HIERARCHICAL COMMUNITY STRUCTURE IN COMPLEX (SOCIAL) NETWORKS*

Emanuele Massaro

Dept. of Information Engineering and CSDC, University of Florence, Italy†

Franco Bagnoli

Dept. of Physics and Astronomy, and CSDC, University of Florence, Italy
and
INFN sezzione Firenze, 50019 Sesto Fiorentino, Italy

(Received March 25, 2014)

The investigation of community structures in networks is a task of great importance in many disciplines, namely physics, sociology, biology and computer science, where systems are often represented as graphs. One of the challenges is to find local communities in a graph from a local viewpoint, in the absence of the access to global information, and to reproduce the subjective hierarchical vision for each vertex. In this paper, we present the improvement of an information dynamics algorithm in which the label propagation of nodes is based on the Markovian flow of information in the network under cognitive-inspired constraints. We introduced two more complex heuristics that allow to detect the hierarchical community structure of the networks from a source vertex or a community, adopting fixed values of model's parameters. Experimental results show that the proposed methods are efficient and well-behaved in both the real-world and synthetic networks.

DOI:10.5506/APhysPolBSupp.7.379
PACS numbers: 07.05.Kf, 89.65.–s, 89.75.Fb

[*] Presented at the Summer Solstice 2013 International Conference on Discrete Models of Complex Systems, Warszawa, Poland, June 27–29, 2013.

[†] Present addresses: Risk and Decision Science Team, US Army Engineer Research and Development Center, 696 Virginia Rd., Concord, MA 01742, USA and Department of Civil and Environmental Engineering, Carnegie Mellon University, 5000 Forbes Av., Pittsburgh, PA 15213, USA.

1. Introduction

We live in a world of networks. The networks around us and ourselves, as people, we are part of the network of social relations between individuals. Examples of networks in the world are the WWW, the rail network, the subway, neural networks, the telephone network, or less concrete entities, such as the relations of knowledge and collaboration between people. In general, a graph or network is a very general approximation of a system constituted by many entities, called nodes (persons, computers, proteins, chemicals, *etc.*) linked to each other and interacting through connections (which may be, therefore, cables among computers, hyperlinks between web pages, collaborations among people, reactions among chemical substances, *etc.*). The study of networks is, therefore, very relevant in many disciplines, given the wide variety of structures and systems of the real world that can be incorporated into the category of "complex networks".

A complex network is a system with non-trivial topological features which would not be detectable in simple graphs. The majority of real networks, being them social, biological or technological, may be considered complex: this depends on some features such as, *e.g.*, the nodes' degree of distribution [1, 2], the assortativity among vertexes [3], relatively short path lengths [4, 5] and, often, an evident hierarchical structure [6]. Another important feature of complex networks is the presence of a community structure [7]. Many real networks are not homogeneous and do not consist of a single block of indistinct nodes, rather they exhibit community structures, *i.e.*, groups of vertexes exhibiting a high densities of internal arches, compared to a relatively lower number of connections with other groups.

In recent years, the possibility of automatically identifying communities within networks has been explored in detail, giving rise to a new field of research called "community detection" [8, 9]. The problem of community detection has attracted the attention of various researchers coming from different disciplines from physics to social sciences. Nowadays, no common definition for community has been agreed upon: it is widely accepted that a community is a group of vertexes more linked than between the group and the rest of the graph.

This is clearly a poor definition, and indeed, on a connected graph, there is not a clear distinction between a community and the rest of the graph. In general, there is a continuum of nested communities whose boundaries are somewhat arbitrary. Moreover, in complex networks, and, in particular, in social networks it is very difficult to give a clear definition of the community concept: this is due to the fact that nodes are often attributed to overlapping communities, since they belong to more than one cluster or module or community. For instance, people usually belong to different communities at the same time, depending on their families, friends, colleagues, *etc.* [10]: so people, making a subjective screening of the network in which they live, have their local vision of communities to which they belong.

In order to have a subjective vision of the surrounding world, it is necessary to develop community detection algorithms based on the local link structure of the network, algorithms that work by getting information from neighbouring sites [11–16].

In this paper, we show a method for detecting communities where each node discovers its representation of the network by "talking" with the neighbours and then elaborating the information trough a cognitive-inspired mechanism [17]. In particular, we introduce more accurate heuristics in order to free the method from the tuning of parameters. The main feature of these two heuristics called respectively *IDA + LTE* and *Double pruning* is to reveal the hierarchical community structure of complex networks from a local viewpoint. We demonstrate that a node can discover the network's multi-levels giving its subjective vision of the world until it wants or can explore the graph without needing any global information about it.

