Soft Computing manuscript No.
(will be inserted by the editor)

Fuzzy Community Structure Detection by Particle Competition and Cooperation

Fabricio Breve · Liang Zhao

Received: date / Accepted: date

Abstract Identification and classification of overlapping nodes in networks is an important topic in data mining. In this paper, a network-based (graph-based) semi-supervised learning method is proposed. It is based on competition and cooperation among walking particles networks to uncover overlapping nodes by generating continuous-valued outputs (soft labels), corresponding to the levels of membership from the nodes to each of the communities. Moreover, the proposed method can be applied to detect overlapping data items in a data set of general form, such as a vector-based data set, once it is transformed to a network. Usually, label propagation involves risks of error amplification. In order to avoid this problem, the proposed method offers a mechanism to identify outliers among the labeled data items, and consequently prevents error propagation from such outliers. Computer simulations carried out for synthetic and real-world data sets provide a numeric quantification of the performance of the method.

F. Breve · L. Zhao
Department of Computer Science, Institute of Mathematics and Computer Science (ICMC), University of São Paulo (USP)
Av. Trabalhador São-carlense, 400, 13560-970, São Carlos, SP, Brazil
Tel.: +55 16 3373-9713
Fax: +55 16 3371-2238
E-mail: {fabricio, zhao}@icmc.usp.br

Present address: of F. Breve
Department of Statistics, Applied Mathematics and Computation (DEMAC), Institute of Geosciences and Exact Sciences (IGCE), São Paulo State University (UNESP)
Avenida 24 A, 1515, 13506-900, Rio Claro, SP, Brazil
Tel.: +55 19 3526-9138
Fax: +55 19 3534-1511
E-mail: fabricio@rc.unesp.br

Keywords Graph-based method · community detection · particle competition and cooperation · overlapping nodes · outliers

1 Introduction

Over the last decade there has been an increased interest in network research, with the focus shifting away from the analysis of single small graphs to consideration of large-scale ones, called *complex networks*. Complex network based machine learning and data mining have triggered much attention. This is because such networks are ubiquitous in nature and everyday life. Many data sets are already represented by networks, such as the Internet, WWW, telecommunication networks, transportation networks, biological networks and social networks. Many other kinds of data sets can be transformed to network representations. For example, a table based relational data base can be transformed into a network by simply connecting the k nearest neighbors of each data point. One of the main motivations of graph theory research is the ability to describe topological structure of the original system. In machine learning domain, it has been shown that the topological structure is quite useful to detecting various cluster (class) forms by a data clustering (classification) algorithm with a unique distance measure (Karypis et al 1999; Fortunato 2010).

One of the striking phenomena of complex networks is the presence of *communities*. The notion of *communities* in networks is straightforward, each community is defined as a subgraph whose nodes are densely connected within itself but sparsely connected with the rest of the network. Therefore, community detection in networks has turned out to be an important topic in

data mining (Newman and Girvan 2004; Newman 2006; Duch and Arenas 2005; Reichardt and Bornholdt 2004; Danon et al 2005). In graph theory, community detection corresponds to graph partition, which has been shown to be a NP-complete problem (Fortunato 2010). For this reason, a lot of efforts have been paid to develop more efficient approximate solutions (See (Fortunato 2010) and references there in).

In practice, there are common cases where some nodes in a network can belong to two or more communities at the same time. For example, in a social network of friendship, individuals often belong to several communities: their families, their colleagues, their classmates, etc. These nodes are often called *overlapping nodes*, and few community detection algorithms can deal with this problem. Therefore, uncovering the overlapping community structure of complex networks is still an open problem (Zhang et al 2007a; Palla et al 2005; Zhang et al 2007b).

It is rare that we know nothing on a given data set. On the contrary, in many real world data sets, we know the labels of some elements. For example, one certainly does not know the majority of the people from Brazil, but we usually know some of them, such as Pelé, a famous soccer player, or Ayrton Senna, who was a famous racing driver. These labeled data, without a doubt, help to correctly determine the labels of the remaining ones. For this reason, we consider fuzzy community structure detection in a semi-supervised environment.

