A Kind of Community Detecting Algorithm Based on Modularized Label Propagation

Fang Li[1], Wentao Zhao[2], Zhifeng Sun[3], Bin Dong[4]
Department of network engineering
School of Computer, National University of Defense Technology
Changsha, China
e-mail: lengxuelf@163.com[1], wtzhao@nudt.edu.cn[2], zhifengsun@163.com[3], dobi110@qq.com[4]

Abstract—**the discovery of high-quality community is a hot spot in social network analysis and many algorithms have been proposed to discover communities. To find potential community structures, a community discovery algorithm based on label propagation will propagate the label of nodes in social networks, but this method contains uncertainty and randomness, and is very sensitive to the structure of social networks, which causes the final result to be highly unstable and contains a huge number of small and fragile societies. Therefore, a novel called Modular-Label-Propagate-Based algorithm for community discovery is proposed. This algorithm starts with the propagation of the network vertexes, and then binds the nodes with compact structure. Label propagation of the vertexes is executed according to the order of the degree. The thought of greedy method is applied, which works according to the sequence of the neighbor node size of current vertexes, if the module value increases, the sequence will be renewed. Experiments have been conducted on data sets with different characteristics. The experimental results show that modular label propagation algorithm can significantly improve the quality, effect and stability of the found communities, and be close to a linear complexity.**

Keywords-component; label propagation; modularity; complex network; community structure; greedy

I. INTRODUCTION

With the rapid development of Web 2.0 and the rise of online social networks, finding community structures in massive user data has become a hot topic in network analysis. Social network is commonly abstracted into a graph. In which the vertexes stand for users, while edges stand for existent relationship among users. In relevant researches, many researchers focus on social network's internal organization structure and its evolution and network topology structure's influence on these dynamic social system [1-2].

Many complex networks have community structure. Research shows that social networks [3, 4], biochemical networks [5] and so have a clear community structure. Community structure is an important topological property in social network. In general terms, the social network community is made up of a group of similar vertexes [6]: The community of social network consists of a vertex group in a network structure, the internal links between vertexes are denser in the same community, while sparser in different communities. Network structure and network capabilities have close connection. Studying the network community structure can reveal the hidden laws in the complex network and help us to predict and control the behavior. Thus, by discovering and analyzing the community structure of the complex network we can well understand its structure and its behavior, so it has a profound research value and significance.

Therefore, huge efforts have been invested on the definition of social network communities, community detection, community discovery and identification and some other related content. Community detecting algorithm was put forward to spot the structure of such communities. However, the formerly proposed algorithms have been mostly focused on how to divide the network structure to generate communities but did it consider neither the propagation characteristics of social networks and the quality of generated communities, nor the actual demand of large-scale social networks for algorithm complexity. Therefore, there are many scholars begin to focus on how to improve the efficiency of community discovery algorithm and the quality of generated communities.

II. RELATED WORK

Many scholars investigated the issues on how to find the community structures in network from different aspects. To date, a variety of community discovery methods have been proposed, which can be divided into many categories, including methods based on graph theory, optimization-based community discovery methods, methods based on heuristic strategies and some other methods. Optimization-based community discovery methods contain spectral method and module optimization method; Methods based on graph theory, are like the random walk method [7] and sorting method [8], spectroscopy [9-10] and Figure segmentation heuristic [11]; Methods based on heuristics are GN algorithm [12], WH-Hagerman algorithm [13], MFC algorithm [14], HITS algorithm [15], CPM algorithm [16], edge-clustering coefficient and so on. In addition, there are other effective community discovery methods, such as method based on hierarchical clustering methods (type and split cohesion), method based on non-negative matrix factorization and other associations.

