International Journal of u- and e- Service, Science and Technology
Vol.8, No.1 (2015), pp.51-60
http://dx.doi.org/10.14257/ijunesst.2015.8.1.05

A Fast and Accurate Algorithm for Detecting Community Structure in Social Networks

Junheng Huang, Yushan Sun, Yang Liu and Bailing Wang*

School of Computer Science and Technology, Harbin Institute of Technology, Weihai 264209, China
wbl@hitwh.edu.cn

Abstract

Studies of community structure and evolution in large social networks require fast and accurate algorithms for community detection. Among the existing algorithms for community detection, the label propagation algorithm (LPA) and the Newman modularity Q algorithm (NMA) have been widely used and studied in the community detection in large social networks, since the LPA has the advantages of near-linear running time, easy implementation and without requiring parameters, and the NMA is a relatively fast algorithm and has a clear metrics to measure community structure. However, the LPA has the shortcomings that the result of the community detection is instable and has a low quality. At the same time, disadvantages of the NMA are that it bases its decisions on purely local information about individual communities and gets the local optimal solution. In this paper, combined with these two algorithms, we propose a new community detection algorithm (LP-NMA), which extends the above two algorithms (the LPA and the NMA is a special case of the new algorithm respectively). The new algorithm not only retains the advantages of these two algorithms, but also has improved the stability and quality of community detection. Experiments on real social networks have proved that this method is better than the original LPA and NMA.

Keywords: *Community Detection, Label Propagation Algorithm, Modularity*

1. Introduction

In the past decades, research of community structure detection in a variety of networks aroused great interest of scholars [1-2]. Community structure detection means that a social network is divided into a number of groups with dense connections within groups and sparse connections between groups. This problem is known to be NP-hard and has been studied as the graph partitioning problem. A variety of heuristic algorithms that can be used to find reasonably good quality communities have been proposed and improved extensively. These methods can be basically divided into the following categories.

1.1. Splitting Algorithm

First, the entire network is regarded as a single community, then, some edges in the network are deleted iteratively based on certain indicators, and finally, the network is divided into a number of connected components and each connected component is a community. The key factor of these methods is what kind of indicators should be adopted to remove the edges and the final stop condition.

Among the splitting algorithms, the most famous algorithm is divisive hierarchical clustering algorithm proposed by Girvan and Newman [3], called GN algorithm. The authors

ISSN: 2005-4246 IJUNESST

define the betweenness of an edge as the number of shortest paths that pass through this edge in the network. The main process of GN is to iteratively remove the edge with the highest betweenness until no edge remains. This algorithm takes advantage of the non-local structure information, so it works well on real-world networks. However, a disadvantage of GN is for a network of n nodes and m edges, the time complexity is O (m^2n) or for a sparse network, the time complexity is O (n^3). Radicchi *et al.* [4] put forward an edge clustering coefficient by considering the number of triangles formed by the edges. Edges that connect vertices of different communities tend to contain few or no triangles, and therefore the edge's clustering coefficient of different communities is relatively small. Its time complexity is O (M^2), where M is the number of edge in the network. Random walk algorithm has been used for discovering community successfully [5-7]. Its idea is that the walk tends to be trapped in the dense regions and the dense regions are likely to be a community. The complexity for Random walk algorithms is O (n^3). Markov clustering algorithm [8] is an unsupervised clustering algorithm based on simulated flow. In a sense, it is a recession random walk algorithm. In addition, there are some multi-state spin models [9-10] and an electric circuit method [11]. The time complexity of these methods are all more than O (n^2).

1.2. Aggregating Algorithms

In contrast with the splitting algorithm, firstly, the aggregating algorithms regard each node of network as a community, and then two smaller communities are merged into a larger community, following some indicators iteratively and until communities meet certain conditions.

The most famous aggregating algorithm is K-means method [12] and its main process is as following. First, randomly selected n (n is the number of communities) nodes as separate communities, and then calculate the distance between each node to each community's center, and incorporate it into its nearest community. Recalculate each community's center and incorporate into the calculation again, until all the nodes are added to the respective community. K-means algorithm is easy and effective, but still there are two significant drawbacks, the first disadvantage is that randomly selected initial centers tend to fall into local minimum. Secondly, in practical applications, the initial value of K is difficult to determine. Newman proposed a faster agglomerative hierarchical clustering algorithm [13], called NMA, which starts with a state in which each node is a single community. Then, it repeatedly merges pairs of communities into one, and at each step, chooses the merger that result in the greatest increase in modularity (termed Q). NMA has time complexity O((m+n)n) or O (n^2) on a sparse network.