In this paper, we present a new community detection method, which uses competition and cooperation among particles walking in the network. It is inspired by the community detection method proposed by Quiles et al (2008), in which only hard labels can be produced. That model features walking particles in the network competing with each other in order to possess as many nodes as possible. Later, Breve et al (2011) extended that model to perform semi-supervised learning including not only competition among particles spreading different labels, but also cooperation among the particles which are spreading the same class label. However, it also provides only hard labels. Both Quiles et al (2008) and Breve et al (2011) provide analysis on time and storage complexity of these methods, showing that they have lower order of computational complexity than other unsupervised and semi-supervised graph based methods. While most semi-supervised graph-based methods have cubic complexity order ($O(n^3)$) (Zhu 2005), the methods proposed by Quiles et al (2008) and Breve et al (2011) have only linear complexity ($O(n)$), thus they can be applied to larger data sets.

A preliminary work to determine overlapping nodes by particle competition has been presented by Breve et al (2009). In that work, partial knowledge of networks is not taken into consideration, thus only the competition mechanism between particles is implemented. In this paper, we extend that model to semi-supervised learning case by introducing the cooperative mechanism through the concept of teams of particles. Particles in the same team cooperate with their teammates and compete against particles of other teams. We also transform the unsupervised learning mechanism into a semi-supervised learning mechanism, in order to take advantage of a small portion of labeled samples that usually are available in many real data sets. The proposed model produces a fuzzy output (soft label) for each node of the network. Such continuous-valued output can be treated as the levels of membership of each node to each community. Therefore, it is able to uncover the overlapping community structure in networks.

Another problem faced by machine learning algorithms is the presence of outliers or mislabeled samples in the training data. In practical applications, semi-supervised learning may entail risks because error propagation may be embedded in normal label propagation. For example, in a medical diagnostic system, a classification mistake may have serious consequences to a person's health. In semi-supervised learning domain, this situation gets worse because the classification mistake may propagate to the sub-system or even the whole system, resulting in wrong diagnosis of other cases. The model proposed in this paper gives special consideration to label propagation safety by introducing a mechanism to identify outliers among the labeled data items, and consequently prevent error propagation coming from outliers.

It is worth noting that many graph-based semi-supervised learning methods have been developed (Chapelle et al 2006). However, most of them are similar to each other (Zhu 2005) in such way that they can be seen as regularization frameworks, differing only in the particular choice of the loss function and the regularizer (Blum and Chawla 2001; Zhu et al 2003; Zhou et al 2004; Belkin et al 2004, 2005; Joachims 2003). Moreover, most of these methods spread the labels globally, i.e., at each iteration, all nodes propagate their labels to all other nodes accordingly to edges weights. The method proposed in this paper is essentially different from the others because the nature-inspired particle movement, competition and cooperation mechanisms allow it to spread the labels locally, at each algorithm step, i.e., only those nodes which are being visited by a particle are updated. Both the fuzzy output (overlapping community structure detection) and the outlier detection mechanisms are extracted naturally from the particle dynamics.

The rest of this paper is organized as follows: Section 2 describes the model in details. Section 3 shows some experimental results from computer simulations, and in Section 4 we draw some conclusions.

2 Model Description

In this section, we introduce the particle competition and cooperation algorithm. It takes a undirected and unweighed network as input. So, if the data set is already in that form, it can be used directly. Otherwise, the input data set must be transformed into an undirected and unweighed network. For each labeled data item, a corresponding particle is generated and put in the network. A group of particles having the same label is called a *team*. Each node in the network possesses a vector of elements, which corresponds to the domination level of each team of particles over that node. As the system runs, each particle uses a random-greedy rule to choose a neighbor to visit. In this chosen node, there is an increase of the domination level of the particle team, while there is a decrease of the domination levels of other teams. Teams of particles will act cooperatively trying to dominate as many nodes as possible while preventing intrusion of other teams. We keep track of each node visits and, at the end of iterations, we calculate the membership degrees of each node to each class by using the information of domination levels.

The input of the algorithm is a graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V} = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes and $\mathbf{E}$ is the set of edges (v_i, v_j), which can also be represented by an adjacency matrix $\mathbf{W}$:

$$W_{ij} = \begin{cases} 1 & \text{if } v_i \text{ and } v_j \text{ are neighbors} \\ 0 & \text{otherwise} \end{cases}, \qquad (1)$$

so W_{ij} specifies whether there is an edge between the pair of nodes v_i and v_j. The algorithm also requires a vector $Y = \{y_1, y_2, \ldots, y_n\}$, where y_i takes the hard label of the node v_i if it is known, or 0, otherwise. The label set is defined as $L = \{1, 2, \ldots, c\}$, where c is the amount of classes/communities, so a number is assigned to each class/community and 0 is reserved for nodes which label is unknown. If a labeled node is known to be an overlap node, its hard label is chosen after the class/community that it has the higher pertinence level. The goal of the algorithm is to provide a vector of membership degrees to each class for each of the nodes in the graph, no matter if they are initially labeled or unlabeled.