The reminder of the paper is organized as follows. First, in Section 2, the general method [17] is presented. Then, we introduce two more complex heuristics applied to this model in order to reveal the multi-resolution community structure of both synthetic and real networks. In the result, we also compare out method with other well-known community detection algorithm by evaluating the so-called Normalized Mutual Information (NMI) on LFR benchmarks [18]. The paper is closed by final remarks and conclusions.

2. The method

We consider N individuals, labelled from 1 to N. Let us denote by A the adjacency matrix, $A_{ij} = 1$ (0) indicates the presence (absence) of a link from site j to site i. Each individual i is characterized by a state vector S_i, representing his/her knowledge of the outer world. We interpret S as a probability distribution, assuming that $S_i^{(v)}$ is the probability that individual i belongs to the community v. Thus, $S_i^{(v)}$ is normalized on the index v. We shall denote with $S = S(t)$ the state of the all network at time t, with $S_{iv} = S_i^{(v)}$. We shall initialize the system by setting $S_{ij}(0) = \delta_{ij}$, where δ is the Kroneker delta, $\delta_{ij} = 1$ if $i = j$ and zero otherwise.

In other words, at time 0 each node knows only about itself. In this implementation, we assume that each individual spends the same amount of time in communications, so the information delivered depends on the connection degree. Let us consider the node i that communicates with the node j as shown in figure 1. We can assume that people with more connections dedicate less time to each of them. Since the amount of available time is limited, we say that the knowledge of node i about node j depends on the connectivity degree of j, then $S_{i\leftarrow j}$ is a function of the connectivity degree k of node j. Alternatively, we can consider that the information depends on the connectivity of i, in this way, we give much more importance to the *speaking phase*.

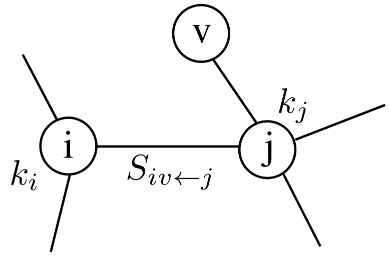


Fig. 1. Communication phase. Let us consider the node i that communicates with the node j. The node j passes his knowledge of the world to the node i. Here, we can consider three different scenarios: the first one takes into account that the knowledge depends on the connection degree of the incoming node. In this way, the most connected nodes have less time to talk with others. In the second hypothesis, the spread information depends of the connectivity degree of the selected nodes: then Eq. (1) becomes $S_{iv\leftarrow j} = mS_{iv} + (1-m)S_{jk}/k_i$. In the third one, it depends both on the degree of node i and on the degree of node j: $S_{iv\leftarrow j} = mS_{iv} + (1-m)S_{jv}/\sqrt{k_i^2 + k_j^2}$.

The dynamics of the information is given by an alternation of communication and elaboration phases. Communication is implemented as a simple diffusion process, with memory m. The memory parameter m allows us to introduce some limitations in human cognitive such as the mechanism of oblivion and the timing effects: in fact, the most recent information has more relevance than the information gathered in the past. The knowledge of a generic node v for the node i trough j occurs via a communication phase where the state of the system evolves as

$$S_{iv\leftarrow j} = mS_{iv} + (1-m)\frac{S_{jv}}{k_j}\,. \tag{1}$$

The representation of the model is consistent with the ecological rationality of the system since the knowledge is distributed among the network elements (state vectors S_i). Our model is an improvement of the MCL algorithm proposed by Van Dongen [19] that simulates a random walk within the graph by an alternation of two phases called expansion and inflation.

In our method, we consider the Markov matrix $M_{i,j} = \frac{A_{ij}}{k_i}$, where k_i is the connectivity degree of node i. Finally, the evolution of the system is given by the sequence $S^0 \to S^1 \to S^2$.

The communication phase based on the connectivity of incoming nodes is given by the following

$$S^1 = mS^0 + (1-m)M'S^0 \,, \tag{2}$$

where $0 < m < 1$ is a memory parameter and M' is the transpose of M. In this way, the resulting matrix is still stochastic, *i.e.*, the matrix elements (on each row) correspond to probability values. The second phase, called elaboration or inflation, is implemented as the rising of each element of the the matrix S_1 to the power α, where $\alpha > 1$ corresponds to the inflation parameter

$$S^2_{ij} = \left(S^1_{ij}\right)^{\alpha} \,. \tag{3}$$

After this phase, in order to have a probability matrix, we normalize the matrix S_2 over the columns. We assume that individuals have a large enough memory so that they can keep track of all information about all other individuals. In a real case, one should limit this memory and apply an input/output filtering.