The NMA is a relatively fast algorithm and has a clear metrics to measure community structure, with shortcomings that it bases its decisions purely on local information about individual communities and gets the local optimal solution. We will discuss the NMA algorithm detailedly in section 2, since our proposed new algorithm will use it.

1.3. Other Algorithms

Spectral clustering [14] firstly map the network to a space, and then find the community with a fast clustering algorithm. Optimized spectral method was proposed by Newman [15] and its time complexity is O (n^2) on the sparse network. Raghavan *et al.*, [16] proposes a method called label propagation algorithm (LPA) to identify community in large networks. This algorithm has many desirable properties, such as no parameters, easy to implement and run faster in the actual network. It runs linearly in the number of edges, thus linearly also in

the number of nodes for sparse networks. However, it has the shortcomings that the result of the community detection is instability and has a low quality.

In this paper, combined with the LPA and the NMA, we propose a new community detection algorithm named LP-NMA. The LPA and the NMA is a special case of the new algorithm. The new algorithm integrates the network structure and modularity of community; it not only retains the advantages of these two algorithms, but also has improved the stability and quality of community detection. Experiments running on real social networks have proved this method is better than the original LPA and NMA.

2. Related Work

In this section, firstly, the original LPA is reviewed, then, its disadvantages are pointed out using examples, and finally, some new ideas to improve the LPA are presented.

The original LPA is as follows: (1). First, assign a unique label for each node in network. (2). At every step, one node (in asynchronous version) or each node (in a synchronous version) changes its label to the one carried by the largest number of its neighbors. (3). Run step (2) iteratively until the results appear convergence or swing. As a result, nodes with the same label form a community. Significant drawbacks of LPA include the fact that it returns different solutions in different executions and the quality of some solutions is poor. This is because the quality of LPA's solution depends on the local minima that it reaches. In the following, we analyze the shortcomings of the LPA by using examples and propose an improved method.

2.1. Case Study of LPA

Figure 1A shows a 11-nodes network. According to the LPA, initially, the labels of 11 nodes are a, b, c, d, e, f, g, h, i, j, k respectively. Next, in each iteration, it first generates a random sequence of these 11 nodes, then follow the order of the random sequence, each node change its label to the one carried by the largest number of its neighbors. Suppose that initially the generating random sequence is R = (a, g, d, j, b, e, f, i, k, h, f). Figure 1 is a result generated by running the LPA in this order and shows a best result.

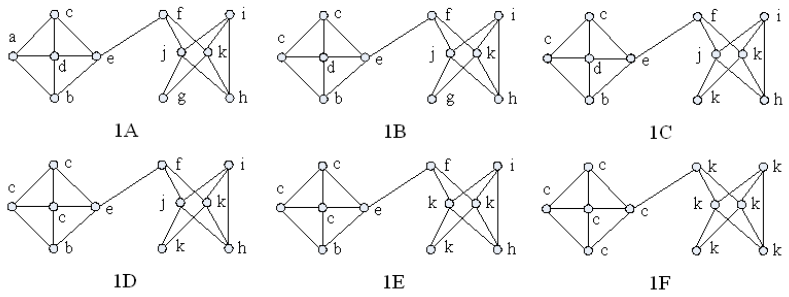

Figure 1. A Solution of the LPA Run in Figure 1A

In Figure 1, Figure 1A illustrates the initial graph, with each node changing their labels in accordance with the order of sequence R; the "a" node's label is changed into "c" in Figure 1B; the "g" node's label is changed into "k" in Figure 1C; the "d" node's label is changed into "c" in Figure 1D; the "j" node's label is changed into "k" in Figure 1E; Figure 1F shows the final result and obviously Figure 1 is divided into two communities (nodes with the same label form a same community).

