

Partitioning networks into communities by message passing

Darong Lai

*MOE-Microsoft Laboratory for Intelligent Computing and Intelligent Systems,
Department of Computer Science and Engineering,
Shanghai Jiao Tong University, 800 Dong Chuan Road, 200240, Shanghai, China*

Christine Nardini

*Key Laboratory of Computational Biology, CAS-MPG Partner Institute for Computational Biology,
Chinese Academy of Sciences, 320 Yue Yang Road, 200031, Shanghai, China*

Hongtao Lu

*MOE-Microsoft Laboratory for Intelligent Computing and Intelligent Systems,
Department of Computer Science and Engineering,
Shanghai Jiao Tong University, 800 Dong Chuan Road, 200240, Shanghai, China*

(Dated: December 13, 2010)

Community structures are found to exist ubiquitously in number of systems conveniently represented as complex networks. Partitioning networks into communities is thus important and crucial to both capture and simplify these systems' complexity. The prevalent and standard approach to meet this goal is related to the maximization of a quality function, *modularity*, which measures the goodness of a partition of a network into communities. However, it has recently been found that modularity maximization suffers from a resolution limit, which prevents its effectiveness and range of applications. Even when neglecting the resolution limit, methods designed for detecting communities in undirected networks cannot always be easily extended, and even less directly applied, to directed networks (for which specifically designed community detection methods are very limited). Furthermore, real-world networks are frequently found to possess hierarchical structure and the problem of revealing such type of structure is far from being addressed. In this paper, we propose a scheme that partitions networks into communities by electing community leaders via message passing between nodes. Using random walk on networks, this scheme derives an effective similarity measure between nodes, which is closely related to community memberships of nodes. Importantly this approach can be applied to a very broad range of networks types. In fact, the successful validation of the proposed scheme on real and synthetic networks shows that this approach can effectively (i) address the problem of resolution limit and (ii) find communities in both directed and undirected networks within a unified framework, including revealing multiple levels of robust community partitions.

PACS numbers: 89.75.Hc, 89.75.Fb

I. INTRODUCTION

Real complex systems can be conveniently modeled as networks, where the systems' elements are identified with nodes and the relations between elements correspond to edges [1–3]. Such a representation is becoming increasingly common and critical to both capture and simplify systems' complexity, notably, via the partitioning of networks into communities. Communities are subgroups of nodes that interact more with nodes in the same groups but far less with nodes in different groups, and they are found to exist ubiquitously in many networked systems. In the World Wide Web, communities are groups of web pages sharing identical or similar topics [4]; in food webs, they correspond to compartments [5]; in biological systems, they are related to functional modules like pathways [6] or proteins having the same function within the cell [7]; in social systems, they correspond to different working or friendship circles [8].

Community finding is usually performed by using only the information encoded in the network topology, and the goodness of a network partition into communities

is usually measured by a widely used quality function called *modularity* [9], proposed by Newman *et al.* Modularity is often computed by referring to a network with the same number of nodes and edges rewired at random but preserving the associated total edges' weight of each node. The partition showing the highest modularity is considered to be the most likely one, consequently, finding communities in networks is often transformed into maximizing modularity, and several heuristics for the optimization of this process have been proposed [6, 10–13] and compared [14]. However, recently, modularity optimization has been shown to suffer from a resolution limit [15]. This indicates that communities found by maximizing modularity are often incompatible with the real community structure of a network, even if communities in that network are cliques. Many efforts have been spread to address such a difficulty. Ruan *et al.* proposed to recursively maximize modularity for each single cluster found by previous partitions, without warranty of finding the proper communities [16]. Li *et al.* proposed *modularity density* as an alternative quality index for modularity [17], however, this approach still suffers from resolu-

tion limit. In fact, although a tunable parameter is used to overcome the resolution limit, it remains difficult to choose a proper value that identifies the most probable partition, in particular if the network does not show a hierarchical (multi-level) community structure. Alternatively, a *spin model* based formulation has also been proposed [18], although this approach was not originally designed to address resolution limit but to generalize modularity in the framework of statistical mechanics. Still, such a generalized quality function (Hamiltonian) suffers from resolution limit [19]. Arenas *et al.* proposed to add to every node a self-loop with identical weight and used modularity optimization with large numbers of different values of self-loop weight to detect dozens of possible candidate partitions [20]. Such a method needs to sample a large range of values of self-loop weights, and it is usually difficult to choose one or more probable partitions from so many possible candidates if *a priori* information is lacking. For larger networks this becomes increasingly difficult since the number of candidate partitions with similar ranges of possible self-loop weights is often increasing quickly [for a fixed self-loop weight the resolution limit remains]. Finally, with the concept of *stability* of a network [21], Lambiotte *et al.* proposed a framework to optimize the modularity of a different network whose edges are weighted by continuous time random walk at different time scales [22]. Such a framework can be used to screen a network at multiple levels of resolution, but it is difficult to identify the most probable partition if the network has only one modular description.

In this paper, we propose a simple but effective unified framework, effectively addressing the problem of resolution limit, to find communities in directed and undirected networks. Similarly to some of the efforts mentioned above, the proposed framework can also be used to screen multiple levels of the structure if the network shows a hierarchical community structure. However, differently from all the aforementioned efforts, the framework can easily identify the most probable partition(s) of a network into communities, since the most likely solutions can be evaluated and ranked with a meaningful statistic.

Just like communities in social systems can be identified by their charismatic leaders, working groups by their managers, labs by their directors, universities or institutes by their principals and so on, similarly, communities in more general types of networks can be assumed to have community leaders, that uniquely identify their members. In this frame, it is possible to imagine that a network is a social organization and that finding communities corresponds to electing group leaders in this organization. The election process can be divided in two phases.

First, every member (node) in the organization needs to evaluate his influence on all his direct neighbors. Let assume for the moment that each member has plenty of *agents* (clarified later) to help him gather enough information on the roles that his direct neighbors play in this

organization. With such information, a member will give high weights to the relations (edges) towards his direct neighbors who will also highly influence him. Each explicit relation (existing edges) will thus be re-evaluated after one round of information gathering. As a direct consequence, the organization level of the system becomes clearer than it was in its original form. The clarification continues as information gathering continues for a certain number of rounds and each explicit relation is thus re-evaluated accordingly. Before entering the second phase, each member should evaluate how well he is suited to be the group leader of other members (with direct or indirect connections) in the whole organization (network).

Secondly, each member takes into account his potential to be a group leader as well as his competitiveness, and sends messages to other members to compete for support from them, in return he receives feedback for his request of support. The qualification of each member to be a group leader is judged by an electoral commission: all members are considered to be candidates and a leader of a group cannot be member of other groups. In such election, the number of underlying groups is not known in advance, however, after the election is over, groups are formed, indexed by their corresponding group leaders.

In this paper, we use *random walk* as the surrogate of agents to gather information. The information acquired is then translated into weights to re-evaluate the network edges. Message passing between nodes is implemented by *Affinity Propagation*, a recently devised efficient clustering method [23], which is also used to define the potential of a node to be a group leader. The electoral commission interprets the constraints on the message passing procedure. It must be pointed out that one of the desirable requisites of a community finding procedure is the automatic determination of the number of communities. Affinity Propagation fulfills this requirement by setting for each node a suitable *preference* that represents how likely it is that such a node will be elected as its community leader. Unlike other clustering methods, like, for example, C-means or C-median [24], the preference setting strategy is much easier to define than the number of communities. In this paper, we suggest a simple preference setting strategy to easily and effectively determine the number of communities.

The direct application of Affinity Propagation to address the community finding problem has been first presented in our previous work [25], where the similarity metric used makes a strong assumption on the edges, i.e. pairs of connected nodes are to some extent similar [26]. Since, for many networks with community structure, edges between pairs of nodes do not always indicate nodes' similarity, the previously suggested procedure is suitable only for a few types of networks. In this current work, we relax such an assumption and adopt a similarity based on community membership of nodes. In practice, the new similarity is the likelihood that a pair of nodes is in the same community. This is based on the fact that

two random walks triggered on two nodes in the same community will have quite similar trajectories and, conversely, relatively different trajectories if they are triggered on nodes in different communities [27, 28]. The trajectory of a random walk triggered on a node can be quantified as an N -dimensional vector in a metric space, where N is the number of nodes. This is thus equivalent to embed each node of a network into an appropriate Euclidean metric space. Consequently, finding communities in networks is transformed into clustering vector points in a metric space. Embedding nodes into a metric space to find communities of a network has been previously suggested elsewhere. In particular, Donetti *et al.* [29] embedded nodes of a network into a metric space by means of eigenvectors of the Laplacian matrix of the network and used hierarchical clustering with suitable distance measures to find communities in that network. Such a method needs to compute the first H eigenvectors corresponding to the first smallest eigenvalues of the Laplacian matrix. The number H needs to be searched in the range $[2, \dots, N]$, which together with the computation of eigenvectors makes the whole process inefficient. Moreover, modularity was used in that framework to select the proper H , carrying therefore the known resolution limit. In order to find communities in directed networks by means of node embedding, we previously proposed to embed nodes of directed networks into the metric space via PageRank random walk induced network embedding [30]. Such a method transforms a directed network into an undirected one via network embedding, therefore modularity is reformulated for directed network which contain undirected ones as a special case. This approach succeeds in several types of real and synthetic networks but it still relies on the above mentioned assumption that pairs of connected nodes are somewhat similar. It is noted that Affinity Propagation in its original form can be regarded as clustering data points in a network, but the edges (and associated weights) of the target network must reflect pair-wise similarity relationships between nodes as well [similar to the case when directly applying another popular clustering method-hierarchical clustering-to partition networks]. In contrast, the proposed scheme incorporating Affinity Propagation is more general, since it does not impose any strong restriction on edges, therefore it can be used to cluster nodes into communities in very general directed and undirected networks.