3. Improvement of the method

In this section, we want to explore the performances of our method taking into account two different heuristics in order to free the method from the precise tuning of the parameters m and α. The main feature of these two heuristics called respectively *IDA + LTE* and *Double pruning* is to reveal the hierarchical community structure of complex networks from a local viewpoint.

3.1. IDA + LTE

Recently, Huang *et al.* [16] proposed a local algorithm for detecting communities in networks. Here, we report a general description of this method, known as Local Tightness Expansion (LTE), while it is well described in Appendix. This method can reveal the community from a starting vertex via local optimization of the tightness measure which is proposed in Ref [20], but the expansion is given by a tunable parameter which is not ecological in social networks where people elaborate the information trough mental scheme or cognitive heuristics. The LTE algorithm can be summarized in the following:

1. Pick a vertex $s \in V$ as the starting vertex.
   Let $C = \{S\}$ and $N = \mathit{\Gamma}(S) - \{S\}$.

2. Select the vertex $a \in N$ that possess the largest similarity with vertices in C.

3. If ${\tau_C}^{\beta}(A) > 0$,
   set $C = C \cup \{a\}$ and $N = N \cup \mathit{\Gamma}(a) - C$.

4. Repeat 2 and 3 until $N = \varnothing$.

The LTE algorithm is very efficient for detecting communities in networks from an individual point of view. When $N = \varnothing$, we arrive at a static convergence. Here, we want to introduce a new method merging together our algorithm, that we call Information Dynamics Algorithm (IDA), and LTE for analysing how the information dynamics can improve the performance of LTE and *vice versa*. We describe the new algorithm in the following (its representation is illustrated in Fig. 2):

1. Pick a vertex $s \in V$.

2. Run LTE and discover C_{LTE} for the vertex s.

3. Run IDA and take information of other nodes by community nodes.

4. Run LTE and decide to accept new nodes.

5. Repeat 3 and 4 until a fixed memory or when the vertex s knows the whole network.

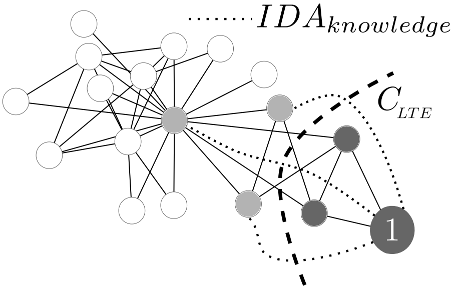


Fig. 2. Schematic representation of IDA + LTE algorithm. We suppose that LTE detect for the node 1 the dark grey nodes as community nodes (C_{LTE}): after that, we compute the IDA on the selected nodes. The node takes the information from those accepted as its community creating a virtual link with these (dotted line in figure). The LTE decides whether to accept the new nodes members of own community or not.

For testing purposes, we use two real networks analyzing and discussing our model peculiarities. The two case studies, of growing or different complexity, are the Zachary's Karate Club network [21] as reported in the Fig. 3

and Fig. 4, and the NCAA College Football network [22] as shown in Fig. 5 and Fig. 6. For all the simulations, we used always the same parameters. For the IDA, we assumed $\alpha = 1.4$ and $m = 0.2$. For the LTE, we used $\beta = 0.3$ for the first step and $\beta = 2$ during LTE+IDA dynamics. As shown in these figures, our algorithms are able to discover the multi-resolution community structure of each node detecting the different local inner circles [23, 24].