Due to the randomness of LPA in the order of node's updates and the selection of node's label, the solutions obtained from different runs of the LPA are different. Figure 2 is the solution obtained from a LPA execution and as one can see, a relatively poor result is got.

Suppose initially the randomly generated sequence is R = (e, d, j, k, c, a, b, i, g, h, f), Figure 2 shows a record from running the LPA, and obviously, this result is not what we want.

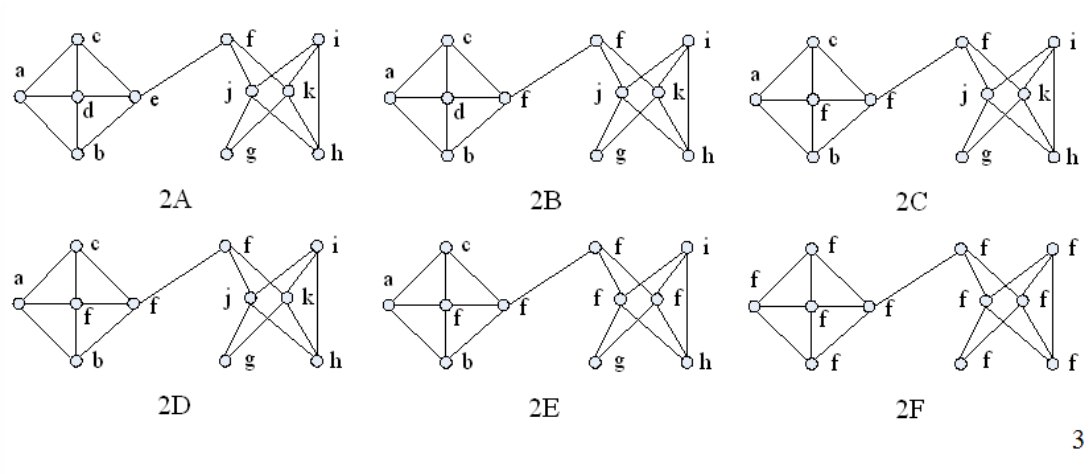

Figure 2. Another One Solution of the LPA Run in Fig1A

In Figure 2, Figure 2A (equal to Figure 1A) is the initial graph; the "e" node's label was changed into "f" in Figure 2B; the "d" node's label is changed into "f" in Figure 2C; the "j" node's label was changed into "f" in Figure 2D; the "k" node's label is changed into "f" in Figure 2E; Figure 2F shows the final result and obviously the whole network has been divided into a single community (note that the network as one single community satisfies the stop criterion). Obviously, this is not an expected result.

To address the nonuniqueness and poor quality of the LPA's solution, according to the fact that those nodes with relatively large degrees are usually the center of a community in network, we propose the following new approach. In the process of the LPA, if a node's degree is relatively large, then its label remains unchanged. Hereinafter, the node that its label remains unchanged is called a fixed node. When a node A's label needs to be updated in LPA, if there is a fixed node B in A's neighbors, then A's label is changed according to B's label. If there are multiple fixed nodes in A's neighbors, then A's label is changed according to the fixed node with the largest degree.

For example, in Figure 1A, the maximum degree is 4 (the nodes are d, j and k). If we choose to fix two node's labels and suppose these two nodes are d and j, then in this case, Figure 3 is the only result that the LPA is executed in Figure 1A and is the best solution.

In Figure 3, the LPA get a correct solution by fixing labels of node d and j. However, in practice, which nodes that their label should be fixed is a thorny issue. We first use the following method: "Suppose V is a set of nodes in a network, $\forall v \in V$, denote degv as node v's degree, and for a threshold δ, when degv $\geq \delta$, then let node v's label remain unchanged." For example, in Figure 1, let $\delta = 4$, so the labels of node d, j, and k remain unchanged, and then run the LPA in Figure 1A, to get the solution expressed in Figure 4. As one can see, originally a community in the right of the figure is mistakenly divided into two.