In the following, we first briefly introduce some basic concepts (*modularity*, *resolution limit* and *random walk*) in Section II, and then propose our community finding scheme in Section III, in particular by introducing the concepts of Affinity Propagation, the similarity based on community affiliation and the message passing procedure. We apply the proposed scheme to a wide variety of real and synthetic directed and undirected networks in Section IV. Conclusion and discussions are finally drawn.

II. MODULARITY, RESOLUTION LIMIT AND RANDOM WALK

A weighted network is a collection of nodes and weighted edges (or links) between pairs of nodes, associated with a weighted adjacency matrix W : $w_{ij} > 0$ if there is an edge with positive weight between nodes i and j , and $w_{ij} = 0$ otherwise. Furthermore, if $w_{ij} = w_{ji}, \forall i, j$, the network is undirected, directed otherwise. A binary network is a special type of weighted network with all non-null weights equal to 1.

To partition a network into communities, an appropriate and meaningful quality index is needed. Modularity is such an index and, given a partition of an undirected network into C groups, it is defined as [9, 11, 31]:

$$Q = \sum_{s=1}^C \frac{w_{ss}}{M} - \left(\frac{w_s}{2M}\right)^2 \quad (1)$$

$$= \frac{1}{2M} \sum_{i,j} (w_{ij} - \frac{w_i w_j}{2M}) \delta(c_i, c_j)$$

where M is the total weight of edges of the network; w_{ss} is the total weight of edges within group s ; w_s and w_i are called weighted degrees of group s and node i , respectively, and they represent the total weight of edges associated with group s and that of edges associated with node i , respectively. The Kronecker function $\delta(\cdot, \cdot)$ ensures that the summation is performed over edges in the same groups and c_i is the group label of node i . For a directed network, the quality of partitioning it into communities is similarly measured by the modularity adapted to directed networks as follows [31]:

$$Q^{Dir} = \frac{1}{M} \sum_{i,j} (w_{ij} - \frac{w_i^{out} w_j^{in}}{M}) \delta(c_i, c_j) \quad (2)$$

where w_i^{out} and w_i^{in} are out-degree and in-degree of node i , respectively. The directed modularity tends to vote for a statistically surprising configuration: if a node i has high out-degree but low in-degree while node j is in the reverse situation, there is more likely a directed edge from node i to node j than vice versa. Finding communities in a directed or undirected network is usually done by maximizing the corresponding modularity.

However, as recalled above, modularity maximization suffers from a resolution limit [15], which sets an intrinsic scale to the communities that can be found. Whether resolution limit will occur or not when maximizing modularity depends on the whole size of a network (the total weight) and on the inter-connectedness between communities in that network. The theoretical analysis of resolution limit was initially performed on undirected binary networks, but has since been naturally extended to the case for weighted networks [27]. Moreover, since there is a relationship between directed and undirected modularity [31], it is expected that maximizing directed modularity also suffers from resolution limit, a fact that has

been empirically validated [28]. Other alternatives effectively addressing resolution limit for community finding are therefore needed, the alternative proposed in this paper is to view community finding as clustering into groups vector points that are efficiently embedded in an appropriate Euclidean metric space.

We implemented such an alternative for community finding by means of random walk on a network. Random walk is a dynamic process that can be simulated on a network to explore its underlying structure and has been successfully applied to reveal community structure in networks [22, 27, 28, 30, 32–34]. At each step, the walker starting from a node of an undirected network randomly selects one neighbor to move to in the next step, with a probability proportional to the weight of the edge connecting the nodes. Formally, a random walk is characterized by the transition matrix P , which elements p_{ij} represent the ratio between the weight of edge (i, j) and the weighted degree of node i , i.e. $p_{ij} = w_{ij}/w_i$, or in matrix form $P = D^{-1}W$ (the degree matrix D is a diagonal matrix: $D = \text{diag}(w_1, \dots, w_N)$). If a network is connected, random walk on it is *irreducible* (it is possible to reach any node from any node) and *aperiodic* (return to any node can occur at irregular time steps), there exists a stationary distribution of the random walk [34] and the probability of random walker being on a node in stationary state only depends on the degree of that node. However, for a given finite time step t , the probability of a random walk starting from any node on a network to reach other nodes in t steps (*random walk length*) is specified in matrix P^t , each row i of which records the expected trajectory of the random walk triggered on node i . To ensure that random walk on a general directed network is irreducible, i.e. a stationary distribution exists, a random walker is converted into a *random surfer* that can teleport with a certain probability instead of walking along the edges on a network [35]. The behavior of a random surfer is characterized by a similar transition matrix P^{Dir} :

$$P^{Dir} = \alpha(D_{out}^+W + \frac{1}{N}\mu e^T) + (1 - \alpha)ee^T \quad (3)$$

where α is the probability that a surfer starting from a node randomly moves along any directed edge to any node that the edge points at, and $1 - \alpha$ is the probability that the surfer randomly and uniformly jumps to any node of the network; D_{out}^+ is the pseudoinverse of the out-degree matrix $D_{out} = \text{diag}(w_1^{out}, \dots, w_N^{out})$ of the directed network; μ is an N -dimensional vector with all zeros except that $\mu_i = 1$ if $w_i^{out} = 0$, and e is another N -dimensional vector with all ones, i.e. $e = (1, \dots, 1)^T$. The stationary distribution is then specified by $\pi = (P^{Dir})^T \pi$, where $\pi = (\pi_1, \dots, \pi_N)$ and $\sum_i \pi_i = 1$. In the following, the value of α is always set to 0.85 as suggested by Page and Brin [36].

III. PARTITIONING NETWORKS BY MESSAGE PASSING

A. Affinity Propagation

Message passing between nodes is here implemented by Affinity Propagation clustering [23]. Affinity Propagation is used to find good partitions of data points (nodes) into groups (communities) and to associate each group with its representative (exemplar) so that the overall similarities between data points and their exemplars is maximized (therefore Affinity Propagation is also known as *exemplar-based* clustering method). In order to explain more vividly the mechanism of Affinity Propagation, nodes (data points in a network) can be visualized as voters, exemplars as group leaders, and the goal of community finding as election. Affinity Propagation takes as input a collection of real-valued similarities between voters, where the similarity $sim(i, k)$ is the likelihood of voters i and k to be in the same community and represents how well a voter k is suited to be the group leader for voter i . Two types of messages -*responsibility* and *availability*- are derived from these similarities and recursively transmitted along the edges of the network to communicate between voters and their group leaders. Responsibility $r(i, k)$, sent from voter i to candidate group leader k , informs k to what extent voter i will support him to be the leader of i by taking into account other potential leaders for voter i . Availability $a(i, k)$, sent from candidate leader k to target voter i , tells i how much evidence there is from other voters that support k to let k be their group leader. At the very beginning, Affinity Propagation simultaneously considers all voters as potential group leaders and thus initializes the availability to zeros, i.e. $a(i, k) = 0, \forall i, k$.

The messages are updated by several simple formulas, based on the maximization of an appropriately chosen function:

$$\begin{aligned} S(c) &= -E(c) + \sum_{i=1}^N \delta_i(i, c_i) \\ &= \sum_{i=1}^N sim(i, c_i) + \sum_{i=1}^N \delta_i(i, c_i) \end{aligned} \quad (4)$$

where $c = (c_1, \dots, c_N)$ are the yet unknown labels of community of nodes; the energy function to be minimized is $E(c) = \sum_{i=1}^N sim(i, c_i)$. Not all label configurations are valid since it is forbidden for voter i to choose k as its group leader without k having been correctly labeled as a leader. A constraint (regulated by the electoral commission) is then added to enforce valid configurations:

$$\delta_i(i, c_i) = \begin{cases} -\infty, & \text{if } c_i \neq i, \text{ but } \exists j : c_j = i, \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

The message updating rules are then deduced:

$$r(i, k) \leftarrow sim(i, k) - \max_{k' \neq k} \{a(i, k') + sim(i, k')\} \quad (6)$$

$$a(i, k) \leftarrow \min\{0, r(k, k) + \sum_{i' \neq i, k} \max\{0, r(i', k)\}\}, \forall i \neq k \quad (7)$$

$$a(k, k) \leftarrow \sum_{i' \neq k} \max\{0, r(i', k)\} \quad (8)$$

Rather than requiring to specify in advance the number of communities, Affinity Propagation takes as input the preference $sim(k, k)$ to indicate how likely node k is to be chosen as community leader. The number of communities is jointly determined by the values of preference and the message passing procedure. The preference $sim(k, k)$ can be regarded as the potential of node k to become a leader, but this also depends on its competitiveness with respect to other candidates.

The responsibility $r(i, k)$ in equation (6) represents the message containing support from voter i to candidate leader k , which is the outcome of competitions between candidate k and other candidates who also persuade voter i to support them. If the competitiveness of node k on acquiring leadership of voter i is even weaker than a voter k' that is no longer a candidate leader (i.e. $a(i, k') < 0$), it is hard for voter i to support candidate k to be its group leader. The self-responsibility $r(k, k)$ is regarded as the self-confidence of node k to be a group leader, based on its potential to be a leader and its competitiveness among all other candidates.

The availability $a(i, k)$ in equation (7) only considers positive support from voters except i since the candidate leader k does not need to show how poor are its scores from voters who tend to not support him (negative responsibility). If candidate k lacks of self-confidence to be a leader ($r(k, k) < 0$) even if it has positive support from other voters, it is better for it to be headed by other candidates and thus availability is thresholded to zero to limit the influence of strong incoming support.

Finally, the self-availability $a(k, k)$ is the total support from voters who are willing to choose node k as their group leader. After the election is over (corresponding to the message-passing procedure convergence), all the nodes can find their corresponding group leaders and the communities are indexed by group emergent leaders. The rule for voter i to identify its group leader is:

$$c_i = \operatorname{argmax}_k \{a(i, k) + r(i, k)\} \quad (9)$$

which means that node i is the group (community) leader if $c_i = i$ or that c_i is the group leader of node i .

