ideas of community detection methods better and select appropriate method on demand more easily.

Keywords: social network mining, community detection, comparative analysis

1. Introduction

The popularity of internet-based social media has made it possible for people to communicate with each other at any time and anywhere freely. People can create content, share information, comment interesting issues and build connections with any person that they like. Thus, the social media has accumulated large number of data about users' interacting behaviors and their social relationships, which implies big value. Social networking analysis and mining has received a great deal of attention and become a popular research area[1,2,3].

Community detection aims to identify cohesive clusters or groups in real-world graphs such as social networks, web graphs and biological networks [4,5]. It is a problem of considerable practical interest and has received a great deal of attention [6,7,8,9]. Community is a very important structure in social networks. The quantitative definition of community has been widely discussed by scholars from different areas. Until now, there is still no well-accepted quantitative definition. In most cases, the community is often dependent on the specific application scenario. For

* Corresponding author. Zhongying Zhao, College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China. Tel.: +86 532 86057524; Fax: +86 532 86057758; E-mail: zysu@163.com

** Corresponding author. E-mail: 1008lichao@163.com

* Corresponding author. Zhongying Zhao, College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China. Tel.: +86 532 86057524; Fax: +86 532 86057758; E-mail: zysu@163.com

** Corresponding author. E-mail: 1008lichao@163.com

example, the community in a social network represents a group of people with similar characteristics; the community in biological networks means the module of biological tissues with similar functions; the community in a web network is considered to be a cluster of web document with related topics. Qualitatively speaking, community is typically considered as a group of nodes with interactions amongst its members than between its members and the remainder of the network [10].

Although there is still no wide-accepted quantitative definitions for the community, numerous techniques have been developed to detect communities effectively and efficiently. Girvan and Newman proposed a method which adopted "betweenness centrality" to find community boundaries [11]. Then, they present "modularity" to evaluate the overall quality of a network partition [12]. It inspires researchers to design an objective function to detect communities hidden in networks. Since the objective is often NP-hard to optimize exactly [13,14], one employs heuristics [15] or approximation algorithms [16] to find sets of nodes which could be considered as communities. Once extracted, such communities are usually interpreted as organizational units in social networks, functional units in biological networks, ecological niches in food web networks, or scientific disciplines in citation and collaboration networks [17,18]. According to the technical principles that they used, the existing representative methods could be divided into five categories: Hierarchical clustering method [11,17,19], Modularity optimization method [12,20,21], Label propagation method [22,23,24], Information flow method [25] and Matrix decomposition method [26,27,28].

It is important to note that different methods often tend to perform particularly well or particularly poorly on certain kinds of graphs. Moreover, one might prefer to choosing specific method with respect to the target application or the properties of the network that to be analyzed. Thus, it is necessary to analyze the characteristics including the advantages and disadvantages clearly. To achieve this goal, in this paper, we conduct a comparative study on those proposed community detecting methods. The main contributions are summarized as follows.

- We make a brief description on the representative methods of community detection.
- These methods are compared and analyzed in terms of the technical principles, time complexity and networking size.
- We implement the above methods and visualized the detected communities with python on different real-world networking data. The running time and the performances on modularity are also compared and discussed.

The results of this work will help researchers understand the ideas of the community detection methods deeply and select appropriate method on demand easily.

2. The Comparative Analysis on Representative Methods on Community Detection

In this section, we analyze six representative methods belonging to hierarchical clustering, modularity optimization, label propagation, information flow and matrix decomposition. The algorithms include GN algorithm, CNM algorithm, FU algorithm, LPA algorithm and Feature Vector algorithm. Those methods are also compared and analyzed from three aspects.