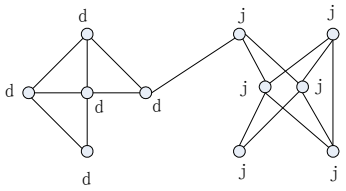

Figure 3. The LPA's Solution by Fixed Two Node's Labels (d and j) in Figure 1A

In summary, the original LPA has the shortcomings that the solutions of the LPA's community detection are instability and poor quality. In order to improve it, the labels for the nodes with relative high degrees should remain unchanged. After doing so, an error has occurred that one community has been divided into multiple small communities, which is not agree with the fact.

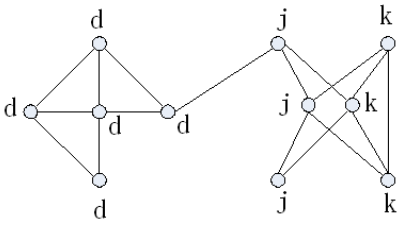

Figure 4. The LPA's Solution by Fixed Three Node's Labels (d, j and k) in Figure 1A

For this case, we find that the reason for a community to be wrongly divided into some smaller communities is that some node's labels are fixed, so that there must be a relatively strong connection between these little communities. The NMA proposed by Newman, which we would introduce in the following, can be used to solve this problem.

2.2. The NM Modularity Clustering Algorithm

In order to determine whether a network partitions is reasonable, Newman et al define a function ("modularity" Q) [2] to calculate the score of a network partition and the higher score indicates better quality of the partition. The modularity is defined as follows, assuming network G is divided into communities G_1, G_2, ...,and G_k, let the number of edges that connect nodes in group G_i to nodes in group G_j are N_{ij}, the total number of edges in the network G is m. Let $e_{ij}=N_{ij}/2m$. where $e_{ii}=N_{ii}/m$ and N_{ii} is the number of edges that two nodes all in group i.

It can be seen from the above definition, $\sum_i e_{ii}$ is a proportion of the number of edges that two nodes located in same group to the total number of edges in the network. If two nodes of every edge are in same group, then $\sum_i e_{ii}$ gets a maximum 1. A good community partition is that a network is divided into groups with dense connections within groups and sparse connections between groups. Clearly a reasonable network division should have a high value of $\sum_i e_{ii}$. The $\sum_i e_{ii}$ on its own, however, is not a good indicator of the quality of the division, e.g., placing all vertices into a single community would give the maximal value of $\sum_i e_{ii}=1$ while giving no information about community structure at all.

Let $a_i = \sum_j e_{ij}$ represent the proportion of edges that connect to vertices in community i to the

number of total edges. If the ends of edges are connected together randomly, the proportion of the resulted edges that connect vertices within group i is a_i^2, so the modularity can be defined as

$$Q = \sum_i (e_{ii} - a_i^2) \quad (1)$$

If a particular division gives no more in-community edges than would be expected by the random situation, this modularity is Q=0, and so values other than 0 indicate deviations from randomness, and in practice values greater than about 0.3 appear to indicate significant community structure.

The increment in Q upon joining two communities c_i and c_j is given by

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

If the reason that a community is divided into some communities is because some node's labels are fixed in LP, there must be a relatively strong connection between these little communities. Merging two these smaller communities will inevitably lead to increases of Q value. So merging community iteratively according to the increment of Q value can overcome this defect (*e.g.* Figure 4).

3. The LP-NMA

Let G = (V, E) represent a social network, V = {v_1, v_2, ..., v_n} be the set of nodes G, E be the set of edges(edges are undirected). The degree of a node v is denoted by deg (v). Let δ be a threshold of positive integer. Node that set V is divided into two subsets by threshold δ, two subsets are $V_1=\{v_i | \deg(v_i)\geq\delta\}$, $V_2=\{v_i | \deg(v_i)<\delta\}$ and $V=V_1\cup V_2$, $V_1\cap V_2=\varnothing$. Node v's neighbor set is denoted as $N(v)= \{w | (v,w)\in E\}$. Denote t times node v_i's label as $L_{vi}(t)$, and let function $f(L_{N(vi)}(t-1))$ represent the largest number of vi's neighbors in t-1 times, LP-NMA is as follows:

(1) Initially, assign a unique identifier as a community tag to each node i in network and each node is an independent community. Let $L_{vi}(0)=i$.

(2) Let t=1.

(3) A random sequence W that contains all the nodes in V was generated.