B. Community membership based node-node similarity

1. Random walk induced behavioral quantities

To apply Affinity Propagation to find communities in general networks, a suitable similarity metric is needed. Since edges in a network do not necessarily indicate that

the nodes they connect are similar pair-wise, it is necessary to design an appropriate similarity metric to reflect the extent to which nodes are in the same communities. Since it has been observed that random walk on a network will persist for a longer time in the same community than between communities [22, 34], therefore, two random walks triggered on two nodes in the same community will have very similar trajectory patterns, while relatively different if the triggering nodes are in different communities. Such type of similarity has been used and proved efficient in our previous work for iterative edge weighting to enhance the ability of modularity-based community finding algorithms [27] and to reveal communities in directed networks [28].

As mentioned before, P^t records the expected trajectory of a random walk and thus each row of P^t can be regarded as an N -dimensional vector in the Euclidean space. Consequently, P^t is here used as a fundamental variable to find communities in undirected networks [34]. However, there are limitations if only one type of random walk with length being exactly t is considered. These limitations are: sensitive dependence on parts of network far from the target nodes whose pair-wise similarity is being computed, and an unstable measure resulting from fluctuations if the network is nearly bipartite. To reflect the likelihood of pair-wise nodes being in the same communities, a useful similarity measure should emphasize the contributions from nodes near the target nodes currently considered. The underlying principle is that identification of two target nodes in the same community mainly depends on the interconnectedness between target nodes and the nearby nodes (locally interconnected more often within community), not on that between target nodes and far away ones. It is thus more suitable to use t types of random walks with length ranging from 1 to t to derive a stable measure. We used $\sum_{\tau=1}^t P^\tau$ instead of P^t to derive a more appropriate similarity measure to reflect the likelihood of two target nodes being in the same community, as this has been found to perform well in community finding in our previous work [27, 28]. Each row i of $\sum_{\tau=1}^t P^\tau$ can be thought of as the expected number of times that a random walker starting from node i visits all nodes on the network within t steps. For convenience, we denoted these quantities $B = \sum_{\tau=1}^t P^\tau$ and termed them *behavioral quantities* of nodes. B_i is an N -dimensional vector for node i , which can be regarded as an indicator of the influence of an autonomous agent (node i) on nodes it meets on its random walk. As mentioned before, a random walker will visit more often nodes that belong to the same community of the triggering node. The likelihood of pair-wise nodes being in the same community can be computed in terms of the similarity between their corresponding behavioral quantities.

2. Similarity Metrics

To choose a suitable similarity metric, it must be first considered that for most of the networks, the dimension of the behavioral quantity is often very high. Second, what the similarity needs to reflect is the consistence of two behavioral quantities, for instance to what extent they are parallel, not the absolute distance to indicate how far apart they are. We empirically found that three similarity computation strategies are more useful to extract information from behavioral quantity: cosine similarity (sim_{cos}), Pearson correlation (R) and exponential similarity (sim_{exp}).

Consider two behavioral quantities B_i and B_j , the cosine similarity between them is computed as:

$$sim_{cos}(B_i, B_j) = \frac{(B_i, B_j)}{\sqrt{(B_i, B_i)}\sqrt{(B_j, B_j)}} \quad (10)$$

where (B_i, B_j) is the inner product of B_i and B_j .

Pearson correlation between two vectors [37] is also appropriate to measure the consistence of two behavioral quantities:

$$R_{ij} = \frac{\sum_{k=1}^N (B_{ik} - \bar{B}_i)(B_{jk} - \bar{B}_j)}{\sqrt{\sum_{k=1}^N (B_{ik} - \bar{B}_i)^2} \sqrt{\sum_{k=1}^N (B_{jk} - \bar{B}_j)^2}} \quad (11)$$

where B_{ik} is the behavior influence (expected number of times that a random walk triggered on node i visits node k within t steps) of node i on node k and \bar{B}_i is the average behavior influence of node i on all nodes in the network.

In addition, a specially designed exponential similarity is also suitable, which has been proved relatively useful for directed networks [28]. The exponential similarity is:

$$sim_{exp}(B_i, B_j) = exp(2t - \|B_i - B_j\|_{L_1}) - 1 \quad (12)$$

where $\|B_i - B_j\|_{L_1} = \sum_{k=1}^N |B_{ik} - B_{jk}|$. It is easy to show that exponential similarity sim_{exp} is always non-negative and its maximum is $exp(2t) - 1$.

If two behavioral quantities B_i and B_j are highly consistent, i.e. $\|B_i - B_j\|_{L_1} \rightarrow 0$, $sim_{cos}(B_i, B_j)$, R_{ij} and $sim_{exp}(B_i, B_j)$ approach their maximal values: 1 for both cosine similarity and Pearson correlation and $exp(2t) - 1$ for exponential similarity, indicating that nodes i and j are certainly in the same community. In contrast, if B_i and B_j are alternately 0, i.e. they are orthogonal (note that all the entries of behavioral quantity are non-negative), all three measures approach their minimal values: 0 for both cosine and exponential similarity and -1 for Pearson correlation, meaning that nodes i and j are definitely in different communities. By normalizing the values calculated by these similarity measures -namely, $(1 + R)/2$ for Pearson correlation, $sim_{exp}/(exp(2t) - 1)$ for exponential similarity, and already naturally normalized cosine similarity-, the similarities can be used as probabilities to indicate how likely pair-wise nodes are to be in the same community.

The behavioral quantities derived from the original network are sufficient to capture the information needed to induce likelihoods of nodes being in the same communities. Iteratively reweighting the original edges of networks by the newly induced likelihoods will make the behavioral quantities of nodes in the same communities become highly consistent and relatively dissimilar if nodes are in different communities [27]. The community membership based node-node similarity is then derived from the weighted network transformed from the original network by iterative edge reweighting.

C. Community finding by message passing

1. Node preference

In this section we present two alternative ways to define a node's preference as the likelihood of becoming a community leader, one based on the above introduced measures of similarity, and one based on the concept of node degree.

For the first case, let us recall that partitioning a network into communities is equivalent to label nodes of the network with community indexes. If the likelihoods of pair-wise nodes being in the same communities are available, they are used as evidence to indicate how well some nodes are suited to be the group leaders of a set of sub-groups of nodes. With these likelihoods, message passing is executed by Affinity Propagation via message updating rules given in equations (6) to (8). Since Affinity Propagation is very efficient if the input similarities are log-likelihoods, -e.g. negative Euclidean distance, then, the similarities computed from equations (10) to (12) need to be transformed in an appropriate form to efficiently apply Affinity Propagation. For example, cosine similarity can be transformed into negative angle distance and exponential similarity into $\|\cdot\|_{L_1} - 2t$. The values of Pearson correlation for behavioral quantities are bound in $[-1, 1]$, and its corresponding log-likelihood can be simply calculated as: $R - 1$. Although negative angle distance from cosine similarity and $\|\cdot\|_{L_1} - 2t$ from exponential similarity can be used as input log-likelihoods, we prefer to use log-likelihood $R - 1$ derived from Pearson correlation to act as input similarities for the message-passing procedure, since the range of values of $R - 1$ makes it much easier to tune node preferences to finding all probable partitions of a network into communities. The number of communities is jointly determined by user defined preferences and message-passing procedure. The larger value of preference of a node indicates that the node tends to be elected as community leader. In the absence of *a priori* knowledge, it is recommended to use a common value (common value preference, cp) such as median or minimum of the input similarities [23], to show the equal chance to be group leaders for all nodes. The number of communities can be varied by tuning the value of preferences for nodes, from the finest partition to the coarsest

one.

As anticipated, cp is not the only way to define the preference for a node, in fact, information based on the degree distribution can, also, give meaningful information. If communities were regarded as groups of nodes with statistically more edges within communities than it would be expected by chance, we could conclude that nodes with higher degrees tend to have higher probabilities to be community leaders. However, this is not always the case: for instance a node with high degree could represent a node acting as mediator between groups, therefore using node degree would be misleading. On the other hand, such mediator nodes would show low correlations with their neighbors. For this reason, we use correlation degree to replace node degree used in the original network. The correlation degree $Rdeg_i$ of node i is the total correlations of all its neighbors. The preference $pref_i$ of node i associated with correlation degree can thus be defined:

$$pref_i = -\gamma \frac{Rdeg_{max}}{Rdeg_i} \quad (13)$$

here γ is a parameter used to tune the values of preference in order to screen all possible partitions of a network into communities and $pref_i \leq -\gamma, \forall i$.

Therefore, cp and γ are two possible ways to tune the preference of a node, and ultimately to find communities in a network.

2. The proposed scheme: APCOM

The scheme of message passing between nodes by incorporating community membership based node-node similarity to find communities in a network is summarized as follows:

- (1) Compute the behavioral quantity matrix $B = \sum_{\tau=1}^t P^\tau$ by releasing from each nodes t different random walks with lengths ranging from 1 to t .
- (2) For each pair of connected nodes, compute their likelihood of being in the same community by either equation from (10) to (12).
- (3) Set the induced pair-wise likelihoods as new edge weights.
- (4) Iteratively operate from (1) to (3) to make behavioral quantities of nodes in the same communities highly consistent and those of nodes in different communities as dissimilar as possible.
- (5) Calculate log-likelihoods of Pearson correlations of all possible pairs of nodes, and determine the preferences of nodes.
- (6) Combine log-likelihoods and preferences to derive messages by Affinity Propagation, and output the most probable partition(s) of the network into communities indexed by their community leaders after message passing between nodes converges.