If the input data set is not a undirected unweighed network, it must be first transformed into one. For instance, if the input data set is vector-based, as in $\chi = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^m$, one may transform it into an undirected and unweighed graph by transforming each element x_i into a node v_i, and connecting it its k nearest neighbors according to some distance measure, like the Euclidean distance. In this case, the adjacency matrix $\mathbf{W}$ may be build as follows:

$$W_{ij} = \begin{cases} 1 & \text{if } x_j \text{ is among the } k\text{-nearest neighbors} \\ & \quad \text{of } x_i \text{ or vice-versa} \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

where W_{ij} specifies whether there is an edge between the pair of nodes x_i and x_j. Of course, one can also use faster methods to estimate the nearest neighbors, specially in larger data sets where the prior graph construction may be more time consuming than the algorithm iterations, as the algorithm has lower order of computational complexity than the prior graph construction step.

For each labeled node v_i (i.e., $y_i \neq 0$) in the network, a particle ρ_i is generated and its initial position is at the node v_i. Thus, there is a particle for each labeled node in the network. If v_i is the initial position of particle ρ_i, we call it the *home node* of particle ρ_i. At each iteration, each particle changes its position and register the distance it is from its home node. Particles generated for nodes with the same class/communities labels form a *team* and cooperate with each other to compete with other teams. So, each team represents a class/community of the network.

Each particle ρ_j comes with two variables: $\rho_j^{\omega}(t)$ and $\rho_{\mathbf{j}}^{\mathbf{d}}(\mathbf{t})$. The first variable $\rho_j^{\omega}(t) \in [0, 1]$ is the particle strength, which indicates how much the particle can affect a node levels at time t. The second variable is a distance table, i.e., a vector $\rho_{\mathbf{j}}^{\mathbf{d}}(\mathbf{t}) = \{\rho_i^{d_1}(t), \rho_i^{d_2}(t), \ldots, \rho_i^{d_n}(t)\}$, where each element $\rho_j^{d_i}(t) \in [0, \quad n-1]$ corresponds to the distance measured between the particle's home node v_j and its current position.

Each node v_i has two variables. The first variable is a vector $\mathbf{v_i^{\omega}}(\mathbf{t}) = \{v_i^{\omega_1}(t),\ v_i^{\omega_2}(t), \ldots, v_i^{\omega_c}(t)\}$ called *instantaneous domination levels*, and each element $v_i^{\omega_\ell}(t) \in [0, 1]$ corresponds to the level of domination of team ℓ over node v_i. At each node, the sum of the domination levels is always constant, as follows:

$$\sum_{\ell=1}^{c} v_i^{\omega_\ell} = 1. \qquad (3)$$

This relation is possible because particles increase the node domination level of their own team and, at the same time, decreases the other teams' domination levels. The second variable is the *long term domination levels*, which is a vector $\mathbf{v_i^{\lambda}}(\mathbf{t}) = \{v_i^{\lambda_1}(t), v_i^{\lambda_2}(t), \ldots, v_i^{\lambda_c}(t)\}$, and each element $v_i^{\lambda_\ell}(t) \in [0 \quad \infty]$ represents long term domination level by team ℓ over node v_i.

Long term domination levels can vary from zero to infinity and they never decrease.

Each node v_i has an initial value of its instantaneous domination vector $\mathbf{v_i^\omega}$ set as follows:

$$v_i^{\omega_\ell}(0) = \begin{cases} \frac{1}{c} & \text{if } y_i = 0 \\ 1 & \text{if } y_i = \ell \\ 0 & \text{otherwise} \end{cases}, \tag{4}$$

i.e., for each unlabeled node ($y = 0$), the domination levels of all particle teams are set to the same value $\frac{1}{c}$, where c is the number of classes/communities (number of teams); and for each labeled node ($y \neq 0$), the domination level of the dominating team is set to the highest value 1, while the domination levels of other teams are set to the lowest value 0. On the other hand, in all nodes, all long term domination levels $\mathbf{v_i^{\lambda_\ell}}(\mathbf{0})$ have their initial values set to zero, for all the classes ℓ no matter if the corresponding data item is labeled or unlabeled.