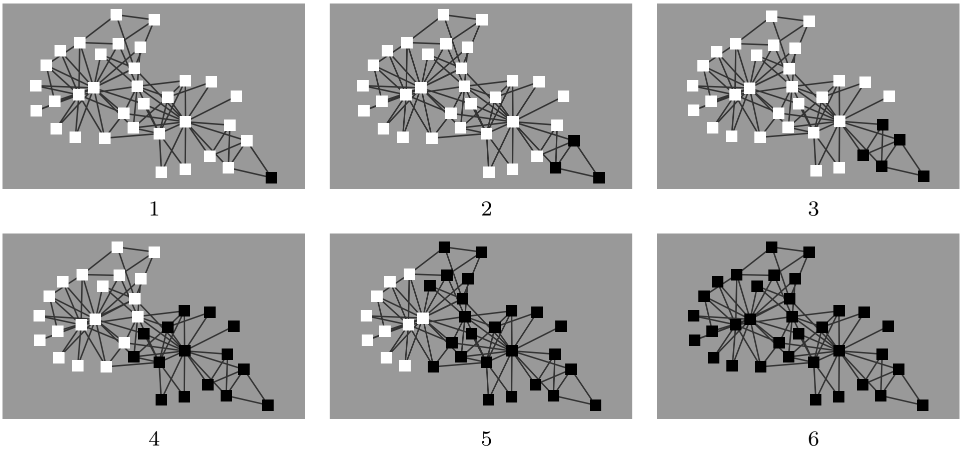


Fig. 3. Node 17 in the Zachary's Karate Club network. In the first step, the LTE algorithm accepts 3 nodes for the local community. After that, we compute the IDA + LTE algorithm: the knowledge of the other nodes permits to discover the structure of the whole network. From the starting vertex (black node at $T = 1$) at time $T = 2$ the community nodes are 17–6–7 (black nodes) and finally, all the network is discovered by the node 17 thanks the information dynamics algorithm.

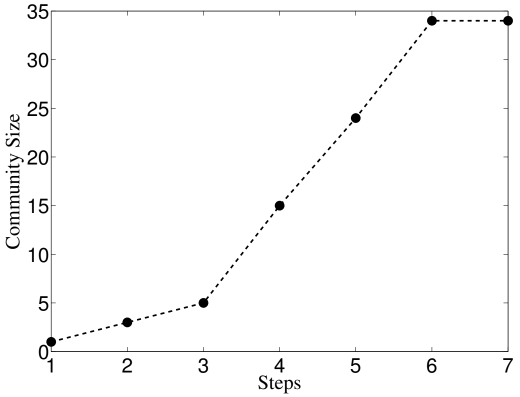


Fig. 4. Temporal evolution (x-axis) of community size (y-axis) for the node 17 in the Zachary's Karate Club network.

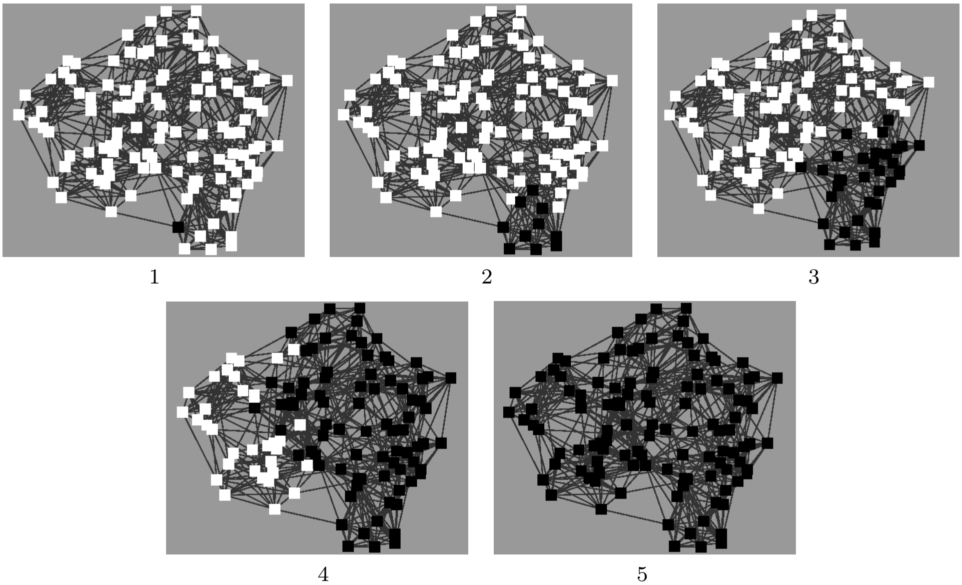


Fig. 5. Washington (black node in 1) in NCAA College Football network. Black nodes indicate the community of Washington for different time steps.

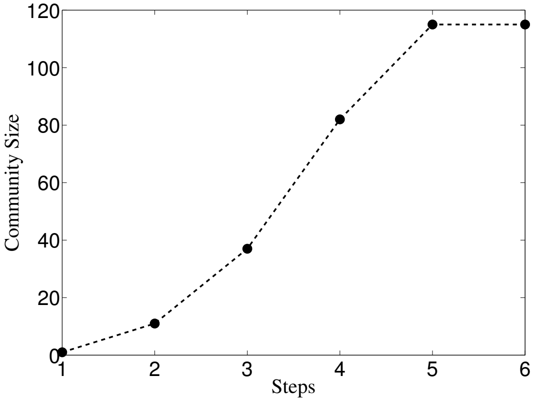


Fig. 6. Temporal evolution (x-axis) of community size (y-axis) for Washington in the NCCA College Football network.