We call such a community finding scheme **APCOM** (**A**ffinity **P**ropagation for **COM**munity find-

ing). The complexity of APCOM consists of iterative edge reweighting, log-likelihood computation and times of message passing in Affinity Propagation. The edge reweighting can be done in $O(INk_{max}^t)$ or on average $O(IN \langle k \rangle^t)$ for a sparse network with bounded node degree as it is always the case in real world applications, where I is the number of iterations and k_{max} and $\langle k \rangle$ are the maximal and average degrees of the network, respectively. Due to the advantage of standard, hardware optimized and linear algebra software, computations of log-likelihoods of pairs of nodes are done in $O(N^2)$ if the log-likelihoods of all possible pairs of nodes are available, or $O(J)$ for the case that only the log-likelihoods of J pairs of nodes are available. Affinity Propagation can be implemented with complexity proportional to the number of available log-likelihoods [23]. Consequently, the total complexity of APCOM is $O(INk_{max}^t + 2N^2)$ or $O(INk_{max}^t + 2J)$, which is always approximately proportional to the number of available log-likelihoods.

3. Model Selection Tuning

Before applying APCOM, some practical issues must be addressed.

In some circumstances, numerical oscillations will arise when iteratively updating the messages in Affinity Propagation. To avoid such an undesirable effect, each message is set to the convex combination of its value from the previous iteration (multiplied by a damping factor λ) with its prescribed updated value (multiplied by $(1-\lambda)$), with $0 \leq \lambda \leq 1$. Higher values correspond to heavy damping, needed if oscillations occur. We used the default value $\lambda = 0.9$ suggested by the implementation of Frey *et al.* in all our applications throughout the paper.

Another important practical issue is how to select the most probable partition(s) of a network especially when the network possesses hierarchical community structure. The interpretation of nodes of a network as N -dimensional behavioral quantities vectors provides a natural tool for differentiating network partitions, by incorporating an appropriate statistic. We use the pseudo F -statistic in data clustering [38] to support the identification of the most probable partitions of a network into communities. The pseudo F -statistic is defined:

$$F_C = \frac{\sum_{s=1}^C N_s (\overline{B}^{(s)} - \overline{B})(\overline{B}^{(s)} - \overline{B})^T}{\sum_{s=1}^C \sum_{i=1}^{N_s} (B_i^{(s)} - \overline{B}^{(s)})(B_i^{(s)} - \overline{B}^{(s)})^T} \cdot \frac{N - C}{N - 1} \quad (14)$$

where C is the number of groups in the given partition; N_s is the number of nodes in group s ; $B_i^{(s)}$ is the behavioral quantity of node i in group s ; $\overline{B}^{(s)}$ and \overline{B} are the average behavioral quantities of nodes in group s and nodes in the network, respectively; $(\cdot, \cdot)^T$ represents the transpose of a vector. Higher values of F_C mean that the network is more likely to be partitioned into C groups. To apply APCOM in a practical context, we can first exam-

ine the range of preference values for which the network is partitioned in the same number of groups. We then select the partition with the largest range of preference values as the most probable community partition. In the cases of networks with clear community structures and only one level of communities, such a strategy works very well as we will see in the following section. In other cases, when a network shows hierarchical community structure, and thus several community partitions of the network are all the most probable ones, we not only examine the range of preference values but also calculate the values of the pseudo F -statistic to select the most probable partitions. The partitions can thus be ranked first by the range of preference values and secondly by the values of the F -statistic.

The final issue is the choice of values of the random walk length t and the number of iterations I , although they do not show a very crucial influence on the results. Basically, t should not be chosen too large, i.e. possibly not greater than $O(\log(N))$, due to the exponential convergence speed of random walk. When $t \rightarrow \infty$ (random walk in this extreme has already converged), the behavioral quantity matrix B converges to a constant matrix with all identical rows. Edge reweighting in this extreme case is equivalent to discarding all edge weights, which discards valuable information that defines communities of weighted networks and should therefore be avoided. In addition, edge reweighting tends to give a coarser description of a network when t increases, and consequently the number of communities found will sometimes decrease. However, the number of communities found by APCOM will mainly be determined by the values of node preference, and the community structures that tend to be hidden by increasing value of t will be disclosed again by tuning the values of node preference. As a result, the value of t will have little impact on the results as long as it is not very large. The number of iterations I is not a crucial parameter and it is somewhat dependent on the fuzziness of the community structure of a network. If a network has very clear community structure, only a few iterations are sufficient to capture the necessary information for computing likelihoods. If the community structure of a network is very fuzzy, it is more challenging with edge reweighting to determine as accurately as possible the likelihoods of pairs of nodes to be in the same communities, this, however, represents a challenge for other state-of-the-art community finding methods.

IV. APPLICATIONS

APCOM is here applied to real and synthetic networks, all with known community structures. Given a network, similarities between all N^2 possible pairs of nodes are immediately available by either equation from (10) to (12). APCOM is thus implemented and applied by using all such information. Without explicit declaration, the results shown in this section are obtained by using random

walk length as $t = 6$ and iterating 10 times to reweight the edges. But it should be pointed out that the results obtained by APCOM are robust with respect to the values of the parameters used. Three state-of-the-art modularity based community finding algorithms are chosen for comparison: an eigenvector based method proposed by Newman denoted *EigenMod* [11], its variant for directed networks denoted *DEigenMod* [39], and a very efficient algorithm denoted *FastMod* (also referred to as Louvian method) by Blondel *et al.* [12]. Generally, other modularity based algorithms can also be chosen for comparison, such as the ones presented in [6] and [10]. The ones used in this paper were chosen for their efficiency and comparatively high performance.

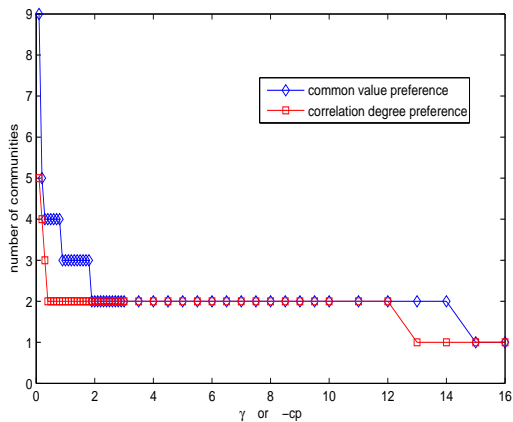
A. Zachary karate club network

This network consists of 34 nodes as members of the karate club and 78 edges as friendships between members [40]. The network had been split into two disjoint groups during the years that W.W. Zachary studied it, due to the disagreement between the administrator and the instructor of the club.

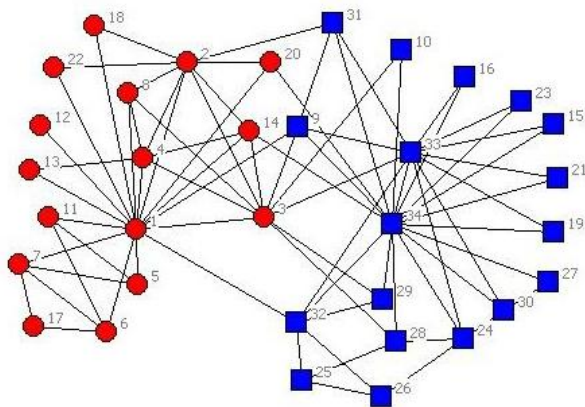
We used APCOM to partition this network by varying preferences. Two different preference setting strategies are used: correlation degree preference and common value preference. Compared to other similarity metrics such as negative Euclidean distance, the values of log-likelihood of correlation are bound in a narrower range $([-2, 0])$, which makes it much easier to tune the values of preference. Since the minimal value of $R-1$ is -2, we sampled more values of both γ and $-cp$ in the interval $[0, 2]$. As shown in FIG. 1 (a) APCOM using the correlation degree preference setting strategy obtained most of the partitions as 2 communities with nodes 1 and 34 being the corresponding groups leaders, which is the expected splitting of this network, as shown in FIG. 1 (b). In contrast to correlation degree preference taking into account different node roles in the network, common value preference strategy produces other two less frequent partitions: 3 and 4 communities, which are the subdivisions of the real splitting, as always found by modularity-based methods [10–12]. We also tested the performance with different values of t : smaller value ($t = 3$), moderate value ($t = 6$) and larger value ($t = 13$), and found that this gave very consistent results, meaning that APCOM is robust with respect to the parameter of random walk length t . The reason is that the influence imposed by t will be reduced by varying the values of preference as discussed in the previous section.

B. Football network of American college

We further applied APCOM to the network of American college football games during the 2000 fall regular season [8]. This network has 115 teams as nodes and



(a)

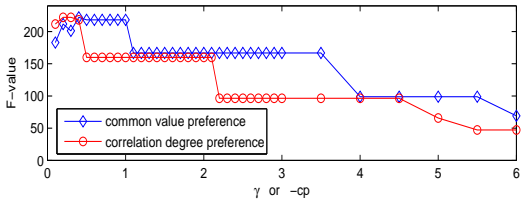
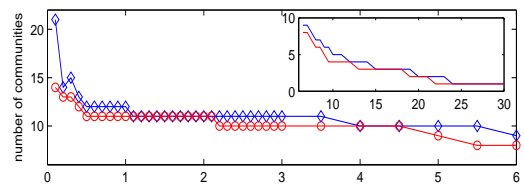


(b)

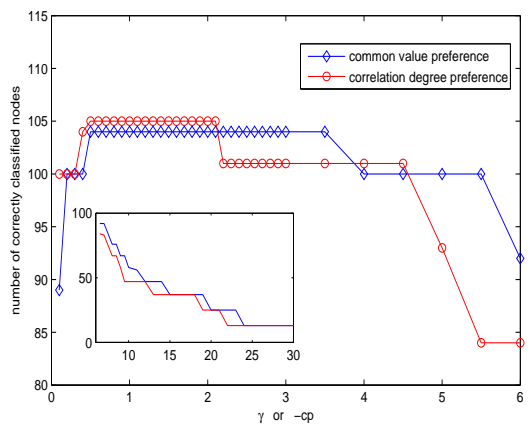
FIG. 1. (Color online) Results of applying APCOM to the Zachary karate club network. (a) Number of communities found by setting different preferences of nodes. (b) The partition most frequently found by APCOM.