Each particle has its initial position set to the corresponding home node, and their initial strength is set as follows:

$$\rho_j^\omega(0) = 1, \tag{5}$$

i.e., each particle starts with maximum strength.

Particles have limited knowledge of the network, they only know the distances from their home node to nodes that they already visited. Distances are recalculated dynamically at each particle movement. Thus, the distance table of each particle is set as follows:

$$\rho_j^{d_i}(t) = \begin{cases} 0 & \text{if } \; i = j \\ n-1 & \text{if } \; i \neq j \end{cases}, \tag{6}$$

i.e., for each particle, the distance from its home node is set to zero, and all the other distances are assumed to be the largest possible value $n - 1$

At each iteration, each particle will select a neighbor to visit. There are two different kinds of movements a particle can use: *random movement* and *greedy movement*. During *random movement*, a particle randomly chooses any neighbor to visit without concerning domination levels or distance from its home node. This movement is used for exploration and acquisition of new nodes. Meanwhile, in *greedy movement*, each particle prefers visiting those nodes that have been already dominated by its own team and that are closer to their home nodes. This movement is used for defense of both its own and its team's territories. In order to achieve an equilibrium between exploratory and defensive behavior both movements are applied. Therefore, at each iteration, each particle has probability p_{grd} to choose greedy movement and probability $1 - p_{\text{grd}}$ to choose random movement, with $0 \leq p_{\text{grd}} \leq 1$. Once the random movement or greedy movement is determined, the target neighbor node $\rho_j^\tau(t)$ will be chosen with probabilities defined by Eq. (7) or Eq. (8), respectively.

In *random walk* the particle ρ_j tries to move to any node v_i with the probabilities defined as:

$$p(v_i|\rho_j) = \frac{W_{qi}}{\sum_{\mu=1}^{n} W_{q\mu}}, \tag{7}$$

where q is the index of the current node of particle ρ_j, so $W_{qi} = 1$ if there is an edge between the current node and any node v_i, and $W_{qi} = 0$ otherwise.

In *greedy movement* the particle tries to move to a neighbor with probabilities defined according to its team domination level on that neighbor $\rho_j^{\omega_\ell}$ and the inverse of the distance ($\rho_j^{d_i}$) from that neighbor v_i to its home node v_j as follows:

$$p(v_i|\rho_j) = \frac{W_{qi} v_i^{\omega_\ell} (\rho_j^{d_i} + 1)^{-2}}{\sum_{\mu=1}^{n} W_{q\mu} v_\mu^{\omega_\ell} (\rho_j^{d_\mu} + 1)^{-2}}. \tag{8}$$

Once more, q is the index of the current node of particle ρ_j and $\ell = \rho_j^f$, where ρ_j^f is the class label of particle ρ_j.

Particles of different teams compete for owning the network nodes, when a particle moves to another node, it increases the instantaneous domination level of its team in that node, at the same time it decreases the instantaneous domination level of the other teams in that same node. The exception are the labeled nodes, which instantaneous domination levels are fixed. Thus, for each selected target node v_i, the instantaneous domination level $v_i^{\omega_\ell}(t)$ is updated as follows:

$$v_i^{\omega_\ell}(t+1) = \begin{cases} \max\{0, v_i^{\omega_\ell}(t) - \frac{\Delta_v \rho_j^\omega(t)}{c-1}\} \\ \quad \text{if } y_i = 0 \text{ and } \ell \neq \rho_j^f \\ v_i^{\omega_\ell}(t) + \sum_{q \neq \ell} v_i^{\omega_q}(t) - v_i^{\omega_q}(t+1) \\ \quad \text{if } y_i = 0 \text{ and } \ell = \rho_j^f \\ v_i^{\omega_\ell}(t) \quad \text{if } y_i \neq 0 \end{cases}, \tag{9}$$

where $0 < \Delta_v \leq 1$ is a parameter to control changing rate of the instantaneous domination levels and ρ_j^f represents the class label of particle ρ_j. If Δ_v takes a low value, the node instantaneous domination levels change slowly, while if it takes a high value, the node domination levels change quickly. Each particle ρ_j increases the instantaneous domination level of its team ($v_i^{\omega_\ell}$, $\ell = \rho_j^f$) at the node v_i when it moves to it, while it decreases the instantaneous domination levels of this same node of other teams ($v_i^{\omega_\ell}$, $\ell \neq \rho_j^f$), always respecting the conservation law defined by Eq. (3). The instantaneous domination level of all labeled node v_i^ω are always fixed, as defined by the third case expressed by Eq. (9).