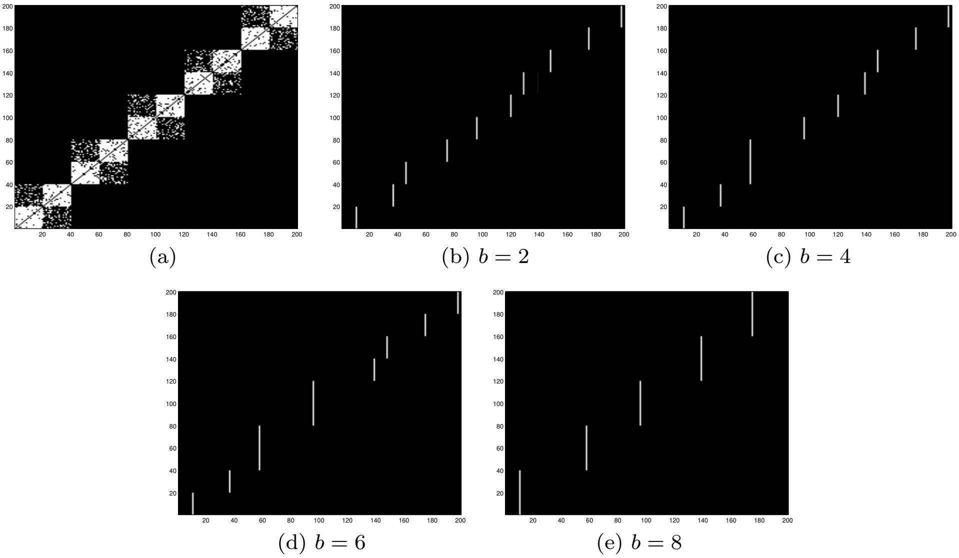
3.2. Double pruning

Our model has two free parameters m and α and in previous works we have shown that it is very difficult to find the specific values of these parameters needed to have a good representation of community distribution for different networks [17, 25, 26]. However, there is a simple procedure that can be used: at some moments for each node we evaluate the histogram of the state vector with 3 bins and we set to zero the values that are lower then the second bin values, as reported in Algorithm 1.

Algorithm 1 pseudocode for double pruning heuristic

```
$T$, $T_{\max}$, $t_b = 0$, $t_1 = 0$, $c = 0$, $a = 0$
$m = 0.2$, $\alpha = 1.4$
while $t < T$ do
    $t = t + 1$; $b = b + 1$;
    if $b < 2$ then
        $t_{\max} = b * 50$
    else
        $t_{\max} = b * 10$
    end if
    for $t_1 = 1 : t_{t\max}$ do
        $a = a + 1$;
        if $a = b$ then
            we evaluate the state vector of each node, then $S^1$ is vector state of node 1, $S^2$ of node 2 and so on...
            for $i = 1 : N$ do
                $dx^i = (S^i_{\max} - S^i_{\min})/3$
                for $j = 1 : N$ do
                    if $S_{i,j} < S^i_{\min} + 2 * dx^i$ then
                        $S_{i,j} = 0$
                    end if
                end for
            end for
            we normalize $S$
        else
            $S_1 = mS_0 + (1 - m)M'S_0$, $S_2 = S_1^{\alpha}$
            we normalize $S$
            a=0
        end if
    end for
    $S = I(N)$
end while
```

In this way, we are able to generate different view of the clustering levels on hierarchical community structure networks as shown in Fig. 7. We use the Lancichinetti–Fortunato–Radicchi (LFR) benchmark graph [18, 27] to evaluate the accuracy of this method. We adopt the normalized mutual information (NMI) to evaluate the quality of detected communities which is currently widely used in measuring the performance of graph clustering algorithms [27]. The accuracy of our method is compared with three other well-known community detection algorithms. The observed results on our hierarchical networks allow us to state that for $b = 4$, we are able to detect the principal communities. For this reason, for comparing our method with others, we set $b = 4$.



(a) (b) $b = 2$ (c) $b = 4$

(d) $b = 6$ (e) $b = 8$

Fig. 7. (a) Adjacency matrix of a network composed by 200 nodes and 3 levels, where $p_1 = 0.9$, $p_2 = 0.2$ and $p_3 = 0$. Trough the double pruning heuristic it is possible to detect the different community levels reaching the discovery of the four principal communities for $b = 8$.