613 edges representing games played between the teams joined. 115 teams come from 12 conferences and games are more frequent between teams from the same conference than between teams from different conferences.

By varying γ or the value of common preference cp , two types of partitions are more frequently found, as shown in the upper subfigure of FIG. 2 (a). These two types of partitions produce 11 and 10 communities, respectively, which are both probable if we lack of additional information. In this case, the difference between the ranges of preference for the two possible partitions is so undistinguishable that it cannot easily tell how to identify the most probable one, unless the network shows hierarchical structure. Fortunately, since APCOM views each node of a network as an N -dimensional vector in a metric space, we can use the pseudo F -statistic to address such issue, as shown in the lower subfigure of FIG. 2 (a). The ranges of preference that give the same partition and the values of F -statistic jointly determine the partition of 11 communities to be the most probable one. FIG. 2 (b) gives the



(a)



(b)

FIG. 2. (Color online) Results of applying APCOM to the football network of American college. (a) Number of communities found by APCOM with different preferences (upper), and the corresponding F -statistic value (lower); (b) Number of correctly classified nodes out of 115. Insets show the further results of larger values of γ or common preference $-cp$.

corresponding accuracies in terms of correctly classified nodes out of the total 115 (the accuracy was calculated by searching the largest common members between the communities found versus the real ones, as in [28]), which indicates that the partition of the network into 11 communities is indeed the most probable one. Comparison between two different preference setting strategies shows that correlation degree is much easier to find the most probable partition since it takes into account the different roles played by the nodes in the network. When applying modularity based community finding algorithms to this network (EigenMod and FastMod), they tend to partition the network into 10 communities and thus miss the most probable community partition of this network.

C. Dolphins' social network

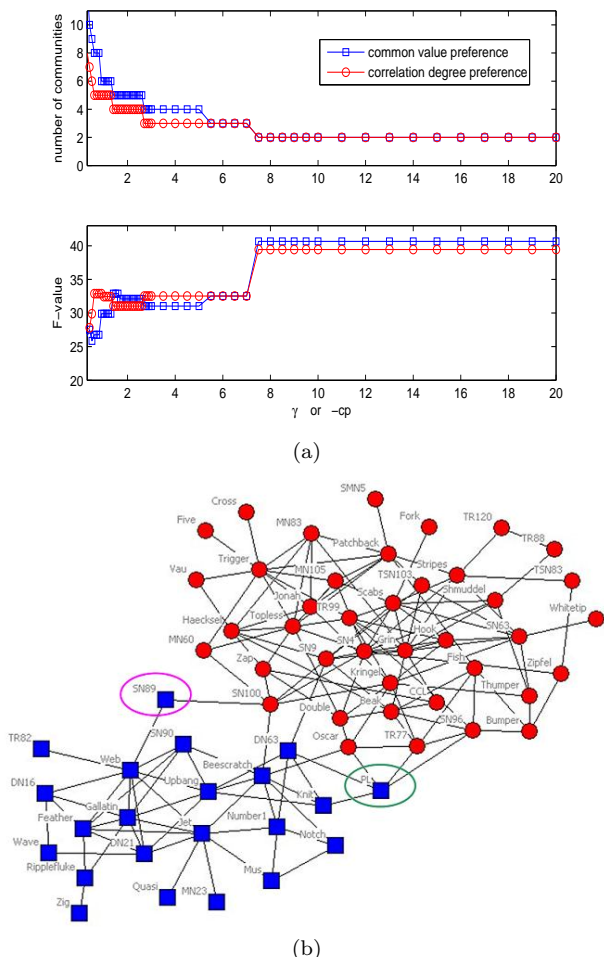


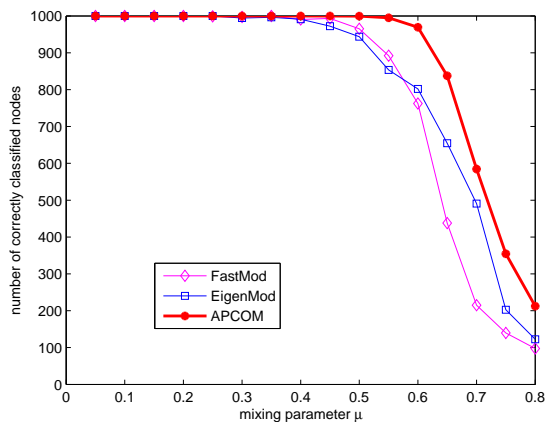
FIG. 3. (Color online) Results of applying APCOM to dolphins' social network. (a) Number of communities found by APCOM by varying the values of preference (Upper) and the corresponding F -statistic value (Lower) to show the compactness of each partition. (b) The most probable partition into 2 communities by APCOM with correlation degree preference. Colors and shapes indicate different community memberships. Nodes in ellipses are misclassified nodes according to the real communities at $t = 6$.

We apply APCOM to another social network analyzed by Lusseau *et al.*, which is a network of bottlenose dolphins living in Doubtful Sound in New Zealand [41]. The network consists of 62 dolphins as nodes, and the edges between nodes indicate that the dolphin pairs were seen more often than expected by chance. The dolphins separated in two groups after a dolphin denoted SN100 left the place for some time. Applying APCOM with correlation degree preference or common value preference to this social network, the partitions of the network into 2 communities were found to be the most probable partitions (see FIG. 3 (a)). The partition obtained by APCOM

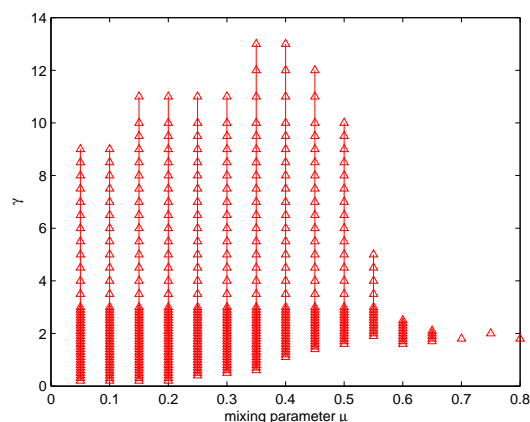
with correlation degree preference misclassifies dolphins denoted SN89 and PL (nodes in ellipse in FIG. 3 (b)), while the one with common value preference misclassifies the dolphin denoted PL. It is interesting to observe that the second largest eigenvalue of $D^{-1}W$ is very close to 1 (> 0.98), meaning that random walk on this network will converge slowly. Thus t can be set as large as possible before random walk converges. When $t \geq 14$, APCOM with correlation degree preference, partitions the network into 2 communities with only the dolphin denoted PL being misclassified, like it happens when common value is set as preference. The similarity of the partition with larger t to that with smaller t explains again the robustness of APCOM with respect to parameters variations. When applying modularity based community finding algorithms to this network, the network is split into more than 2 communities (4 communities by EigenMod and 5 communities by FastMod), failing to produce the expected partition. In contrast, APCOM reveals several possible partitions of this network and suggests the partition into 2 communities as the most probable.

D. Undirected LFR benchmark networks

Besides applying APCOM to real networks, we turn to use a set of recently introduced synthetic benchmark networks with realistic features [42]. The degrees and the community sizes of these networks are both distributed as power law with different exponents t_1 and t_2 , respectively. The ratio between the external degree of each node with respect to its community and the total degree of the node is determined by a common mixing parameter μ . The value of μ controls the fuzziness of community structure of a network and larger value means that the network is harder to be decomposed into its real communities. We generated different instances of such type of networks with 1000 nodes each, and used the default values of parameters: $t_1 = -2$ and $t_2 = -1$. The average degree of each network is 20 while the maximal degree is 50. The value of μ was varied from 0.05 to 0.8 to obtain networks with different fuzziness of community structures (the minimal and maximal sizes of communities are 20 and 50, respectively). The results of applying APCOM to such type of networks are shown in FIG. 4, with each point averaged on 10 different instances. The accuracy was calculated in terms of number of correctly classified nodes by making a mapping between communities found and the real ones, with the largest common members [27, 28]. As it can be seen from FIG. 4 (a), APCOM shows relatively higher performance than that of FastMod and EigenMod. For a network with clear community structure, accuracy with APCOM is computed by selecting any partition with γ in the range of values that specify the partitions as dominant among all possible partitions obtained by various values of γ , as it is shown for example in FIG. 4 (b). For convenience, the results presented in FIG. 4 (a) were obtained by setting



(a)



(b)

FIG. 4. (Color online) Results of applying different methods to undirected LFR benchmark networks with 1000 nodes. (a) Performance comparison between modularity-based methods and APCOM. (b) The ranges of values of γ that correctly find the true number of communities in a sampled network with 30 built-in communities.

$\gamma = 2$. APCOM gives consistent results also when setting preference to common values (data not shown).

To assess if the performance of APCOM varies on benchmarks with much larger scale, we similarly generated two sets of networks: one consisting of networks with 5000 nodes each and the other with 10000 nodes each. The parameters' value are the same used for generating networks with 1000 nodes in the previous application except for the 10000-node networks where the minimal and maximal sizes are set to 20 and 200, respectively, to obtain networks with more diverse communities. For these larger networks, we compare APCOM to FastMod only (since EigenMod in these cases is not faster nor more accurate than FastMod) employing two different preference setting strategies. To test these networks, we also reduced by a half the length of the random walk to consequently half the number of iterations, i.e., $t = 3$ and

$I = 5$ in these cases. The results are given in FIG. 5, which clearly shows that APCOM performs much better than FastMod. In these cases, even on networks with obvious community structures, FastMod cannot find the communities correctly. The worse behavior of FastMod is due to the severe resolution limit problem, while APCOM can effectively tackle such a difficulty. It is worth adding some notes on the results presented in the bottom right subfigure of FIG. 5. In the case of networks with 10000 nodes when $\mu \geq 0.6$ and preference set to common value, APCOM is not warranted to converge for all values of preference and in such cases every node forms its own group and such a partition becomes the dominant partition. This trivial solution can be simply discarded or eliminated by tuning the damping factor in APCOM and subsequently selecting the dominant partitions as the most probable partitions. However, to avoid the use of *ad hoc* manipulations that force the algorithm to converge, and to simplify the parameters tuning (preserving the consistency of the usage of APCOM in different conditions), we set the accuracies to zero since the performance comparison in the cases of networks with clear community structure ($\mu \leq 0.5$) is much more meaningful.