Regarding long term domination levels, at each iteration, for each selected node v_i in *random movement*,

the long term domination level $v_i^{\lambda_\ell}(t)$ is updated as follows::

$$v_i^{\lambda_\ell}(t+1) = v_i^{\lambda_\ell}(t) + \rho_j^{\omega}(t) \tag{10}$$

where ℓ is the class label of particle ρ_j. Eq. (10) shows that the updating of the long term domination levels $v_i^{\lambda_\ell}(t+1)$ is proportional to the current particle strength $\rho_j^{\omega}(t)$. This is a desirable feature because the particle probably has a higher strength when it is arriving from its own neighborhood, while it has a lower strength when it is arriving from nodes from other teams neighborhoods. When *greedy movement* is selected, long term domination levels are not updated, otherwise a team domination would be amplified by greedy movement visits, which is not desirable as it would introduce bias in the fuzzy output.

Regarding particles strength, they get stronger when they move to a node being dominated by its own team and they get weaker when they move to a node dominated by other teams. Thus, at each iteration t, a particle strength $\rho_j^{\omega}(t)$ is updated as follows:

$$\rho_j^{\omega}(t+1) = v_i^{\omega_\ell}(t+1), \tag{11}$$

where v_i is the target node, and $\ell = \rho_j^f$, i.e., ℓ is the class label of particle ρ_j. Therefore, each particle ρ_j has its strength ρ_j^{ω} set to the value of its team instantaneous domination level $v_i^{\omega_j}$ of the node v_i.

It is important to notice that when a particle moves, it may be accepted or rejected in the target node due to the competition mechanism. First, a particle modifies both the node instantaneous and long term domination levels as explained, then it updates its own strength, and finally it will be accepted in the new node only if the domination level of its team is higher than others; otherwise, a shock happens and the particle goes back to the last node until next iteration.

The distance table purpose is to keep the particle aware of how far it is from its home node. This information is used in the *greedy movement* in order to keep the particle around its own neighborhood most of the time, avoiding letting it susceptible to be attacked by other teams. The instantaneous domination levels together with the distance information also avoid situations where a particle would walk into enemies' neighborhoods and lose all its strength. Each particle ρ_j updates its distance table $\rho_j^{d_k}(t)$ at each iteration t as follows:

$$\rho_j^{d_k}(t+1) = \begin{cases} \rho_j^{d_i}(t) + 1 \text{ if } \rho_j^{d_i}(t) + 1 < \rho_j^{d_k}(t) \\ \rho_j^{d_k}(t) \quad \text{otherwise} \end{cases}, \tag{12}$$

where $\rho_j^{d_i}(t)$ and $\rho_j^{d_k}(t)$ are the distances to its home node from the current node and the target node, respectively.

The distance calculation works as follows: we assume that the particles initially have limited knowledge of the network, i.e., they know how many nodes in the network, but they do not know how the nodes are connected, so they assume all the nodes can be reached in at most $n-1$ steps (the largest possible distance). Every time a particle moves, it checks the current distance table. If the target node distance is higher than the current node distance, the target node distance is updated to the distance of the current node plus 1. This method has advantage to use already known distances without recalculation.

In a first glance, the nodes' instantaneous domination levels $\mathbf{v_i^{\omega}(t)}$ looks like a natural choice for nodes' fuzzy (gradual) outputs, since they indicate the domination levels from each team (class) to each node quantified in terms of continuous values in $[0, 1]$. However, instantaneous domination levels are very volatile under certain conditions. For instance, the dominating team of a non-overlapping node after the last iteration usually owns the node for all or majority of iterations, but this may not happen to overlapping nodes, in which the dominating team changes frequently, and thus the instantaneous domination level of the last dominating team may not correspond to the team which have dominated that node for longer time. Also, due to the competition effect, the instantaneous domination level of the dominating team is largely amplified and it does not correspond to the real overlapping level. And that is why we have the long term domination levels, which represents temporal averaged domination level for each team at each node. In this case, when a team's long term domination level is increased, long term domination levels of other teams are kept without changes. Also, there is no upper limit on long term domination levels, i.e., they can vary from zero to infinity. At the end of iterations, the fuzzy output is derived from the long term domination levels. It is important to note that the long term domination levels are adjusted only when a particle selects the *random movement*, because, like the competition effect, the *greedy movement* amplifies visiting advantage of dominating particle.