(4) $\forall v_i\in W$, if $v_i\in V_1$, $L_{vi}(t)=L_{vi}(t-1)$, else $L_{vi}(t)= f(L_{N(vi)}(t-1))$。

(5) If every node's label is the maximum number of its neighbor's labels, then the algorithm is stopped, else, set t = t + 1 and go to (3).

At the end of iteration 1 through 5, nodes have the same label belong to the same rough community. Assuming that this rough community set is $C=\{c_1, c_2, ...,c_h\}$.

(6) let $(c_a, c_b) = \text{argmax}\{\Delta Q(c_i, c_j) | c_i, c_j\in C\}$, if $\Delta Q(c_a, c_b) > 0$, then merge these two communities c_a, c_b in set C.

(7) Repeat step (6) until the condition is not satisfied.

(8) Finally, the set C is the final result.

4. Evaluation of Performance

4.1. Time Complexity

In LP-NMA, initially, all node's degree need to be calculated, which takes a worst-case time of O(m). All nodes in the network are divided into two sets (the set V_1 of fixed node's label and the set V_2 that node's label is changeable), which takes time of O(n). Steps (1) to (5) in the LP-NMA take time O (m)[16]. Steps (6) to (8) take a worst-case time of O((m+n)n)[13]. Since all nodes in the network have been divided into rough groups after steps (1) through (5), the number of times that steps (6) to (8) executed is greatly reduced. Let

d denotes the number of times that steps (6) through (8) are executed. Experiments show that although the d values are not the same in different networks, d values are all far less than n. So steps (6) to (8) approximately take time of O(m+n). The time complexity of the LP-NMA is about O(m) and consistent with the original time complexity of the LPA.

4.2. Tests on Real-World Social Networks

To test the performance of the LP-NMA, we run the three algorithms (LPA, NMA, LP-NMA) on a variety of real network. In the experiment, firstly, all the nodes in network are sorted by their degree, then a threshold c is used to determine the proportion of fixed nodes. Different values of c are explored. Specifically, we analyze the quality of community detection for values of c taken from a set {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1}. Here, the LP-NMA is the initial LPA when c = 0 and the LP-NMA is the NMA when c = 1. Obviously, the LPA and the NMA were two special cases of the LP-NMA.

a. Zachary's karate club network: Zachary's karate club network is a network with 34 nodes and 78 edges [17]. In real life, the club is split into two factions due to fee issues and each member joins one of the two factions. Different solutions are obtained when the LP-NMA run in zachary's karate club network with c = 0, and three different solutions are given in the following.

{ { 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 17, 18, 20, 22, 25, 26, 28, 29, 32 }, {9, 10, 15,16, 19, 21, 23, 24, 27, 30, 31, 33, 34} },

{ { 1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, 14, 17, 18, 20, 22, }, {9, 10, 15,16, 19, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34} },

{ {5, 6, 7, 11, 17}, {1, 2, 3, 4, 8, 12, 13, 14,18, 20, 22, 25, 26, 28, 29, 32}, {9, 10, 15, 16, 19, 21, 23, 24, 27, 30, 31, 33, 34} }。

Solutions of the LP-NMA are more and more stable as c increases. Fig 5 represents a solution by the LP-NMA run steps (1) to (5) in zachary's karate club with c = 0.2 and here the nodes which labels are fixed form a set V1={1, 2, 3, 32, 33, 34}. The final solution that the LP-NMA run steps (6) to (8) in fig 5 is consistent with fact. The solutions obtained with c = 0.3, 0.4, 0.5, 0.6 and c = 0.2 are same respectively. In contrast, the solutions obtained with c = 0.7, 0.8, 0.9, 1.0 are the same and equal to the solution of the NMA. Classification of 10th node in the NMA is inconsistent with the actual situation [13].

b. U.S. college football network: U.S. college football network is an actual network established by Girvan and Newman et al. and consists of 115 college teams represented as nodes and has 616 edges between teams that played each other during the regular season in the year 2000 [18]. The teams are divided into 12 conferences (communities) and each team plays more games within its own conference than inter conference games. In this network, the LP-NMA was respectively executed 100 times for values of c taken from the set {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1} (NOTE: the LP-NMA with c = 0 is equal to the LPA and the LP-NMA with c = 1 is equal to the NMA). The LP-NMA runs 100 times with different c values respectively. For c=0, 48 different solutions are got; and for c = 0.1, 36 different solutions are got; for c = 0.2, 19 different solutions are got, and obtain a unique solution for c ≥ 0.3. Fig 6 shows the number of solutions that the LP-NMA runs 100 times for each c that takes different values.