We must note that, although FastMod suffers from resolution limit, it has its own advantage over APCOM. In fact, FastMod is very fast (approximately $O(L)$, where L denotes the number of edges) and can be used to roughly partition mega-scale networks into communities provided large memory space is available. Since APCOM uses similarities between all possible pairs of nodes in a network, it thus scales approximately $O(N^2)$ and it is faster than EigenMod but it cannot as fast as FastMod. As a result, APCOM employing similarities of all possible pairs of nodes is more useful for networks with several tens of thousands of nodes (for example biological networks). However, APCOM can be effectively implemented by taking only $O(J)$ pairs of similarities with J approximate to $O(N)$ by some appropriate heuristics. This is briefly further discussed in the Conclusion section.

E. Clique chained networks

Modularity maximization is found to suffer from resolution limit. To examine if APCOM suffers from a similar limitation, we constructed a series of networks chained by 5-node cliques with only single edge between each neighbor pair of cliques [15]. The number of cliques varies from 100 to 1000 and thus the number of nodes from 500 to 5000. After reweighting the original edges of networks, APCOM correctly detected the number of cliques chained in the networks, compared to the failure of FastMod in the whole range, as shown in Table. I. When varying the values of parameter γ for setting preference in Affinity Propagation, APCOM found partitions of networks into their corresponding correct number of cliques as the most probable partitions. The range of γ is so wide that it is easy for APCOM to suggest the most proba-

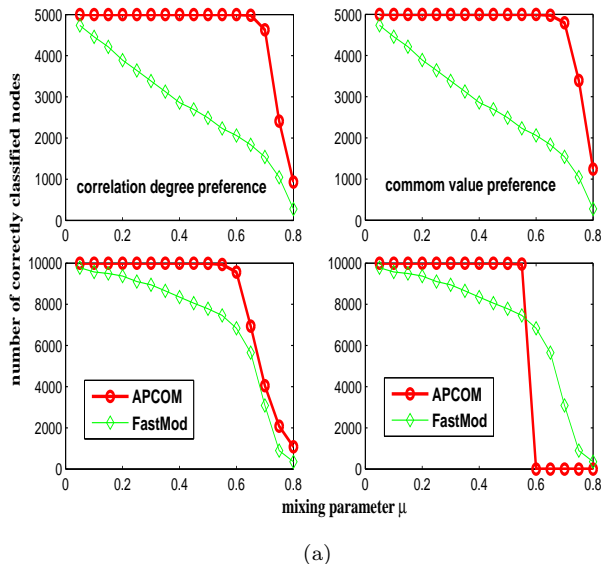


FIG. 5. (Color online) Results of applying different methods to undirected LFR benchmark networks with 5000 (top) and 10000 (bottom) nodes, respectively. (Left) Performance comparisons between FastMod and APCOM employing correlation degree preference. (Right) Performance comparisons between FastMod and APCOM employing common value preference.

ble partitions without any additional information. [Compared to other partitions, the partition of a network into cliques as communities is the dominant partition among the partitions obtained from the whole possible range of γ] APCOM works well also when the preference is set by common values. Consistently with the discussion in the previous section the performance of APCOM is robust to the value of random walk length t .

F. Directed Flow networks

If exponential similarity in equation (12) is used, APCOM can also be applied to detect communities in directed networks. We first considered a directed network with 16 nodes, proposed by Rosvall and Bergstrom [43] to test their information theory based community finding algorithm called *InfoMap*. This network introduces a structure pattern generating persistent movement within nodes of the same color and shape and limited movement between nodes of different colors and shapes, as shown in FIG. 6. The weights of the bold edges are twice those of normal edges. As it has been reported [28, 43], the direct application of modularity-based community finding algorithms to this network could not detect the regularity induced by the network topology. Conversely, when applying APCOM to this network by first extracting information contained in edge directions, the regularity hid-

den in this network was disclosed.

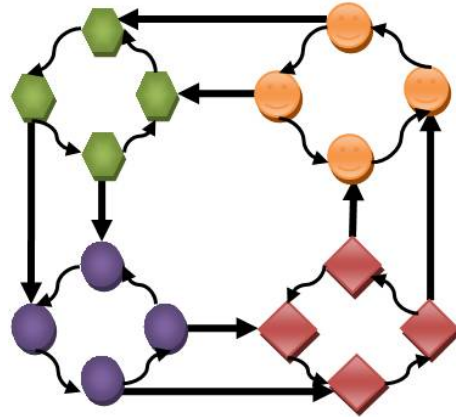


FIG. 6. (Color online) A 16-node directed network with 4 flow circles built in as communities, highlighted by different colors and shapes. The weights of the bold edges are twice the normal ones.

If all the edge weights of the network in FIG. 6 are equal, modularity-based community finding algorithm for directed network such as DEigenMod can correctly find the regularity hidden in the structure, unsurprisingly so can do InfoMap and APCOM. However, InfoMap cannot always detect correctly the regularity in such type of network if the number of chained components (directed flow circles, each of which consisting of four nodes) increases, neither can DEigenMod [39]. To see more clearly the behavioral differences between InfoMap and APCOM, we generated similarly a set of such type of networks by varying different numbers of components [28]. The number of chained components varies from 100 to 1000 and thus the number of nodes from 400 to 4000. Results are summarized in Table. II. Actually, InfoMap cannot detect the correct number of components chained in the network as long as the number is greater than 23, and DEigenMod performs even worse for its bisecting nature, which fails as soon as the number is greater than 5. When the number of chained components becomes larger, InfoMap performs very poorly on this type of networks, as shown in Table.2. The failure of InfoMap is due to the merging of neighbor components, which is a phenomenon similar to the one found in modularity maximization as resolution limit [15]. In contrast, APCOM can find the correct number of components chained in such type of directed flow networks, meaning that APCOM effectively addresses the resolution limit problem. The range of values of parameter γ is also wide ($0.5 \leq \gamma \leq 3$) indicating that the regularity hidden in the networks can be easily detected.

G. Directed LFR benchmark networks

To further evaluate the performance of APCOM, we applied it to LFR directed benchmark networks [42]. The

TABLE I. Performance comparison between FastMod and APCOM on different number (100 to 1000) of cliques chained in the networks. The results of APCOM were obtained as long as $0.05 \leq \gamma \leq 3.5$. FastMod fails to detect the correct number of cliques in the whole range, while APCOM can correctly detects cliques as communities.

Method/cliques	100	200	300	400	500	600	700	800	900	1000
FastMod	25	50	75	50	63	75	88	100	112	125
APCOM	100	200	300	400	500	600	700	800	900	1000

TABLE II. Performance comparison between InfoMap and APCOM on directed flow networks with different number of flow components chained in the networks. The results of APCOM were obtained as long as $0.5 \leq \gamma \leq 3.5$. InfoMap fails to detect the correct number of components in the whole range, while APCOM can correctly find the regularity in the networks.

Method/flow circles	100	200	300	400	500	600	700	800	900	1000
InfoMap	54	109	162	178	219	265	305	350	392	438
APCOM	100	200	300	400	500	600	700	800	900	1000

parameters used were the same as those in the previous section. For directed networks, we first iteratively reweighted edges and then treated them as undirected networks as in [28]. Due to the role of edge direction in directed networks, common value preference is more suitable. The results obtained by APCOM with common value preference are shown in FIG. 7. Compared to DEigenMod and InfoMap, APCOM performs very well even when the community structures of networks are fuzzy ($\mu \geq 0.6$). In addition, APCOM finds the partitions of networks into their real communities as the most probable community partitions, as shown in terms of the wide ranges of values of preference in FIG. 7 (b).

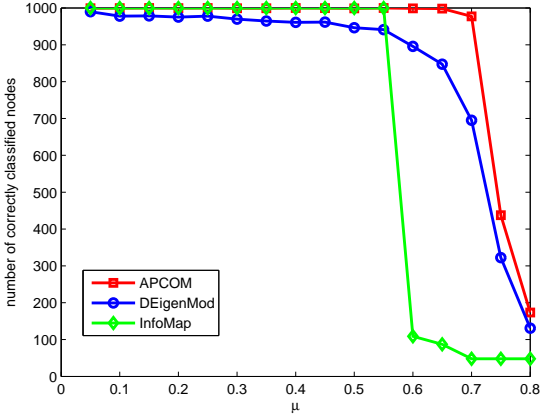
H. Hierarchical networks

From the above described applications, we can see that, by varying the values of preference, APCOM can reveal the hierarchical structure of a network if the network is organized in multiple levels. We thus applied APCOM to two types of networks with two levels of hierarchical structures. The first type of network contains two networks, each of which consists of 256 nodes and is similar to the one used to illustrate synchronization method to find communities [44]. The lower levels of the networks are both organized into 16 communities, each of which consists of 16 nodes. Every 4 communities in the lower level are organized into a larger community in the higher level where 4 larger communities are formed. This type of network is the generalization from a benchmark for community finding algorithms [9]. Since such type of network is originally constructed as a set of random networks, we instead construct them in a slightly different way: the degree of each node is fixed exactly at 18. The internal degrees of nodes of these two networks at lower level are respectively 13 for one and 15 for the other, and the internal degrees of nodes at higher level are 4 and 2, while there is only 1 edge of each node that connects with any community of the rest of the network.