After the last iteration, the degrees of membership $f_i^{\ell} \in [0 \quad 1]$ corresponding to each node v_i are calculated using the long term domination levels, as follows:

$$f_i^{\ell} = \frac{v_i^{\lambda_\ell}(\infty)}{\sum_{q=1}^{c} v_i^{\lambda_q}(\infty)} \tag{13}$$

where f_i^{ℓ} represents the final membership level from the node v_i to community ℓ.

Based on the membership degrees (fuzzy output), we have formed an overlapping measure in order to

easily illustrate the application of the algorithm. Therefore, the overlapping index o_i for a node v_i is defined as follow:

$$o_i = \frac{f_i^{\ell**}}{f_i^{\ell*}} \tag{14}$$

where $\ell* = \arg\max_\ell f_i^\ell$, $\ell** = \arg\max_{\ell,\ell\neq\ell*} f_i^\ell$, and $o_i \in [0 \quad 1]$, where $o_i = 0$ means completely confidence that the node belongs to a single community, while $o_i = 1$ means the node is completely undefined being shared among two or more communities.

If needed, hard labels may be easily obtained through the following equation:

$$y_i = \arg\max_\ell f_i^\ell, \tag{15}$$

i.e., the node is hard labeled after the class with the highest membership level. These hard labels may be very different from those obtained from instantaneous domination levels, like in (Breve et al 2011). They are usually more accurate to classify outliers and nodes around outliers. They may also be used to accurately reclassify wrongly labeled nodes, as instantaneous domination levels are always fixed and, therefore, cannot be used to this purpose.

Overall, the proposed algorithm can be outlined as follows:

Algorithm 1: Particle Competition and Cooperation

```
Set nodes' domination levels by using Eq. (4);
Set initial positions of particles at their corresponding home nodes by using Eq. (5);
Set particle strength and distance tables by using Eq. (6);
for a pre-determined amount of iterations do
    for each particle do
        Select between random or greedy rule with probability defined by p_grd;
        Select the target node by using Eq. (7) or Eq. (8) for random or greedy movement respectively;
        Update target node instantaneous domination levels by using Eq. (9);
        Update target node long term domination levels by using Eq. (10);
        Update particle strength by using Eq. (11);
        Update particle distance tables by using Eq. (12);
Calculate the membership levels for each data item using Eq. (13);
Calculate the overlapping index for each data item using Eq. (14);
If needed, assign a hard label to each data item using Eq. (15);
```

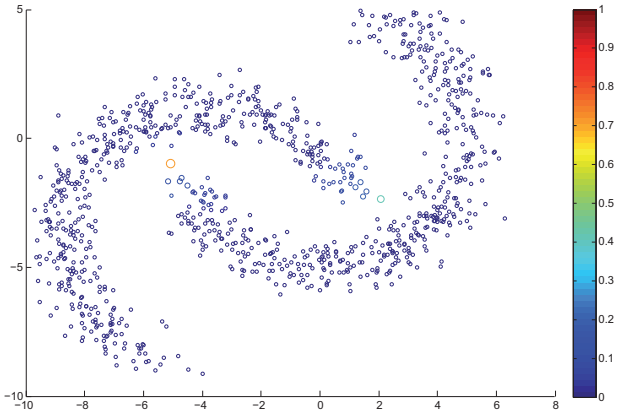
3 Computer Simulations

In this section, we present some simulation results to evaluate the effectiveness of the proposed method. First, the proposed algorithm is applied to artificial data sets with increasing amount of overlapped nodes. Then, the robustness to incorrectly labeled nodes is demonstrated, including the reclassification of these nodes. Next, the proposed algorithm is applied to some real-world data sets, including both network-based and vector-based data sets. Finally, the algorithm is evaluated with the benchmark for undirected and unweighted networks with overlapping communities proposed by Lancichinetti and Fortunato (2009a) (LFR benchmark), in order to make it easier to compare it to other methods.