The above methods are all focused on discovering community structures according to the division of network

978-1-4799-1984-0 /15/$31.00 ©2015 IEEE

structure while failing to consider its propagation characteristics, and the complexity. Raghavan et al proposed community discovery algorithm based on label propagation [17], the algorithm not only considers the network structure, but also takes the propagation characteristics of social networks into account. Nevertheless, the drawback of the algorithm is the poor stability. Huge differences may even occur in the results of the same computing process. Based on this algorithm, we propose a community detecting algorithm based on modular label propagation (Label Propagation Based on Modularity, LPBM), which not only has nearly-linear time complexity, but also improves the quality of community discovery, avoiding instability of the original algorithm.

III. COMMUNITY DETECTING ALGORITHM BASED ON LABEL PROPAGATION

Zhu et al. put label propagation algorithm [18] forward in 2002, It is a graph-based semi-supervised learning methods. The basic idea of which is to use marked vertex's label information to predict that of the unmarked one. Label data is like a source, which can mark unlabeled data. The higher the similarity of the vertexes is, the easier the label can be transmitted. As the algorithm is simple with good classification effect, low complexity and short functioning period, it has aroused the attention from both domestic and oversea scholars and have been widely applied to fields such as multimedia information classification and virtual community mining etc.

Raghavan et al propose a localized community detection algorithm based on label propagation [17], the algorithm considers not only the network structure, but also the propagation characteristics of social networks. The main idea behind the label propagation algorithm is as follows: Suppose that a node v has n neighbors $v_1, v_2, v_3, ..., v_n$ and that each neighbor carries a label denoting the community to which they belong. Then the node v determines its community based on the labels of its neighbors. We assume that each node in the network chooses to join the community to which the maximum numbers of the labels of its direct neighbors belong, with ties broken uniformly randomly. We initialize every node with unique labels and let the labels propagate through the network. As the labels propagate, densely connected nodes in a group quickly reach a consensus on a unique label (see figure 1).Such dense (consensus) groups are massively created throughout the network, and will continue to expand outwards as much as possible. At the end of the propagation process, nodes having the same labels are grouped together as one community.

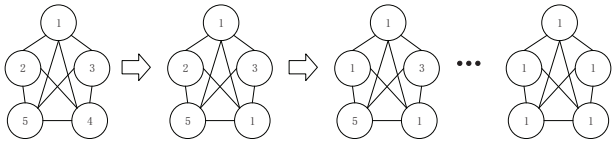

Figure 1 the process of label propagation

Raghavan et al perform this process iteratively, during which at every step, each node updates its label based on the labels of its neighbors. The updating process can be either synchronous or asynchronous. In synchronous updating, node v at the t^{th} iteration updates its label based on the labels of its neighbors at iteration $t-1$. Hence,

$$C_v(t) = f\left(C_{v_1}(t-1), C_{v_2}(t-1), ..., C_{v_n}(t-1)\right)$$

Where $C_v(t)$ is the label of node v at time t. The problem however is that sub graphs in the network that are bi-partite or nearly bipartite in structure lead to oscillations of labels (see figure 2). This is especially true in cases where communities take the form of a star graph.

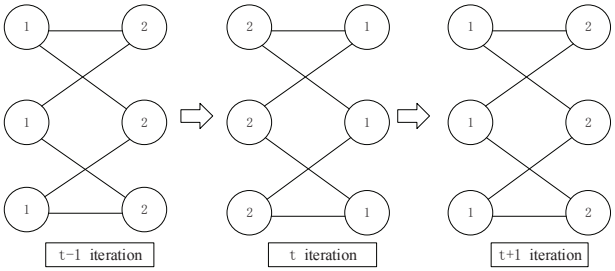

Figure 2 the process of labels oscillations

Therefore, Raghavan et al use asynchronous label updating method, which means in the neighbors of node v, he uses a portion of the labels after the t^{th} iteration and the other portion of the label after the $(t\text{-}1)^{th}$ iteration together to determine the label of node v ,so as to avoiding tag turbulence.

$$C_v(t) = f\left(C_{v_{i1}}(t), ..., C_{v_{im}}(t), C_{v_{i(m+1)}}(t-1), ..., C_{v_{in}}(t-1)\right)$$

Where $v_{i1}, \ldots, v_{im}$ are neighbors of x that the labels of which have already been updated in the current iteration while $v_{i(m+1)}, \ldots, v_{in}$ are neighbors that are not yet updated in the current iteration.