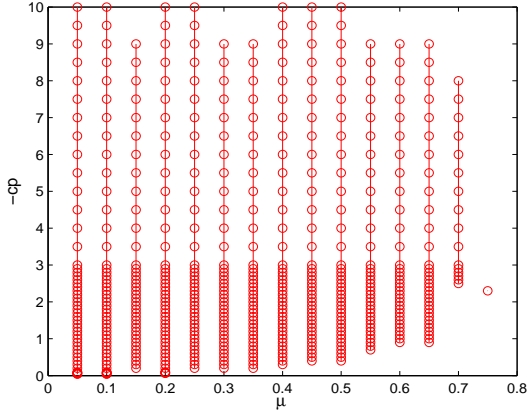
We similarly denote these networks H13-4 and H15-2 for convenience.

For the H13-4 network, two most frequent partitions were found by APCOM when varying the values of parameter γ (results of APCOM with common value preference are the same since nodes in the network are topologically identical), which are the original hierarchical structures by construction: the lower level of 16 groups of 16 nodes and the higher level of 4 groups of 64 nodes (see FIG. 8). In fact, the two levels (partitions) of H13-4 can be interpreted from two different points of view. The much wider range of γ values indicates that the higher level of 4 communities is much more robust than the lower level of 16 communities when varying the tendency of nodes to be community leaders. The result from such a point of view is also found in [45] in terms of the stability of synchronization. Besides, since each node of the network is embedded into an N dimensional space, the results obtained by APCOM can also be interpreted from the compactness point of view. The F -statistic is an objective function that "prefers" compact sphere groups. Compared to the configuration of 4 groups of 64 nodes, the F -statistic value for the configuration of 16 groups of 16 nodes is extremely higher, meaning that the lower level organization tends to cluster nodes of the H13-4 network into more compact groups. In consideration of the fact that for most of the networks N is high, APCOM is used by first examining the range of the values of preference and secondly the F -statistic values. Only in cases that possible ranges of the values of preference cannot easily be distinguished, the F -statistic value takes effect and we observed that it works appropriately throughout a wide range of applications tested in this paper. For the H15-2 network, the lower level of 16 communities with 16 nodes each is suggested to be the most probable partition from both the robustness and the compactness points of view, while its higher level organization is not so obvious. Such a finding is also confirmed in [45]. In fact, the 16 communities in the lower level of the H15-2 network are cliques, which are intuitively interpreted as communities.

The second network tested is a scale-free hierarchi-



(a)



(b)

FIG. 7. (Color online) Results of applying APCOM to directed LFR benchmark networks. (a) Performance comparisons among DEigenMod, InfoMap and APCOM with common value preference. (b) The ranges of values of preference that correctly find the true number of communities in a sampled network with 43 communities built in.

cal network with 125 nodes, proposed by Ravasz and Barabasi [46], denoted RB125. By setting correlation degree preference in Affinity Propagation, APCOM found only two partitions of this network. The most frequent one (as indicated by the range of values of γ) partitions the network into 5 communities (see upper subfigure in FIG. 9), each of which consists of twenty-five nodes, as enclosed in five larger circles shown in FIG. 10. Such a partition is exactly the higher level of RB125 by construction. The other one found by APCOM is the partition of RB125 into 9 communities, as indicated by shadows in 9 circles (four larger and five smaller circles) in FIG. 10. Although it is different from the lower level of the network by construction, this partition of 9 communities interestingly interprets this scale-free network RB125 as a tree-like structure with the most central 5-node group as root. In fact, all the other 8 communities in this par-

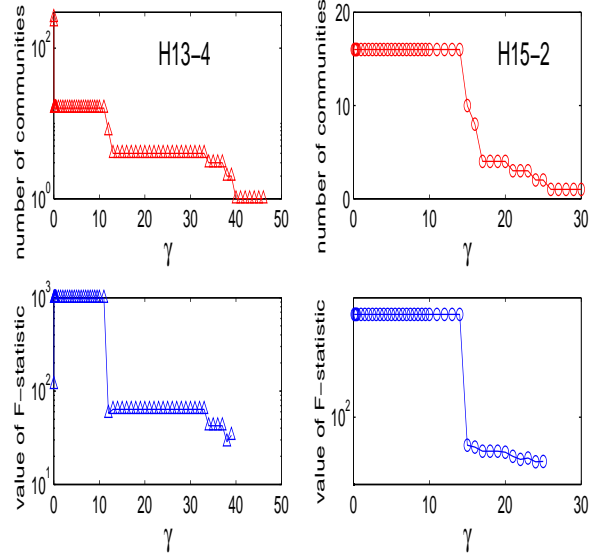


FIG. 8. (Color online) Multi-level partitions of H13-4 network by APCOM. (a) Number of communities found for different values of γ . (b) Values of the F -statistic of the corresponding partitions.

tion are its immediate sons or daughters, showing an affiliation like organization of the RB125 network. Coherently, Arenas *et.al* [20] also found a partitioning of this network into multi-scales, where the most frequent partition is into 26 communities (the most central hub node is separated as a single community). In the framework of APCOM, however, the most central hub node is not isolated as a single community but acts as the community leader of other nodes. When γ is small, nodes are assigned to large preferences and they thus have high probabilities to be community leaders. Since the central hub node is connected to nearly all the nodes of the network (therefore the overall similarities to other nodes is high), this configuration is not in favor of separating the central hub node as a single community according to the target function in equation (4). Intuitively, isolating the central hub node can be thought of as creating a new community from the previous community assignment. Such an operation would cause the increase of the total energy (or decrease of the overall similarity) in equation (4) since the gain from the preference of the central hub node tends to be insufficient to compensate for the loss caused by reassigning the members indexed by the hub node to other communities (indexed by less connected nodes). When γ becomes larger, the competition for community leadership is mainly dominated by the similarities between nodes, and the cost of isolating the central hub as a single community is increasingly high. Such a behavior in the framework of APCOM is somewhat similar to that of modularity-based approaches although they are completely different methods.

We also applied APCOM employing common value

preference setting strategy to the RB125 network, but results are less satisfactory (see FIG. 9 lower subfigure), as the most frequent partition appears to be the one consisting of 4 communities which is somewhat against the construction of the network. By varying cp , APCOM can find two expected partitions: 25 communities of 5 nodes each and 5 communities of 25 nodes each. The partition of 5 communities can be thought of as the second most frequent by inspecting the range of the values of cp . However, the partition of the network into 25 communities seems to lack of strong evidence since in this setting many other partitions coexist, a phenomenon encountered by using other frameworks as well [20].

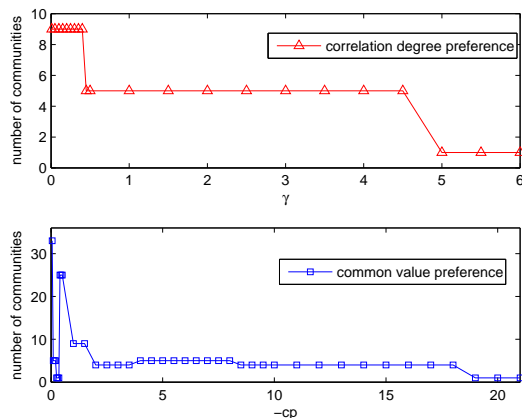


FIG. 9. (Color online) Different partitions of the RB125 network into communities by APCOM with different preference setting strategy. (Upper) Only two stable partitions by using correlation degree preference. (Lower) Partitions obtained by using common value preference.

Two modularity based algorithms, EigenMod and FastMod, were also applied to partition the RB125 network. They both found six communities: four peripheral communities of 25 nodes, one 5-node community and the one consisting of four 5-node groups. The community with 5 nodes is however not the one that contains the most central hub node, but one of the 4 peripheral 5-node groups in the central larger circle, see FIG. 10. To analyze multiple levels of community structure of the RB125 network, we first used InfoMap to partition the network and then construct a new network which nodes are the communities found. The second level of communities was then obtained by applying InfoMap to this new network. The two levels of communities found by InfoMap are: 22 communities in the lower level (2 communities from the division of the most central 5-node group where the most central hub node is separated from other 4 nodes as a single community, 4 communities consisting of the merging of two 5-node groups, and sixteen 5-node communities) and 5 communities in higher level. According to the community structure of the RB125 network by construction, modularity based methods miss the two exact levels of communities and InfoMap misses the lower level.

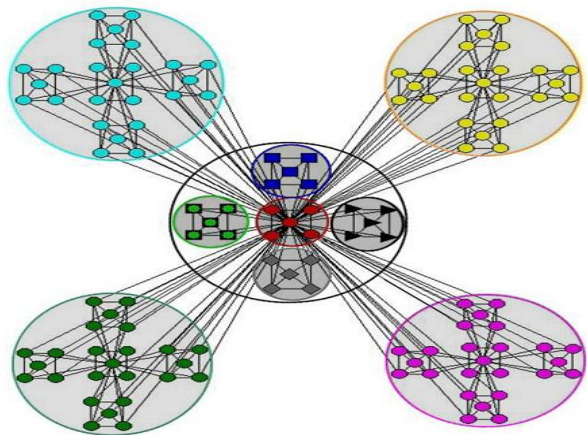


FIG. 10. (Color online) Two stable levels of RB125 network found by APCOM. Shadows in circle represent the lower level organization consisting of 9 communities, while the partition indicated by five larger circles gives the higher level of 5 communities.