In order to evaluate and partially validate our approach, we have applied our algorithm comparing its performance with 3 community detection algorithms namely *Infomap* [28], *Infomod* [29] and *MCL* [19] as reported in Fig. 8. The input parameters of the benchmark graphs used here are: number of nodes $N = 1000$, average degree is $\hat{k} = 20$, maximum degree $\hat{k} = 50$ for all the networks. Moreover, we changed the range of community size generating two kinds of networks of 1000 nodes: 1000S (S stays for small) means that communities have between 10 and 50 nodes and 1000B (B stays for big) means that communities have between 20 and 100 nodes. Results of the performance's comparison between our algorithm and the others are reported in Fig. 8. Our algorithm with the double pruning heuristic is very competitive with the other algorithms except for the Infomap method which is nowadays the best algorithm for detecting communities in static network, even if it cannot be easily applied in dynamic environments.

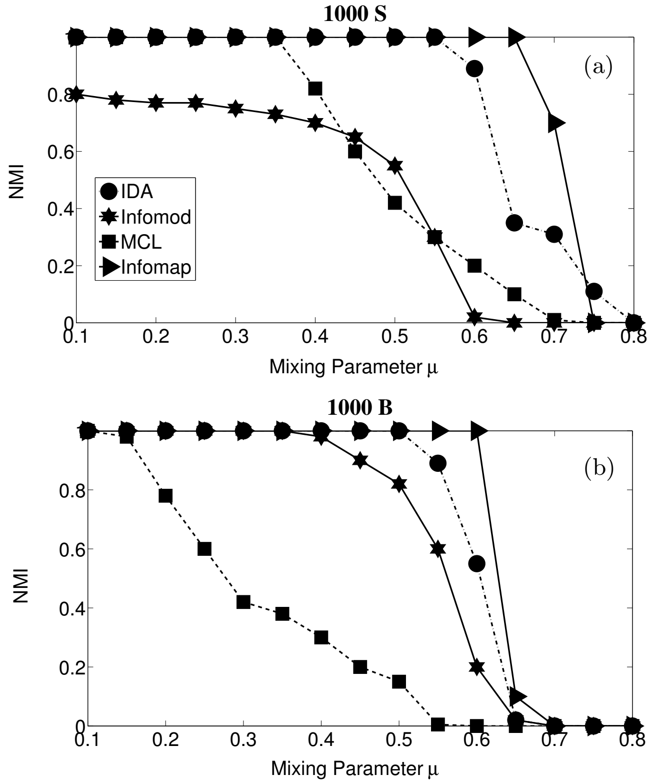


Fig. 8. (a)–(b) Comparison between our method (IDA) and other algorithms (Infomod, MCL and Infomap) on the LFR Benchmark graphs with different size of communities respectively Small (a) and Big (b) increasing the mixing parameter μ.

4. Conclusions

In this paper, we presented two extensions of a community detection algorithm which is based on pure local information propagation inspired by the so-called Markov Clustering Algorithm (MCL) by Van Dongen [19]. The proposed method is able to identify both overlapping and non-overlapping communities, and we have shown that it is capable to identify communities also in dynamic networks [17, 25, 26]. Our algorithm proved to be a multi-level solution that can be used to capture the hierarchical community structure from a local viewpoint at any resolution. Experimental results on the real-world and synthetic datasets show that our algorithm achieves good performance. Concluding, we showed how the adaptation of two heuristics, namely *IDA+LTE* and *Double Pruning* are effective in freeing the method of the precise tuning of parameters.

Appendix

Usually, a network can be represented by a graph $G = (V, E)$, where V is the set of vertices and E is the set of edges.

Definition 1 *(Neighbourhood) Let* $G = (V, E, w)$ *be a weighted undirected network and* $w(e)$ *be the weight of the edge* e*. For a vertex* $u \in V$*, the structure neighbourhood of vertex* u *is the set* $\mathit{\Gamma}(u)$ *containing* u *and its adjacent vertices which are incident with a common edge with* u : $\mathit{\Gamma}(u) = \{v \in V | \{u, v\} \in E\} \cup \{u\}$.

Definition 2 *(Structural Similarity) Given a weighted undirected network* $G = (V, E, w)$*, the structure similarity* $s(u, v)$ *between two adjacent vertices* u *and* v *is*

$$s(u,v) = \frac{\sum\limits_{x \in \mathit{\Gamma}(u) \cap \mathit{\Gamma}(v)} w(u,x) \cdot w(v,x)}{\sqrt{\sum\limits_{x \in \mathit{\Gamma}(u)} w^2(u,x)} \cdot \sqrt{\sum\limits_{x \in \mathit{\Gamma}(v)} w^2(v,x)}} \,. \tag{A.1}$$

When we consider an unweighted graph, the weight $w(u, v)$ of any edge $\{u, v\} \in E$ can be set to 1 and the equation above can be transformed to

$$s(u,v) = \frac{|\mathit{\Gamma}(u) \cap \mathit{\Gamma}(v)|}{\sqrt{|\mathit{\Gamma}(u)| \cdot |\mathit{\Gamma}(v)|}} \,. \tag{A.2}$$

It corresponds to the so-called edge-clustering coefficient introduced by Radicchi *et al.* [20].