IV. COMMUNITY DETECTING ALGORITHM BASED ON MODULAR LABEL PROPAGATION

In label propagation algorithm, when there are many candidate labels to meet the requirements in the neighbors of one node v, the label of v will be randomly selected. This randomness will greatly reduce the stability of the algorithm. Moreover, for each iteration it will choose to traverse the vertexes in the graph with random order, which will add many random factors to each iteration process. The superposition of excessive randomness will lead to the significant differences among results of several experiments, and it will also increase the hardship of reaching a steady state in this iterative procedure, without which there will be a lack of a precise measurement for the withdrawal mechanism.

Considering these limitations, this paper proposes a new algorithm: LPMB based on modular label propagation.

A. Modularity

Newman and Girvan [19] proposed an indicator evaluating the quality of community, called modularity metric, which has been a very popular quality measurement. The Q value is the sum of differences of the fraction of all links within each community minus the expected value of the same quantity in a network in which nodes have the same degrees but links are placed randomly, which is computed as follows:

$$Q = \sum_{c \in L} \left(\frac{Lc}{m} - \left(\frac{Dc}{2m} \right)^2 \right)$$

Where L is the number of communities; m is the number of links in a network; Lc is the number of the links in the c community, i.e., both ends of which are located in the c community; Dc is the sum of the degrees of nodes in the c community. Such a notation is derived from assessing the differences in the expected values of the link ratios before and after community detection. When the Q value approaches 0, nodes in the network are distributed at random partitions and no obvious community structure exists; when the Q value approaches 1, the network has a strong and obvious community structure.

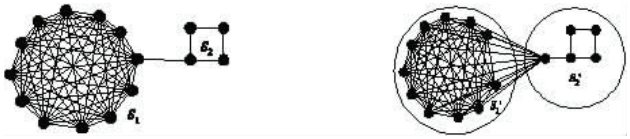

Figure 3 the two community structure of complex networks (The left is a decomposition one, and the right is decomposed two)

In real networks, the value is usually located between 0.3 and 0.7, and for a given network, there will be an actual module value, to which the closer the value obtained, the better the community structure, rather than the bigger the better. As is shown in Figure 3, S_1 is a complete network which contains 100 nodes, S_2 is a cyclic network comprising 4 nodes, obviously decomposition two is extremely unreasonable, But its module value is 0.0017762, more than 0.0016129 of decomposition one

B. The Algorithm Description

Complex network can be abstracted to graph $G = (V, E)$, V represents the collection of vertices in graph G, and E represents the collection of sides.

(1)The first step of label propagation is to distribute a unique label for every vertexes in the net, after that these labels will be continuously updated. When the scale of the network is becoming larger, the resources consumption of these updates is becoming greater. Lots of vertexes in the network can actually be bounded together, if so the initial number of labels can be greatly reduced in the whole, which can improve the efficiency of the updating of the label algorithm. It can be found from the nature of community structure that the internal connection of the community is closely associated, while the external connection is sparser. The vertexes which are closely associated can be bound together, therefore we execute a label propagation on network before the iteration, bind the closely connected nodes together, at the same time mark the vertex and its neighbor vertexes, Is defined as follow:

X is the neighbor vertex set, for each unmarked x belong to X, there is

$$C_{x \in X, \text{visited}[x]=0} = C_v$$

Among them, C_v is the current vertex labels, $visited[x] = 0$ shows that this vertex is unlabeled. Moreover, the pseudo code as follows:

Input: G=(V,E),v

a) $visited[x] = 1$

b) for every $x \in X$ and $visited[x] = 0$ do

c) $C_v = C_x$, $visited[x] = 1$

d) end for

Output: label set

(2) Vertexes in the graph will be traversed with random order in each iteration, which will add numerous random factors towards each iteration process. . Moreover, when there are many candidate labels to meet the requirements in the neighbors of one node, the label of which will be randomly selected. This randomness will greatly reduce the stability of the algorithm. Therefore, before the iteration begins, all of the vertexes in the graph can be sorted according to the degree of $d(i)$ .We hold the view that the higher the degree of the vertex, the more likely that it will have huge influence. Therefore, we update the labels according to the order of the degree of the vertex, and reduce many meaningless label updates. For the specific network structure, this mode will improve the efficiency of the algorithm to a corresponding degree. Before the iteration, all degree of vertexes will be calculated and sorted. In each iteration and multi-candidate-situation, labels will be sorted according to the sequence of the degree of vertexes.

(3) Initially each vertex in the network will be allocated with a unique label. In the process of propagation, the labels will dynamically update which leads to the appearance of a large number of small isolated communities and prevents some real communities from being generated. Those communities with little research significance will cause repeated meaningless judgments in the process of iteration. Since each vertex has a unique label, vertexes with less influence will conversely affect some more influential ones in the process of propagation, which leads to an "upstream" phenomenon of resources consuming.

According to section 3.1's formula, for any communities $Q_c (c \in L)$ it can be defined as follows, where Q_c is the nodular value of the community of c .

$$Q_c = \frac{Lc}{m} - \left(\frac{Dc}{2m}\right)^2$$

Therefore, for the community of L, the total Q value is

$$Q = \sum_{c \in L} Q_c$$

We hold the view that for each Q_c, if its value is big, it indicates that the community c has a better division effect and the value of Q is bigger. According to communities' internal compact characteristics and its' external sparse characteristics, greater degrees of neighbor nodes have illustrated that the current to- be- updated vertex and the neighbor nodes with great degree are more likely to be in the same community. Therefore, when it comes to label update of vertex in LPMB algorithm, we use greedy algorithm. If the current node is isolated, then the update is not performed. If labels exist in adjacent points, then in set X_n formed by all adjacent points v_n, we choose a neighbor node's label as the label of v according to the order of size Calculate Q_c and compare it with the value before the inclusion of vertex v. If Q_c is bigger than the value, then update the label and end the current cycle. If Q_c is smaller than the value, then give up the updating and continue with the next neighbor node. The pseudo code as follows:

Input: $G = (V, E), X_n, v$

a) $visited[v] = 1$

b) for every $v_n \in X_n$ and $visited[v_n] = 1$ do

c) Calculated the Q_c according to the degree of order

d) if(Q_c Increase)

e) $C_v = C_{v_n}$, $visited[v_n] = 1$, continue;

f) end if

g) end for

Output: The label of vertex v

C. Algorithm implementation

According to the above improvement, detailed description of LPBM is as follows:

Step1 First input the adjacency matrix M of the current network.

Step2 Initialize the labels of all vertexes in a network. For a given vertex v,

$C_v(0) = v$。

Step3 Calculate all the vertexes' degree, and put them in descending order.

Step4 According to the order of the vertexes' degree spread the vertex with label propagation once until all vertexes have spread or been spread.

Step5 Set the number of iterations to 1.

Step6 For each $x \in X$, obtain the label according to the order of the vertexes' degree by the following formula

$$C_v(t) = f\left(C_{v_{i1}}(t), ..., C_{v_{im}}(t), C_{v_{i(m+1)}}(t-1), ..., C_{v_{in}}(t-1)\right)$$

During which the label value of the mostly obtained label by the neighbors will be returned.

Step7 For $v_n \in X_n$, Calculate the modularity Q according to the order of the neighbors' vertex degrees.

step8 Compare the value Q of the community structure at time t with that at time $t-1$, if Q increases, then update the labels and end the cycle, otherwise go to step 7

Step9 If the label of iteration of time t remained unchanged from that of time $t-1$, then the algorithm stops. Otherwise, let $t = t+1$, and go to Step6.