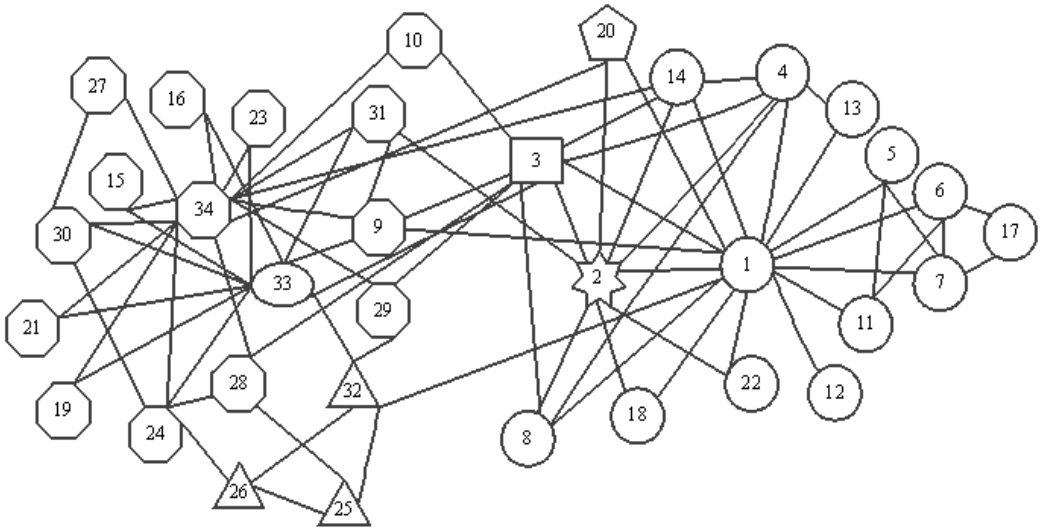

Figure 5. The Solution by the LP-NMA run (1) ~ (5) Steps in Zachary's Karate Club with c=0.2

According to the literature [18], College Football Network has 12 communities. The NMA divides the College Football Network into 6 communities. The LP-NMA divides the College Football Network into 9 communities with c = 0.3, 0.4, 0.5 and 7 communities with c = 0.6, 0.7. That is to say, in comparison of the two algorithms solutions, the LP-NMA is superior to the NMA. The reason why the LP-NMA has such advantages is it combines the idea of the LPA and therefore increases the global measure of NMA.

In summary, with the increase of the c value, the LP-NMA transforms from the LPA to the NMA, with randomness of the algorithm's solutions being gradually decreased and stability of the algorithm's solutions being gradually increased. Experiments have verified that the LP-NMA can obtain solutions superior to the other two algorithms with c = 0.3 ~ 0.5.

5. Conclusion

In this paper, we presented a new community detection algorithm, which has extended the two widely studied algorithms LPA and NMA. The LPA and NMA become a special case of the new algorithm. In the new algorithm, a parameter c ($0 \leq c \leq 1$) is introduced to relate the LPA and NMA. The new algorithm is the LPA for c = 0 and is the NMA for c = 1. Through assigning c with a reasonable value, the new algorithm can improve stability of the LPA's solutions and quality of the NMA's solutions. Experiments have verified that the LP-NMA can obtain solutions superior to the other two algorithms with c = 0.3 ~ 0.5. However, the selection of c is not yet fully understood, and therefore it is the subject worthy of further study. Extending our approach to online community detection is another direction for future research.