2.1. Representative Methods on Community Detection

(1) GN algorithm

GN algorithm[11] is based on a hierarchical agglomeration of division. The process of divisive clustering is top-down. Initially, all nodes are regarded as a community. According to a certain condition or standard, social networks are divided into different parts. Ultimately, the structure of community is shaped. The basic idea of GN algorithm is using the metric of edge betweenness to make judgments on edge removal. Edge betweenness of an edge is quantified as the number of shortest paths between pairs of nodes that run along it. According to the above definition, if a network contains communities or groups that are only loosely connected by a few intergroup edges, then all shortest paths between different communities must go along one of these few edges. Thus, the edges connecting communities will have a higher value on the matrix of edge betweenness. GN algorithm judges and removes the edges of community mainly based on this metric. The basic steps are as follows. Firstly, the betweenness of each edge is calculated in the network. Then the edge with the highest edge betweenness is selected and removed. After

that, the edge betweenness is recalculated. The above two steps are looped until there is no edge remaining in the network. Eventually, the whole network is organized as a hierarchical tree. In order to assess the quality of division of community, Newman et al. proposed a well-known metric named modularity [12] to evaluate the quality of partitioning to community.

In GN algorithm, every removal of an edge requires recalculating edge betweenness of the remaining edges. Therefore, GN algorithm is of very slow speed. That is the reason why GN algorithm can only be applied to small networks.

(2) CNM algorithm

CNM algorithm [20] is based on cohesive agglomeration of modularity. It is a bottom-up clustering process. Initially, each node is treated as a community. According to some metrics, it selects two communities and merges them into a new community in order. The merging stops when there is only one community. Newman proposed a fast network clustering algorithm based on the idea of modularity [21]. It starts with n communities, in which each vertex is considered to be a sole member. During the iteration process, communities are repeatedly merged together by selecting the pairs resulting in the greatest increase in modularity. The progress of the algorithm can be represented as a 'dendrogram', a tree that shows the order of the joins. By cutting through this 'dendrogram' at different levels, we can get a division of the network into larger or smaller numbers of communities. With GN algorithm, we can select the best cut by looking for the maximal value of modularity. CNM algorithm solves the problem of GN algorithm which has a slow processing speed and results in a limitation on small-scale networks.

Clauset and Newman present the basic idea of this fast clustering algorithm and make changes on calculation of value of modularity. This algorithm only calculates the value of modularity, where connection exists between two nodes. But it gives up the situation that there is no connection between two nodes. With this modification, CNM algorithm improves the efficiency of fast clustering algorithm.

(3) FU algorithm

Fast Unfolding (FU) algorithm [22] is a heuristic method that is based on modularity optimization. The basic idea is to assign each node of the network to a different community initially. In this initial partition there are as many communities as the number of nodes. Then we evaluate the gain of modularity of each node and its neighbours when they form community. The node is then placed in the community with the maximum gain. If the value is negative, the node stays in its original community. This process is performed repeatedly until no further improvement can be achieved.

FU algorithm combines the local optimization and multi-level clustering methods together. Compared with CNM algorithm, FU algorithm tries to merge the neighboring nodes to calculate the modularity. The time complexity of FU algorithm is smaller than CNM algorithm. Meanwhile FU algorithm has a high quality of community clustering and suits for large-scale social network.

(4) LPA algorithm

The LPA algorithm [23] is a localized community detection algorithm based on label propagation. The basic idea is as follows. It is assumed that each node is initialized with a unique label to denote the community that it belongs to. Then one node determines its community based on the labels of its neighbors and chooses the community that the maximum number of neighbors belong to. At the end of the algorithm, the nodes with the same label are grouped together as one community. The LPA algorithm uses the network structure to guide its progress and does not optimize any specific measure.

As we will show, the advantage of this algorithm over the other methods is its simplicity and time efficiency. Compared with other community algorithms, the running time of LPA algorithm tends to be nearly linear, which greatly improves the efficiency and reduces the time complexity. A large number of experiments show that 95% of the nodes have converged after 5 iterations. During the updating process of labels, if the number of different labels are the same, LPA algorithm will select the label randomly. This kind of strategy will increase the probability of randomness which results in the instability of this method.