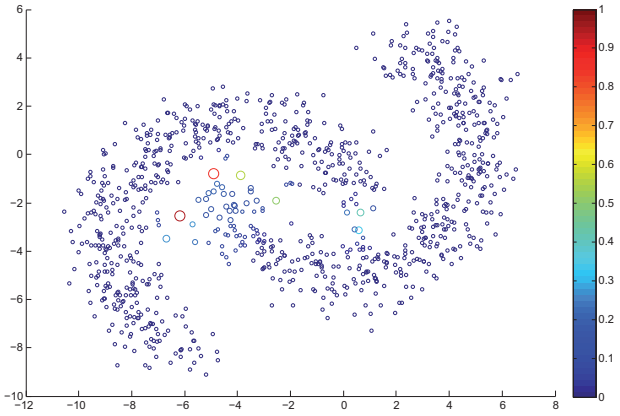
For all vector-based data sets, the networks are built by using Eq. (2), with the parameter k being empirically set for each problem, i.e., a set of simulations is executed by varying k in the interval $[0.01n \quad 0.1n]$, and the value leading to the best results is chosen. The Euclidean distance is used in all cases. The algorithm parameters are also empirically set to $\Delta_v = 0.1$ and $p_{\text{grd}} = 0.5$ for all the experiments. All results shown in this section are the average of 50 to $1,000$ executions.

Figures 1a, 1b, and 1c show the results of the proposed method applied to three banana-shaped classes generated using PRTools (Duin et al 2007) function `gendatb` with $1,000$ elements each (500 per class) and different variance parameter $s = \{0.6,\ 0.8, 1.0\}$. For each data set, 50 elements (5%) were randomly selected as the labeled ones. The size of the nodes in the plot are proportional to their respective overlapping index. We see that there are more overlapping nodes and the overlapping levels are higher as the classes get more mixed. This situation matches well the results we obtain through a direct visual inspection. These experiments are repeated 100 times. The mean standard deviation of the membership levels are 0.0216, 0.0222, and 0.0598, for Figures 1a, 1b, and 1c, respectively. After assigning the hard labels, the average correct classification rates are 0.9945, 0.9923 and 0.9607, respectively; and the standard deviations are 0.0108, 0.0140, and 0.0164, respectively.

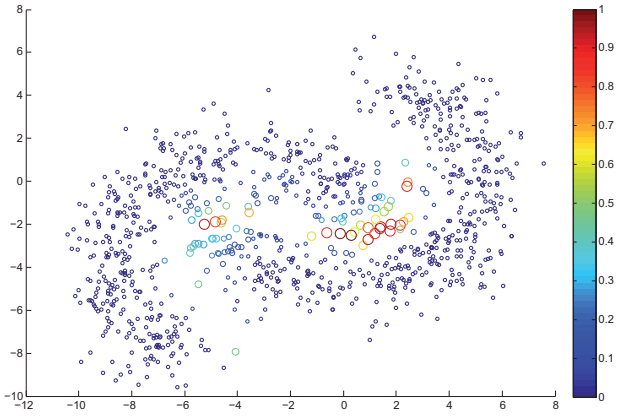
Figure 2a shows a data set with 4 classes with Gaussian distribution, generated by using PRTools (Duin et al 2007) function `gendats` with $1,000$ elements (250 per class) and 20 samples are labeled (5 per class), represented by the *squares*, *triangles*, *lozenges* and *stars*. The algorithm is applied to the data set and the detected overlapping indexes are shown in Fig. 2b. We see that the nodes in the interior of each class are small, i.e., they are clearly non-overlapping nodes. Meanwhile, the nodes in the borders among classes have larger sizes,



(a)

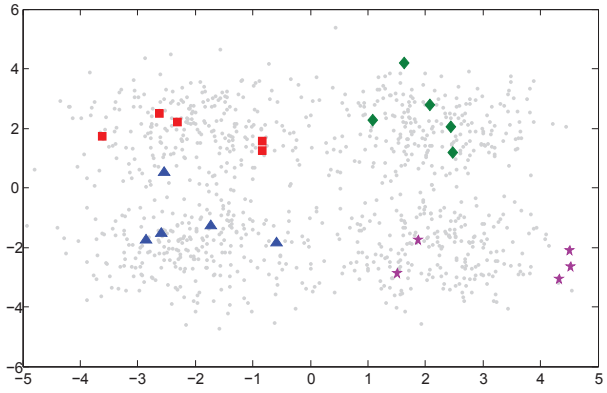


(b)