Step10 Output the communities.

When the algorithm ends, all the vertexes with the same labels will form a community. Upon the original algorithm of finding communities based on label propagation, at the beginning of iteration we bind all the vertexes to reduce the unnecessary cost of judgments during the process of propagation and the formation of scattered communities, which shows a clear exit mechanism of the iterative propagation process and ensures great stability.

D. Complexity Analysis

Let the network be represented by a simple undirected graph $G=(V,E)$, where N is the set of nodes and E is the set of edges, $n = |N|$ is the number of nodes, and $m = |E|$ is the number of edges.

(1) Initializing every node with unique labels requires O (n) time.

(2) Calculate node importance requires O (m);

(3) Label propagating once requires O (n);

(4) The degree of nodes sorting (counting sort) [20] requires O (n);

(5) Each iteration of the label propagation requires O (m), where l is the number of neighbors.

Finally, in LPBM steps, the time complexity is the same as the LPA of O(km).Hence, the overall time complexity is O(O(n)+l(km))+O(m)+O(km)(k is the number of propagation iterations)[17]. Therefore, LPBM keeps the advantage of high speed of the LPA, with near linear time.

V. Experimental Verification

In this experiment, we select several benchmark data to verify LPBM algorithm and we will compare it with other algorithms to test the effect of LPBM algorithm.

A. Data set

Mainly using four benchmark data set in table 1 to test LPBM algorithm. The Followings are a brief introduction about those five data sets. All of these data sets are from http://www-personal.umich.edu/~mejn/netdata/.

TABLE 1: BENCHMARK DATA SET

Data set				
Name	Description	Vertex	Edge	Community
Zachary	Zachary's karate club [21]	34	78	2
football	American College football union [22]	115	613	12
Dolphins	Lusseau's dolphins [23]	62	159	2
Book	Books about US politics [24]	105	441	3

B. Experimental Results and Analysis

For the four benchmark data set, which already have practical standard results, in order to test the effect of LPBM algorithm, we only need to compare the quality evaluation of LPBM with standard result.

Using the above four benchmark data set respectively to test LPBM algorithm, we can get the results as table 2:

TABLE 2: THE MODULARITY Q AND COMMUNITIES OF BENCHMARK DATA SET

Data set		
Name	Q	Community
Zachary	0.3600	2
football	0.5993	11
Dolphins	0.4858	3
Book	0.4986	3

Compare practical community in table 1 and table 2, we can see that the run results of Zachary and Book US politics by the method of LPBM algorithm are identically agree with practical community. While for the other two data sets, the results are in line with the actual situation.

C. Comparison with other Algorithm

Using four benchmark data sets in table 1, a few famous algorithms are chosen to compare with the LPBM algorithm. We compare two important indexes: the comparative results of the numbers of divided communities and module value can be seen from table 3 and table 4.Because the label propagation algorithm is random; the modularity Q is an average value.

TABLE III. COMPARISON WITH OTHER ALGORITHMS ON MODULE VALUE Q

Data set				
Name	LPA	LPAm+	NFA	LPBM
Zachary	0.355-0.399	0.420	0.380	0.3600
football	0.457-0.476	0.605	0.546	0.5993
Dolphins	0.456-0.492	0.529	0.495	0.4858
Book	0.457-0.476	0.527	0.499	0.4986

Table 3 gives the comparative results of the module value of each algorithm. From table 4, we can see that the module value of community structures obtained by several algorithms have little difference. Module values in each data set are higher than that of LPA algorithm, and lower than that of LPAm+ algorithm, as is shown in Figure 4.