(5) Info Map algorithm

Info Map algorithm [25] is an information theoretic approach to reveal community structure in weighted and directed networks. The maps are used to describe the dynamics across the links and nodes in directed and weighted networks. Moreover, they are also employed to represent the local interactions among communities, which induces a system level flow of information. The InfoMap algorithm is designed by understanding the flow of information on

network. It aims to identify communities by optimally compressing a description of information flows on the network. The nodes among which information flows quickly and easily are clustered as a well-connected community.

Info Map algorithm uses the theory of information flow to analyze community network. Thus, the communities it identified are relatively accurate. Info Map algorithm can handle the problem of community detection in large-scale network. But with the increasing of network size, the running time of Info Map algorithm increases dramatically.

(6) Feature Vector algorithm

Feature Vector algorithm [26] aims to detect communities by maximizing the modularity. In this method, the maximization process is written in terms of the modularity matrix, which transfers the optimization task to a spectral problem in linear algebra. The modularity matrix plays a role in community detection similar to that played by the graph Laplacian in graph partitioning calculations. The diagonal elements in the Laplace matrix represent the degree of the nodes. For non-diagonal elements, if there is an edge connecting two nodes, the value of this edge is negative one, otherwise zero. The graph is divided into two sub-graphs according to the second smallest eigenvector in the Laplace matrix. The third smallest feature vector divides the two sub-graphs into four sub-graphs, and so on. Since the Laplace matrix is a sparse matrix, we can adopt numerical solution [29] to accelerate the calculating speed of eigenvector and improve the efficiency of community detection.

With this method, we can derive a series of new and competitive algorithms for identifying communities. For instance, a method for detecting bipartite or k-partite structure in networks, and a new community centrality measure that identifies vertices which play a central role in the communities to which they belong.

2.2. A Comparative Analysis of Community Detection Methods

In this section, the above algorithms are compared and analyzed from three aspects: technical principles, the network size and time complexity. The comparing results are shown in Table 1.

Table 1
Comparison of community detection methods

Aspect	GN	CNM	FU	LPA	Info Map	Feature Vector
Technical principles	Hierarchical clustering	Hierarchical clustering	Modularity optimization	Label propagation	Information Flow	Matrix decomposition
Network size	small	medium	large	large	medium	medium
Time complexity	$O(\frac{m^4}{n^2})$	$O(n \log^2 n)$	$O(n)$	$O(m + n)$	$O(n(m + n))$	$O(m * n)$

It is worth mentioning that the size of the network is considered at three different levels. If the network contains several thousands of vertices or less, then it is considered to be small-scale. If the number of vertices in the network is between 10,000 to 100,000, then the network is considered to be medium-scale. Otherwise, if the network contains more than 100,000 vertices, then it is considered to be large-scale. In Table 1, n denotes the number of vertices in the network, m denotes the number of edges in the network.

3. Implementation and Analysis

3.1. Real-World Networking Data

Four kinds of real-world data sets are used to examine the above different methods. They are zachary's karate club, books about US politics, power grid, and condensed matter collaborations. These data are described as follows:

(1) Zachary's karate club (Karate):

It is a social network of a karate club, which was studied by Wayne W. Zachary for a period of three years from 1970 to 1972. The network captures 34 members of a karate club, documenting 78 pairwise links between members who interacted outside the club.

(2) Books about US politics (Polbooks):

It is a network of books about US politics published during the period of 2004 presidential election and sold by the online bookseller of Amazon. Vertices represent books that sold by Amazon. Edges between books represent frequent co-purchasing of books by the same buyers. The network of political book consists of 105 nodes and 441 edges.

(3) Power grid (Power):

The data represents the topology of the Western States Power Grid of the United States, which refers to the connection between generating equipments. It is an undirected, unweighted network. Nodes represent generators, transformers and substation equipment. Edges represent the relationship of transmission for high voltage between these devices. The network consists of 4941 nodes and 6594 edges.

(4) Condensed matter collaborations (Cond-mat-2005):

It is a network of collaborations between scientists posting preprints on the Condensed Matter E-Print Archive. Nodes represent the researchers in physics area. Edges represent the co-authorship of researchers who have published paper in the area of condensed material. This data set describes the co-authorships of scientists from Jan. 1, 1995 to Mar. 31, 2005. The network consists of 40421 nodes and 175693 edges.