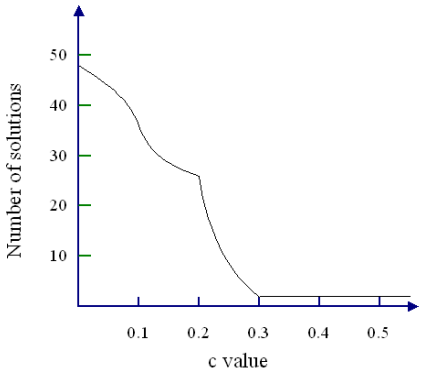

Figure 6. The Nucmber of Solutions that the LP-NMAs Runs 100 Times for c Take of Different Values

Figure 6. The Nucmber of Solutions that the LP-NMAs Runs 100 Times for c Take of Different Values

Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 61371177 and No. 61170262). We would like to thank the anonymous reviewers for their helpful comments.

References

[1] S. H. Strogatz, "Exploring complex networks", Nature (London), vol. 410, **(2001)**, pp. 268-276.
[2] M. E. J. Newman, "The structure and function of complex networks", SIAM Rev., vol. 45, no. 2, **(2003)**, pp. 167-256.
[3] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks", Proc Natl Acad Sci USA, vol. 99, no. 12, **(2002)**, pp. 7821–7826.
[4] F. Radicchi, C. Castellano and F. Cecconi, "Defining and identifying communities in networks", PNAS, vol. 101, **(2004)**, pp. 2658–2663.
[5] H. Zhou and R. Lipowsky, "Network brownian motion: A new method to measure vertex-vertex proximity and to identify communities and subcommunities", Lect. Notes Comput. Sci., vol. 3038, no. 1062-1069, **(2004)**.
[6] Y. Hu, M. Li, P. Zhang, Y. Fan and Z. Di, "Community detection by signaling on complex networks", Phys. Rev. E, vol. 78, no. 1, **(2008)**, p. 16115.
[7] P. Pons and M. Latapy, "Computing communities in large networks using random walks", J. Graph Algorithms Appl., vol. 10, no. 2, **(2006)**, pp. 191–218.
[8] S. van Dongen, "Graph clustering by flow simulation", Ph.D. dissertation, University of Utrecht, **(2000)**.
[9] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," Proc. Natl. Acad. Sci. USA, vol. 104, **(2007)**, pp. 36–41.
[10] J. Kumpula, J. Saramaki, K. Kaski and J. Kertesz, "Limited resolution in complex network community detection with potts model approach", Eur. Phys. J. B, vol. 56, **(2007)**, pp. 41–45.
[11] F. Wu and B. A. Huberman, "Finding communities in linear time: a physics approach", Eur. Phys. J. B, vol. 38, **(2004)**, p. 331.
[12] J. A. Hartigan and M. A. Wong, "A K-Means Clustering Algorithm", Applied Statistics, vol. 28, no. 1, **(1979)**, p. 100-108.
[13] M. E. J. Newman, "Fast algorithm for detecting community structure in networks", Phys. Rev. E, vol. 69, **(2004)**, p. 66-133.
[14] S. White and P. Smyth, "A spectral clustering approach to finding communities in graphs", in Proc. of SIAM International Conference on Data Mining, **(2005)**, pp. 76–84.
[15] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices", Phys. Rev. E, vol. 74, **(2006)**, pp. 36-104.

[16] U. N. Raghavan, R. Albert and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks", Phys. Rev. E, vol. 76, **(2007)**, pp. 36-106.
[17] W. Zachary, "An information flow model for conflict and fission in small groups", Journal of Anthropological Research, vol. 33, **(1977)**, pp. 452–473.
[18] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks", Proc Natl Acad Sci USA, vol. 99, no. 12, **(2002)**, pp. 7821–7826.

Authors

Junheng Huang. He is working for Harbin Institute of Technology (abstract as HIT) as an associate professor. His research is mainly on social network, data mining, artificial intelligence and bioinformatics.

Bailing Wang, He is working for Harbin Institute of Technology (abstract as HIT) as a professor. He got the Ph.D. degree from HIT in 2006. His research is mainly on information security, network security, parallel computing.

Yang Liu, He is associate professor and his research fields include Network information Security Technology, Internet of Things Security Technology, etc. He has participated in many projects of Ministry of Information Industry and National Science, and he has published over 20 academic papers in journals and conferences both home and abroad.

Yushan Sun, He is working for Harbin Institute of Technology (abstract as HIT) as a professor. His research is mainly on software architecture and artificial intelligence.