Definition 3 *(Tightness) By employing the structural similarity, we introduce tightness, a new quality function of a local community* C*, which is given as follows*

$$T(C) = \frac{S_{in}^C}{S_{in}^C + S_{out}^C} \,, \tag{A.3}$$

where $S_{in}^C = \sum\limits_{u \in C, v \in C, \{u,v\} \in E} s(u,v)$ *is the internal similarity of the community* C *which is equal to two times of the sum of similarities between any two adjacent vertices both inside the community* C*;* $S_{out}^C = \sum\limits_{u \in C, v \in N, \{u,v\} \in E} s(u,v)$ *is the external similarity of the community* C *which is equal to the sum of similarities between vertices inside the community* C *and vertices out of it.*

The tightness measure is extended from the weak community definition proposed in Ref. [20]. Similar to other community definitions, the tightness value of a community C, denoted by $T(C)$, will increase when sub-graph C has high internal similarity and low external similarity. The whole network without outward edges will achieve the maximal value 1, but the problem here is to find the local optimization of the measurement for each community. Suppose a community C is detected from a certain vertex s. We explore the adjacent vertices in the neighbourhood set N of S as shown in Fig. 2. So the variant tightness of the community $C \cup \{A\}$ becomes

$$\begin{aligned} T(C \cup \{A\}) &= \frac{S_{\text{in}}^C + 2S_{\text{in}}^a}{\left(S_{\text{in}}^C + S_{\text{in}}^a\right) + \left(S_{\text{out}}^c - S_{\text{in}}^a + S_{\text{out}}^a\right)} \\ &= \frac{S_{\text{in}}^C + 2S_{\text{in}}^a}{\left(S_{\text{in}}^C + S_{\text{in}}^a + S_{\text{out}}^c + S_{\text{out}}^a\right)}\,, \end{aligned} \tag{A.4}$$

where $S_{\text{in}}^a = \sum_{\{v,a\} \in E \wedge v \in C} s(v,a)$; $S_{\text{out}}^a = \sum_{\{a,u\} \in E \wedge u \notin C} s(a,u)$. Then, the tightness increment of a vertex a joining in C is

$$\begin{aligned} \Delta T_C(A) &= T\left(C \cup \{A\}\right) \\ &= \frac{S_{\text{in}}^C + 2S_{\text{in}}^a}{\left(S_{\text{in}}^C + S_{\text{in}}^a + S_{\text{out}}^c + S_{\text{out}}^a\right)} - \frac{S_{\text{in}}^C}{S_{\text{in}}^C + S_{\text{out}}^C} \\ &= \frac{2S_{\text{in}}^a \cdot S_{\text{out}}^C - S_{\text{in}}^C \cdot S_{\text{out}}^a + S_{\text{in}}^C \cdot S_{\text{in}}^a}{\left(S_{\text{in}}^C + S_{\text{in}}^a + S_{\text{out}}^c + S_{\text{out}}^a\right)\left(S_{\text{in}}^C + S_{\text{out}}^C\right)}\,. \end{aligned} \tag{A.5}$$

If $\Delta T_C(A) > 0$, then $2S_{\text{in}}^a \cdot S_{\text{out}}^C - S_{\text{in}}^C \cdot S_{\text{out}}^a + S_{\text{in}}^C \cdot S_{\text{in}}^a > 0$ which is equivalent to $\frac{S_{\text{out}}^C}{S_{\text{out}}^C} - \frac{S_{\text{out}}^a - S_{\text{in}}^a}{2S_{\text{in}}^a}$. Then they define the tightness gain in the following [16].

Definition 4 *(Tightness Gain) The tightness gain for the community C adopting a neighbour vertex a can be denoted as*

$$\tau_C(A) = \frac{S_{out}^C}{S_{in}^C} - \frac{S_{out}^a - S_{in}^a}{2S_{in}^a}\,. \tag{A.6}$$

It means that the ratio of external similarity to internal similarity of community C is greater than the ratio of external similarity increment to internal similarity increment caused by adopting vertex a. Obviously, this case will result in the increase of the tightness value of community C. Therefore, $\tau_C(a)$ can be utilized as a criterion to determine whether the candidate vertex a should be included in the community C or not. In the following, they introduce an optional resolution parameter β to control the scale at which we want to observe the communities in a network.