3.2. Experimental Settings

The experiments are conducted on one computer server which is equipped with 4 Quad-Core AMD opteron CPU, 32G memory and 12T disk storage. The CentOS 6.4 is installed as the operating system. Other software environments include python 2.7 and pycharm 4.5.3. In order to overcome the effect of randomness, we implement all these algorithms and run them 50 times. The averaged results are considered to be the final results.

3.3. Evaluation Metric

In this paper, we adopt the metric of modularity to quantitatively evaluate the quality of communities detected with different methods. The metric of modularity is based on the principle that a good community is strongly connected internally and isolated from the rest of the network. Suppose we have a division for the network, then the modularity Q is defined as follows.

$$Q = \frac{1}{2m} \sum_{vw} (A_{vw} - \frac{k_v k_w}{2m}) \sum_i \delta(c_v, i) \delta(c_w, i) \quad (1)$$

where

- v, w are vertices within the network;
- A_{vw} is an element of the adjacency matrix corresponding to the network. And it represents the number of communications occurred between v and w ;
- $m = \frac{1}{2} \sum_{vw} A_{vw}$;
- $k_v = \sum_u A_{vu}$, where u is a vertex;
- c_v denotes the community to which vertex v is assigned;
- i denotes the community labeled with i ;
- $\delta(x, y)$ is 1 if $x = y$; otherwise, it equals to 0.

3.4. Experimental Results and Analysis

(1) Communities visualization

We run the above six algorithms on four real-world data sets. The visualization of the communities are also performed.

This section takes the Polbooks network as an example, showing the resulting communities detected by the above six different methods. Figure 1 and Figure 2 are communities identified with GN algorithm and CNM algorithm, respectively. LPA algorithm divides the nodes into two communities, shown in Figure 4. Info Map algorithm divides nodes into six communities shown in Figure 5. Figure 3 and Figure 6 are the four communities detected by FU algorithm and Feature Vector algorithm respectively.

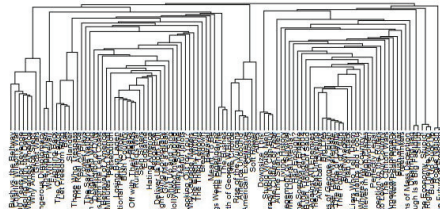


Fig. 1. Communities detected with GN algorithm.

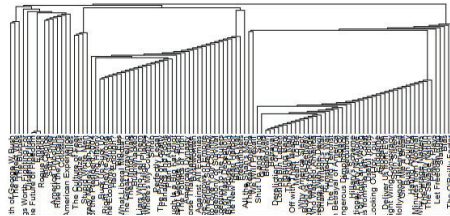


Fig. 2. Communities detected with CNM algorithm.

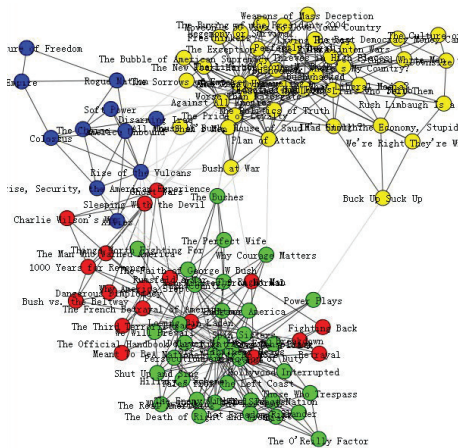


Fig. 3. Communities detected with FU algorithm.

Similarly, we also run the above six algorithms on the other three data sets and obtain the corresponding communities respectively. Due to the large scale of other networks, the visualization of the communities is no longer shown in the paper.