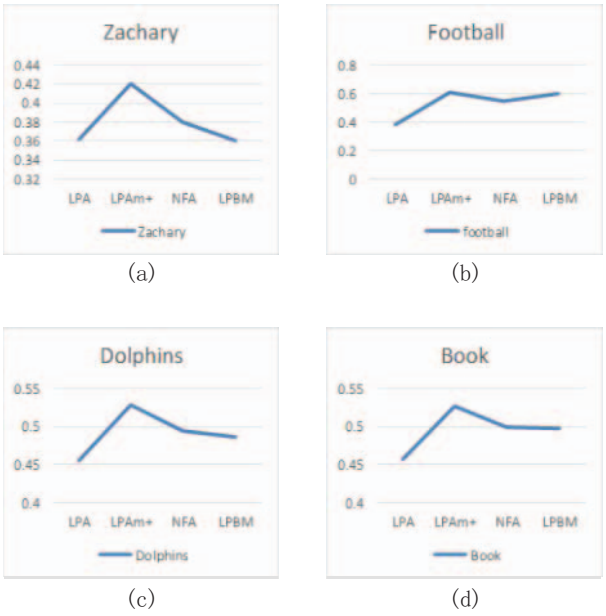

Figure 4. The module value of each algorithm

Table 4 the number of communities generated by each algorithm compares with the number of actual communities.

TABLE IV: COMPARES WITH OTHER ALGORITHM ON Q ,MODULARITY

Data set					
Name	**LPA**	**LPAm+**	**LPBM**	**NFA**	**Reality**
Zachary	4	4	2	3	2
football	8	10	11	7	12
Dolphins	5	5	3	4	2
Book	3	5	3	4	3

The table shows the number of communities obtained by NFA and LPA algorithm is different with the number of actual communities. There are obvious differences in data

sets. For example on the data set Football, the actual number is 12, as is shown in Figure 5.

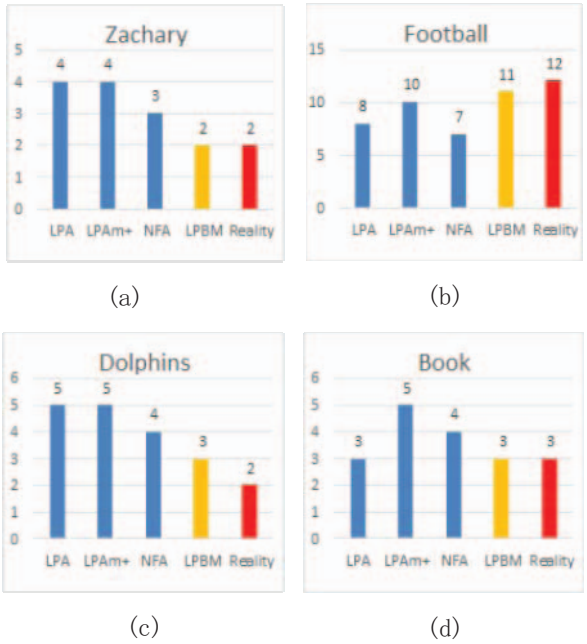

Figure 5 the number of communities of each algorithm

The number of communities LPBM obtained from each data set is almost coinciding with the actual number. It also indirectly shows that the Q value obtained from LPBM algorithm closer to the actual value of the module.

VI. HELPFUL HINTS

LPMB (Label Propagation Based on Modularity) selects one propagation, updates according to vertex degree ranking and uses the method of calculating the impact value of labels to detect community, so it reduces expense of unnecessary judgments in the process of community detection. It also avoids instability of the results of community detection. Therefore, it can get communities of higher quality. Through this experiment, we prove that the experimental result of LPBM algorithm is more consistent with the social network community in reality, and it can get higher -quality community.

Now we propose LPBM algorithms to solve the problem of finding independent disjoint communities. In the future, LPBM algorithms can also be applied to solve the problem of finding the communities with overlapping phenomenon.

ACKNOWLEDGMENT

This work is partially supported by National Science Foundation (NSF) China 61271252.