V. CONCLUSION AND DISCUSSIONS

If the nodes of a general network are effectively embedded in a metric space, finding communities of a network is equivalent to clustering data points in that metric space. By means of random walk on networks, a community finding scheme named APCOM has been proposed in this paper. APCOM operates on data points (embedded nodes of a network) in a metric space and treats partitioning a network into communities as finding community leaders in that network. Community leader election is accomplished by passing messages between nodes. To derive messages, APCOM first evaluates by random walk the likelihood of each pair of connected nodes of a network to be in the same community, and uses then the induced likelihoods to iteratively reweight the original network. Starting from the novel reweighted network, APCOM derives a similarity for every possible pair of nodes to show how well one node is suited to be the community leader of the other. With the derived similarities and with user prescribed preferences for nodes to be community leaders, APCOM adopts Affinity Propagation clustering method to execute a message passing procedure. After message passing converges, communities emerge as indexed by their corresponding community leaders.

We have applied APCOM to a series of real and synthetic directed and undirected networks, and the much higher performance of APCOM than that of state-of-the-art community finding algorithms demonstrates its effectiveness in community finding. Compared to previous methods which also embed nodes of networks into metric space [29, 30], APCOM is more general and efficient. The generality comes from the similarity calculation strategy adopted by APCOM. Such a similarity measure is based on community membership of nodes and induced

by random walk on networks, which relaxes the strong assumption that connected nodes must be similar. Since the similarity is not computed by Eigen-decomposition but instead by matrix-vector multiplications, it can be computed efficiently. In addition, the complexity of message passing process in APCOM is always proportional to the number of messages exchanged. Thus APCOM can be used efficiently. More importantly, even if there are user defined parameters in APCOM, the performance of APCOM is very robust to the values of those parameters, which makes APCOM easy to be operated. Thanks to appropriate similarity calculation strategies, APCOM provides a simple but effective unified framework for finding communities in both directed and undirected networks and simultaneously evaluates the robustness of all possible community partitions by changing the tendency of nodes (node preference) to be community leaders. The unified form of APCOM is important since directly extending community finding methods for undirected networks to methods for directed networks is not so obvious and always challenging.

Compared to other techniques, such as C-means or C-median to prescribe the number of groups [24], the value of user defined node preference in APCOM is much easier to choose, since its value is always bound in a very reduced range and only very limited sample values are needed. As discussed in the last section, APCOM always works by tuning node preference to find robust communities in networks. In this respect, APCOM should be considered as a framework that simultaneously partitions a network into communities and evaluates the robustness of all possible partitions. The robustness can be easily evaluated since in the plot of the number of communities versus the values of γ or cp there will always be at least one obvious plateau if a network has indeed clear community structure. Thus the direct benefit is that, if the networks possess hierarchical community structure, APCOM can also screen their possible community structure at different levels of resolution. In addition, the nature of APCOM to embed nodes into a metric space makes it possible to identify the most probable partition(s) of the network into communities by incorporating appropriate statistical tools for vectors. It is worth noting that such a behavior is very similar to that of a recent interesting work by R. Lambiotte [47], where the proposed framework finds robust partition(s) by modifying quality functions with different values of parameters. However, the intrinsic mechanism of APCOM for finding robust partition(s) is different from the one pro-

posed by R. Lambiotte. In fact, APCOM varies the value of node preference to change the tendency of nodes being the community leaders while it retains the similarity relationship between pairs of nodes. If a network has clear community structure, such type of variation cannot change the strongly preferred community memberships of nodes. In addition, the plateau of the robustness plot tends always to correspond to the same partition in contrast to the case in [47] where the partitions are usually not exactly equivalent, as the strategy proposed in [47] alters the similarity relationships between pairs of nodes by varying the time scales.

Finally, some notes are drawn for future work. Since similarities between all N^2 possible pairs of nodes of a given network are immediately available by either equation from (10) to (12), APCOM is implemented and applied in the current work by using all such information, which somewhat reduces the effectiveness and makes APCOM currently more useful for networks with several tens of thousands of nodes. However, APCOM can in fact be implemented in a much faster way by considering heuristics reflecting the local nature of communities, for example, by considering only similarities between a node and its nearby direct or indirect neighbors, necessitating of a number of similarities in an approximate scale of $O(N)$. Various heuristics for such an implementation, which would allow the handling of networks with hundreds of thousands nodes, together with other possible node preference setting strategies, are left for further research on an important improvement and a complex enhancement of the algorithm not to distract the focus of the current work. Besides, other similarity measures, for instance, average first passage time induced similarity [32, 33], graph kernels based similarity [48], matrix based similarity [26, 29, 30] and so on, can be used to integrated into APCOM to partition networks into communities (although such similarity measures have either high computation complexity or impose strong assumptions on network edges).

ACKNOWLEDGMENTS

The authors thank M.E.J Newman for providing Zachary Karate club network and American football network data. This work is supported by National Natural Science Foundation of China under grant No. 60873133 (NSFC, No. 60873133).

[1] R. Albert and A. L. Barabási, *Reviews of modern physics* **74**, 47 (2002).
 [2] M. E. J. Newman, *SIAM Review* **45**, 167 (2003).
 [3] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. U. Hwang, *Physics Reports* **424**, 175 (2006).
 [4] G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee,

IEEE Computer **35**, 66 (2002).
 [5] A. E. Krause, K. A. Frank, D. M. Mason, R. E. Ulanowicz, and W. W. Taylor, *Nature* **426**, 282 (2003).
 [6] R. Guiméra and L. A. N. Amaral, *Nature* **433**, 895 (2005).
 [7] A. W. Rives and T. Galitski, *Proc. Natl. Acad. Sci. USA*

- 100**, 1128 (2003).
- [8] M. Girvan and M. E. J. Newman, Proc. Natl. Acad. Sci. USA **99**, 7821 (2002).
- [9] M. E. J. Newman and M. Girvan, Phys. Rev. E **69**, 026113 (2004).
- [10] J. Duch and A. Arenas, Phys. Rev. E **72**, 027104 (2005).
- [11] M. E. J. Newman, Proc. Natl. Acad. Sci. USA **103**, 8577 (2006).
- [12] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, Journal of Statistical Mechanics: Theory and Experiment, P10008(2008).
- [13] S. Fortunato, Physics Reports **486**, 75 (2010).
- [14] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, Journal of Statistical Mechanics: Theory and Experiment, P09008(2005).
- [15] S. Fortunato and M. Barthélemy, Proc. Natl. Acad. Sci. USA **104**, 36 (2007).
- [16] J. Ruan and W. Zhang, Phys. Rev. E **77**, 016104 (2008).
- [17] Z. Li, S. Zhang, R. S. Wang, X. S. Zhang, and L. Chen, Phys. Rev. E **77**, 036109 (2008).
- [18] J. Reichardt and S. Bornholdt, Phys. Rev. E **74**, 016110 (2006).
- [19] J. M. Kumpula, J. Saramäki, K. Kaski, and J. Kertész, Eur. Phys. J. B **56**, 41 (2007).
- [20] A. Arenas, A. Fernández, and S. Gómez, New J. Phys. **10**, 053039 (2008).
- [21] J. C. Delvenne, S. N. Yaliraki, and M. Barahona, Proc. Natl. Acad. Sci. USA **107**, 12755 (2010).
- [22] R. Lambiotte, J. Delvenne, and M. Barahona, arXiv: **0812.1770** (2008).
- [23] B. J. Frey and D. Dueck, Science **315**, 972 (2007).
- [24] J. Han and M. Kamber, *Data mining: concepts and techniques* (Morgan Kaufmann, 2006).
- [25] D. Lai and H. Lu, Modern Physics Letters B **22**, 1547 (2008).
- [26] E. A. Leicht, P. Holme, and M. E. J. Newman, Phys. Rev. E **73**, 026120 (2006).
- [27] D. Lai, H. Lu, and C. Nardini, Phys. Rev. E **81**, 066118 (2010).
- [28] D. Lai, H. Lu, and C. Nardini, Journal of Statistical Mechanics: Theory and Experiment, P06003(2010).
- [29] L. Donetti and M. A. Munoz, Journal of Statistical Mechanics: Theory and Experiment, P10012(2004).
- [30] D. Lai, H. Lu, and C. Nardini, Physica A: Statistical Mechanics and its Applications(2010).
- [31] A. Arenas, J. Duch, A. Fernandez, and S. Gómez, New J. Phys. **9**, 176 (2007).
- [32] H. Zhou, Phys. Rev. E **67**, 041908 (2003).
- [33] H. Zhou, Phys. Rev. E **67**, 061901 (2003).
- [34] P. Pons and M. Latapy, Lecture notes in computer science **3733**, 284 (2005).
- [35] A. N. Langville and C. D. Meyer, Internet Mathematics **1**, 335 (2004).
- [36] S. Brin and L. Page, Computer Networks and ISDN Systems **30**, 107 (1998).
- [37] R. A. Johnson and D. W. Wichern, *Applied multivariate statistical analysis* (Prentice Hall Englewood Cliffs, NJ, 1998).
- [38] T. Caliński and J. Harabasz, Communications in Statistics-Simulation and Computation **3**, 1 (1974).
- [39] E. A. Leicht and M. E. J. Newman, Phys. Rev. Lett. **100**, 118703 (2008).
- [40] W. W. Zachary, Journal of Anthropological Research **33**, 452 (1977).
- [41] D. Lusseau, Proceedings of the Royal Society of London. Series B: Biological Sciences **270**, S186 (2003).
- [42] A. Lancichinetti and S. Fortunato, Phys. Rev. E **80**, 016118 (2009).
- [43] M. Rosvall and C. T. Bergstrom, Proc. Natl. Acad. Sci. USA **105**, 1118 (2008).
- [44] A. Arenas, A. Diaz-Guilera, and C. Perez-Vicente, Phys. Rev. Lett. **96**, 114102 (2006).
- [45] A. Arenas and A. Diaz-Guilera, The European Physical Journal-Special Topics **143**, 19 (2007).
- [46] E. Ravasz and A. L. Barabási, Phys. Rev. E **67**, 026112 (2003).
- [47] R. Lambiotte, Arxiv preprint arXiv:1004.4268(2010).
- [48] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis* (Cambridge University Press, 2004).