(2) Running time analysis

Table 2 shows the running time of the six algorithms on four networks. It can be seen that the LPA algorithm runs fastest in all algorithms, both in large-scale networks and small-scale networks. From the perspective of running time, GN algorithm performs not very well. It is not capable of dealing with large-scale network. As to Cond-

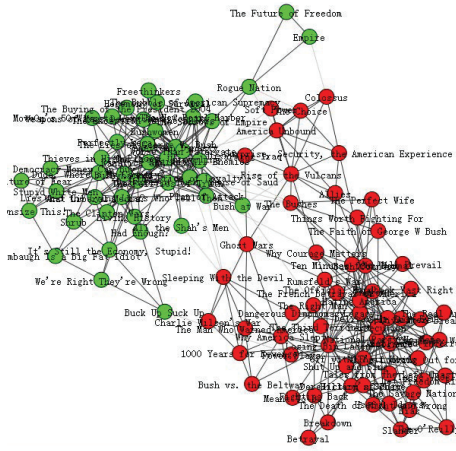


Fig. 4. Communities detected with LPA algorithm.

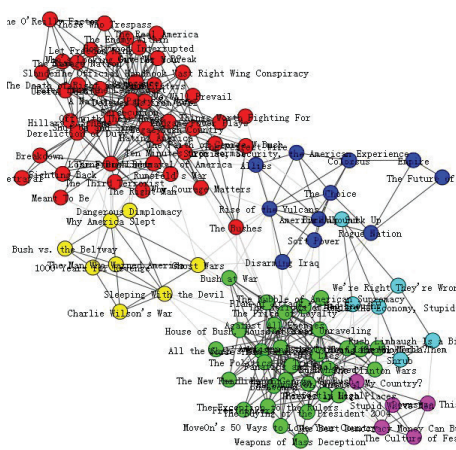


Fig. 5. Communities detected with Info Map algorithm.

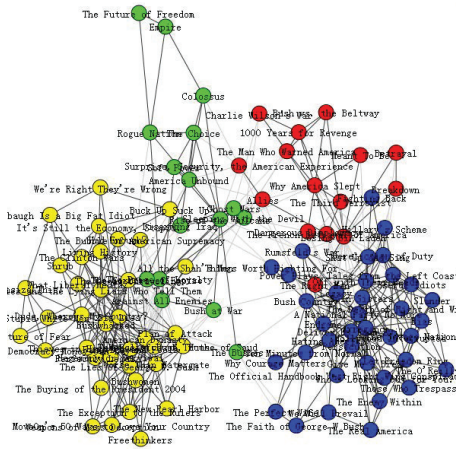


Fig. 6. Communities detected with Feature Vector algorithm.

mat-2005 dataset, it is difficult for GN algorithm to complete the division of community. Info Map algorithm runs faster than GN algorithm on all networking data. CNM algorithm reduces the time complexity, thus it is capable of handling large-scale network. LPA algorithm is of the lowest time complexity but it is not stable on detecting communities because of the random initialization and random selection during the propagating process. It is worth noting that the time complexity of both FU algorithm and LPA algorithm are highly reduced. With the increasing of the size of networks, FU algorithm performs better than LPA algorithm.

Table 2
Runing time of different methods on realworld networking data sets

Realworld data set	GN	CNM	FU	LPA	Info Map	Feature Vector
Karate	0.0718	0.0297	0.0203	0.0058	0.0371	0.1802
Polbooks	2.0837	0.0265	0.0212	0.0172	0.8694	0.2045
Power	1.9×10^3	0.5252	0.3998	0.4996	94.06	63.66
Cond-mat -2005	-	4.7×10^2	12.61	26.60	2.2×10^3	102.86

(3) Modularity evaluating analysis

Table 3 shows the number of communities detected with six algorithms on each network. It can be seen from Table 3 that FU algorithm and CNM algorithm are similar in identifying communities. Table 4 shows the modularity value corresponding to each algorithm on different networking data. According to Table 4, FU algorithm achieves the best performance on the four kinds of networking data. The algorithms of GN, CNM and FU gets the highest value on power grid networking data. On the Polbooks networking data, GN algorithm, FU algorithm and Info Map algorithm get the best performance.