Definition 5 *(Tunable Tightness Gain) The tunable tightness gain for the community C merging a neighbour vertex a can be denoted as*

$$\tau_C{}^{\alpha}(A) = \frac{S^C_{out}}{S^C_{in}} - \frac{\alpha S^a_{out} - S^a_{in}}{2S^a_{in}}\,. \tag{A.7}$$

A parameter $\beta \in (0, \infty)$ is introduced as the coefficient of S^a_{out} which can increase or decrease the proportion of the external similarity of the candidate vertex a. Here, the criterion for accepting a vertex a is changed to $\tau_C{}^{\alpha}(A) > 0$. For $\alpha = 1$, the criteria is moderate and can be used in most normal cases. In [16] the authors shows different scenarios for different values of the free parameter β: setting $\beta \in (0, 1)$, the value of S^a_{out} is reduced by this coefficient which increases the chance of a candidate vertex a joining C and bigger communities will be formed compared to the normal case with $\beta = 1$. On the contrary, it will result in the formation of smaller communities in a network when we set $\beta > 1$.

REFERENCES

[1] P. Erdős, A. Rényi, On the Evolution of Random Graphs, Publication of the Mathematical Institute of the Hungarian Academy of Sciences, 1960, pp. 17–61.

[2] A.L. Barabási, R. Albert, *Science* **286**, 509 (1999).

[3] M.E.J. Newman, *Phys. Rev. Lett.* **89**, 208701 (2002).

[4] R. Albert, H. Jeong, A.L. Barabási, *Nature* **401**, 130 (1999).

[5] D.J. Watts, S.H. Strogatz, *Nature* **393**, 440 (1998).

[6] E. Ravasz, A.L. Barabási, *Phys. Rev.* **E67**, 026112 (2003).

[7] M. Girvan, M.E.J. Newman, *PNAS USA* **99**, 7821 (2002).

[8] S. Fortunato, *Phys. Rep.* **486**, 75 (2010).

[9] M.E.J. Newman, *Eur. Phys. J.* **B38**, 321 (2004).

[10] G. Palla, I. Derényi, I. Farkas, T. Vicsek, *Nature* **435**, 814 (2005).

[11] J.P. Bragrow, *J. Stat. Mech.* **2008**, P05001 (2008).

[12] J.P. Bagrow, E.M. Bollt, *Phys. Rev.* **E72**, 046108 (2005).

[13] A. Clauset, *Phys. Rev.* **E72**, 026132 (2005).

[14] F. Luo, J. Wang, E. Promislow, Exploring Local Community Structures in Large Networks, in: Proc. of Web Intelligence, International Conference onWI 2006 IEEE/WIC/ACM, 2006, pp. 233–239.

[15] A. Lancichinetti, S. Fortunato, J. Kertész, *New J. Phys.* **11**, 033015 (2009).

[16] J. Huang *et al.*, *PLoS ONE* **6**, e23829 (2011).

[17] E. Massaro, F. Bagnoli, A. Guazzini, P. Lió, *Commun. Nonlinear Sci. Numer. Simulat.* **17**, 4294 (2012).

[18] A. Lancichinetti, S. Fortunato, F. Radicchi, *Phys. Rev.* **E78**, 046110 (2008).

[19] S.V. Dongen, Graph Clustering by Flow Simulation, Ph.D. Thesis, University of Utrecht, 2000.

[20] F. Radicchi *et al.*, *PNAS USA* **101**, 2658 (2004).

[21] W.W. Zachary, *J. Anthropological Res.* **33**, 452 (1977).

[22] D. Lusseau *et al.*, *Behav. Ecol. Sociobiol.* **54**, 396 (2003).

[23] R. Dunbar, *J. Hum. Evol.* **6**, 469 (1992).

[24] R. Dunbar, *Behav. Brain Sci.* **4**, 681 (1993).

[25] F. Bagnoli, E. Massaro, A. Guazzini, *Lect. Notes Comp. Sci.* **7495**, 204 (2012).

[26] D. Borkmann, A. Guazzini, E. Massaro, S. Rudolph, A Cognitive-Inspired Model for Self-Organizing Networks, in: Proc. of IEEE Sixth International Conference on Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2012, pages 229–234.

[27] A. Lancichinetti, S. Fortunato, *Phys. Rev.* **E80**, 056117 (2009).

[28] M. Rosvall, C.T. Bergstrom, *PNAS USA* **105**, 1118 (2008).

[29] M. Rosvall, C. Bergstrom, *PNAS USA* **104**, 7327 (2007).