REFERENCES

[1] M. E. J. Newman, "The structure and function of complex networks.," Society for Industrial and Applied Mathematics Review, vol. 45(2), 2003, pp. 167-256

[2] R. Albert, A. L. Barabási, "Statistical mechanics of complex networks," Reviews of Modern Physics, 7vol. 4(1), 2002, pp. 47-97

[3] Y. P. Zhao, E. Levina, J. Zhu, "Community extraction for social networks," Proc. of the National Academy of Sciences of the United States of America, vol. 108(18), pp. 7321-7326, 2001.

[4] S. Kelley, M. Goldberg, "Magdon-Ismail M, et al. Defining and discovering community in social networks," Handbook of Optimization in Complex Networks, vol. 57(2), 2012, pp. 139-168

[5] M. Girvan, M. E. J. Newman, "Community structure in social and biological networks," The National Academy of Science, vol. 99(12), 2002, pp. 7821-7826

[6] S. M. Angeles, M. Boguna, F. Sagues, "Uncovering the hidden geometry behind metabolic networks," Molecular BioSystems, vol. 8(3), 2012, pp. 843-850

[7] P. Pons, M. Latapy, "Computing communities in large networks using random walks," Proc of the 20th Int Symp on Computer and Information Science. Berlin: Springer, 2005, pp. 284-293

[8] G. Palla, I. Derenyi, I. Farks, et al. "uncovering the overlapping community structure of complex networks in nature and society, " Nature,vol. 435(7043), 2005, pp. 814-818

[9] M .Fiedler, "Algebraic connectivity of graphs," Czechoslovakian Mathematical Journal, vol. 23(98), 1973, pp. 298-305

[10] A. Pothen, H. Simon, K. P. Liou, "Partitioning sparse matrices with eigenvectors of graphs," Society for Industrial and Applied Mathematics Journal on Matrix Analysis and Application, vol. 11(3), 1990, pp. 430-452

[11] B. W. Kernighan, S. Lin, "An efficient heuristic procedure for partitioning graphs," Bell System Technical Journal, vol. 49(2), 1970, pp. 291-307

[12] M. E. J. Newman, "Modularity and communities structure in networks," Proc.of the National Academy of Science, 2006, vol. 103(23), pp. 8577-8582

[13] R. Guimera, L. Amaral, "Functional cartography of complex metabolic networks," Nature, vol. 433(7028), 2005, pp. 895-900

[14] G. W. Flake, S. Lawrence, C. L. Giles, et al. "Self-organization and identification of Web communities," IEEE Computer, vol. 35(3), 2005, pp. 66-71

[15] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment ," Journal of the ACM, vol. 46(5), 1999, pp. 604-632

[16] G. Palla, I. Derenyi, I. Farkas, et al. "uncovering the overlapping community structure of complex networks in nature and society," Nature, vol. 435(7043), 2005, pp. 814-818

[17] U. N. Raghavan, R. Albert, S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," Physical Review E, vol.76(3), 2005, pp. 036-106

[18] ZHU Xiao-jin, Z. GHAHRAMANI, Learning from labeled and unlabeled data with label propagation, CMU-CALD-02-107[R]. Pitts-burghers: Carnegie Mellon University, 2002.

[19] M. E. J. Newman, M. Girvan, "Finding and evaluating community structure in networks," Phys. Rev. E , vol. 69, 2004, 026113.

[20] http://www.nist.gov/dads/HTML/countingsort.html NIST's Dictionary of Algorithms and Data Structures: counting sort

[21] W. W. Zachary, "An information flow model for conflict and fission in small groups," J. Anthropol. Res. Vol. 33, 1997, pp. 452–473.

[22] D. Lusseau, "The emergent properties of a dolphin social network," Proc. R. Soc. Lond. B 270, 2003, pp. S1860–1888.

[23] M. Girvan, M. E. J. Newman, "Community structure in social and biological networks," Proc. Natl. Acad. Sci. USA, vol. 99, 2002, pp. 7821–7826.

[24] V. Krebs, A network of co-purchased books about U.S. politics, 2008 [Online]. Available: http://www.orgnet.com.