Table 3
The number of communities detected by different methods

Real-world networks	GN	CNM	FU	LPA	Info Map	Feature Vector
Karate	5	3	3	2	3	4
Polbooks	5	4	4	2	6	4
Power	45	41	40	501	491	35
Cond-mat -2005	-	2253	1876	3389	3811	1801

Table 4
The comparison of modularity value from different methods on realworld networking data sets

Realworld data set	GN	CNM	FU	LPA	Info Map	Feature Vector
Karate	0.40	0.38	0.41	0.36	0.40	0.39
Polbooks	0.52	0.50	0.52	0.48	0.52	0.47
Power	0.93	0.93	0.93	0.79	0.82	0.83
Cond-mat -2005	-	0.63	0.72	0.42	0.63	0.36

4. Conclusions

In this paper, we analyze six representative approaches of community detection, and compare them in terms of the principle techniques, time complexity, networking size, modularity and so on. We implement these algorithms

with python on four kinds of real world networking data. The experimental results have shown that each of the six approaches has its own advantages. Although the GN algorithm has a high time complexity, it is capable of detecting communities in small-scale networks precisely. The CNM algorithm which is based on GN algorithm reduces the running time and makes some improvement on scalability. The advantage of LPA algorithm is that it has the lowest time complexity, thus it has a good scalability and could be used in large scale networking data. Info Map algorithm performs well in terms of community quality, but the time complexity increases with the growth of the network size. Feature Vector algorithm is a method based on matrix decomposition. It does not have special advantage on the four real-world networking data sets. As to the FU algorithm, it performs the best in terms of time complexity and community detection.

To summarize, this work could help researchers to understand the ideas of community detection methods better, and help them to select appropriate method in their practical applications. In our future work, we will study an effective and efficient method to detect communities in dynamic evolving networks.

5. Acknowledgment

This research is partly supported by the National Natural Science Foundation of China (Grant No. 61303167, 61433012, 61702306 and 71772107), Ministry of Education in China Foundation for Humanities and Social Sciences (Grant No. 16YJCZH041 and 17YJCZH262), Scientific Research Foundation of Shandong University of Science and Technology for Recruited Talents (Grant No. 2016RCJJ011) and for Innovative Team (Grant No. 2015TDJH102), and Open Project of Guangxi Key Laboratory of Trusted Software (Grant No. KX201535).

References

- [1] Z. Su, Q. Xu, Q. Qi, Big data in mobile social networks: a qoe-oriented framework, *IEEE Network* 30 (1) (2016) 52–57.
- [2] Z. Zhao, C. Li, Y. Zhang, J. Z. Huang, J. Luo, S. Feng, J. Fan, Identifying and analyzing popular phrases multi-dimensionally in social media data, *International Journal of Data Warehousing & Mining* 11 (3) (2015) 98–112.
- [3] Z. Zhao, Y. Zhang, C. Li, L. Ning, J. Fan, A system to manage and mine microblogging data, *Journal of Intelligent & Fuzzy Systems* 33 (1) (2017) 1–11.
- [4] Z. Bu, C. Zhang, Z. Xia, J. Wang, A fast parallel modularity optimization algorithm (fpmqa) for community detection in online social network, *Knowledge-Based Systems* 50 (3) (2013) 246–259.
- [5] Y. Xu, H. Xu, D. Zhang, A novel disjoint community detection algorithm for social networks based on backbone degree and expansion, *Expert Systems with Applications* 42 (21) (2015) 8349–8360.
- [6] Q. Zhan, J. Zhang, P. Yu, J. Xie, Community detection for emerging social networks, *World Wide Web-internet & Web Information Systems* (2017) 1–33.
- [7] S. Fortunato, Community detection in graphs, *Physics Reports* 486 (35) (2010) 75–174.
- [8] Z. Zhao, S. Feng, Q. Wang, J. Z. Huang, G. J. Williams, J. Fan, Topic oriented community detection through social objects and link analysis in social networks, *Knowledge-Based Systems* 26 (2012) 164–173.
- [9] Y. Xu, H. Xu, D. Zhang, Y. Zhang, Finding overlapping community from social networks based on community forest model, *Knowledge-Based Systems* 109 (2016) 238–255.
- [10] D. Liu, D. Jin, D. He, J. Huang, J. Yang, B. Yang, Community mining in complex networks, *Journal of Computer Research & Development* 50 (10) (2013) 2140–2154.
- [11] M. Girvan, M. E. J. Newman, Community structure in social and biological networks, *Proceedings of the National Academy of Sciences of the United States of America* 99 (12) (2001) 7821–7826.
- [12] M. E. Newman, M. Girvan, Finding and evaluating community structure in networks, *Physical Review E Statistical Nonlinear & Soft Matter Physics* 69 (2) (2004) 026113.
- [13] S. Arora, S. Rao, U. Vazirani, Expander flows, geometric embeddings and graph partitioning, in: *Thirty-Sixth ACM Symposium on Theory of Computing*, 2004, pp. 222–231.
- [14] S. E. Schaeffer, Survey: Graph clustering, *Computer Science Review* 1 (1) (2007) 27–64.
- [15] I. S. Dhillon, Y. Guan, B. Kulis, Weighted graph cuts without eigenvectors a multilevel approach, *IEEE Transactions on Pattern Analysis & Machine Intelligence* 29 (29) (2007) 1944–1957.
- [16] R. Andersen, C. Fan, K. Lang, Local graph partitioning using pagerank vectors, in: *IEEE Symposium on Foundations of Computer Science*, 2006, pp. 475–486.
- [17] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, D. Parisi, Defining and identifying communities in networks, *Proceedings of the National Academy of Sciences of the United States of America* 101 (9) (2004) 2658–63.

- [18] J. Leskovec, K. J. Lang, M. W. Mahoney, Empirical comparison of algorithms for network community detection, 2010, pp. 631–640.
- [19] J. R. Tyler, D. M. Wilkinson, B. A. Huberman, *E-Mail as Spectroscopy: Automated Discovery of Community Structure within Organizations*, Springer Netherlands, 2003.
- [20] A. Clauset, M. E. Newman, C. Moore, Finding community structure in very large networks, *Physical Review E Statistical Nonlinear & Soft Matter Physics* 70 (2) (2004) 066111.
- [21] M. E. J. Newman, Fast algorithm for detecting community structure in networks, *Physical Review E Statistical Nonlinear & Soft Matter Physics* 69 (6) (2004) 066133.
- [22] V. D. Blondel, J. L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, *Journal of Statistical Mechanics Theory & Experiment* 2008 (10) (2008) 155–168.
- [23] U. N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, *Physical Review E Statistical Nonlinear & Soft Matter Physics* 76 (3) (2007) 036106.
- [24] M. J. Barber, J. W. Clark, Detecting network communities by propagating labels under constraints, *Physical Review E Statistical Nonlinear & Soft Matter Physics* 80 (2) (2009) 026129.
- [25] M. Rosvall, C. T. Bergstrom, Maps of information flow reveal community structure in complex networks, *Proceedings of the National Academy of Sciences Usa* (2007) 1118–1123.
- [26] M. E. Newman, Finding community structure in networks using the eigenvectors of matrices, *Physical Review E Statistical Nonlinear & Soft Matter Physics* 74 (3) (2006) 036104.
- [27] F. Wang, T. Li, X. Wang, S. Zhu, C. Ding, Community discovery using nonnegative matrix factorization, *Data Mining and Knowledge Discovery* 22 (3) (2011) 493–521.
- [28] J. Yang, J. Leskovec, Overlapping community detection at scale: a nonnegative matrix factorization approach, in: *ACM International Conference on Web Search and Data Mining*, 2013, pp. 587–596.
- [29] C. G. Baker, U. L. Hetmaniuk, R. B. Lehoucq, H. K. Thornquist, Anasazi software for the numerical solution of large-scale eigenvalue problems, *ACM Transactions on Mathematical Software* 36 (3) (2009) 1–23.