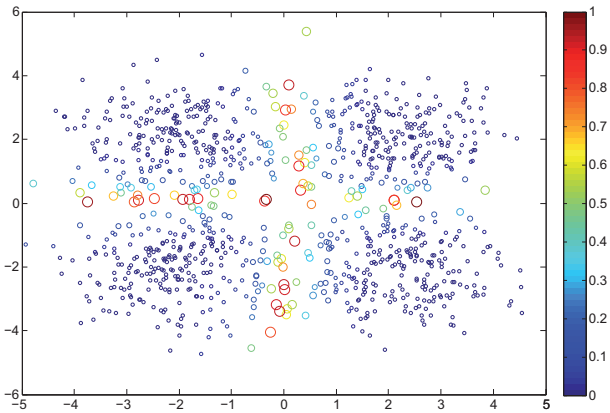


(c)

Fig. 1: Fuzzy classification of two banana-shaped classes generated with different variance parameters: (a) $s = 0.6$ (b) $s = 0.8$ (c) $s = 1.0$. Nodes size and colors represent their respective overlapping index detected by the proposed method.



(a)



(b)

Fig. 2: Classification of normally distributed classes (Gaussian distribution). (a) toy data set with $1,000$ samples divided in four classes, 20 samples are labeled, 5 from each class (squares, triangles, lozenges and stars). (b) nodes size represent their respective overlapping index detected by the proposed method.

which represent their different levels of overlapping. These results are in agreement with our intuition. These experiments are repeated 100 times and the mean standard deviation of the membership levels is only 0.0037. For the hard labels, the average correct classification rate is 0.9546 and the standard deviation is 0.0012.

Referring to Fig. 2a, we note that there is an upper triangle in the space of the square class, it is clearly an outlier. However, it does not mix up the overlapping indexes of the nodes around it. It means that a particle which home node is an outlier has difficulty to defend its neighborhood, since it may be far from its team-mates and receives few or no help from them. A particle whose home node is an outlier can eventually abandon its home, if its neighborhood is dominated by another team. In this case, it may migrate to the neighborhood of one of its nearby team-mates. Although an

outlier can eventually change a little bit of the instantaneous domination levels ($v_i^{\varpi}(t)$) of its neighbors, it has very weak effect to the long term domination levels ($v_i^{\lambda}(t)$) of these same neighbors. Thus, we can achieve good classification results even though the data sets have some outliers. Notice that the instantaneous domination levels are fixed for labeled nodes, but the long term domination levels are not. Thus, through the long term domination levels, a labeled node can be reclassified if it is an outlier using Eq. (15).

In order to show these features, we perform simulations on an artificial data set presented by Fig. 3a, it has 2,000 elements distributed into two banana-shaped classes (1,000 elements per class), 100 (5%) of them are labeled (circles and squares), however, 10 of these labeled nodes have the wrong label representing outliers. Figure 3b shows the classification by the proposed method. The hard labels are obtained through Eq. (15), i.e, the sample is simply classified to the class with the highest membership level. We can see that the wrongly labeled nodes do not affect the classification of their neighbors and the outliers themselves are eventually reclassified to their respective proper classes. These experiments are repeated 100 times, the average correct classification rate is 0.9975 and the standard deviation is only 0.0001.

Next, the proposed algorithm is applied to a network-based real-world data set: the Zachary's Karate Club Network (Zachary 1977), which is already an undirected and unweighed network, so the prior graph construction step is not needed. The data set is presented to the algorithm with only two labeled nodes: 1 and 34, each one representing a different class. The results are shown in Fig. 4, and the overlapping index of each node is indicated by their sizes. Our visual inspection indicates that this is a good result as well. Notice that although the two labeled nodes exhibit some degree of overlapping, the algorithm still produced a good result, even detecting these overlapping degrees in the labeled nodes (notice the slightly larger size). This is also a desirable feature, since we do not need to choose a non-overlapping node to represent a class. The three most overlapping nodes detected by the proposed algorithm are nodes 9, 3, and 20, which matches the results obtained by Zhang et al (2007b). Notice that these results are the mean of 1000 executions, and the standard deviation of the continuous output is only 0.003, i.e., the method output is pretty consistent. By applying Eq. (15), the hard labels are obtained and the algorithm achieves a perfect classification score (100% correct classification rate of the 34 nodes) in all the 1000 repetitions.

As the next step, the proposed method is applied to two vector-based real-world data sets from the UCI Ma-

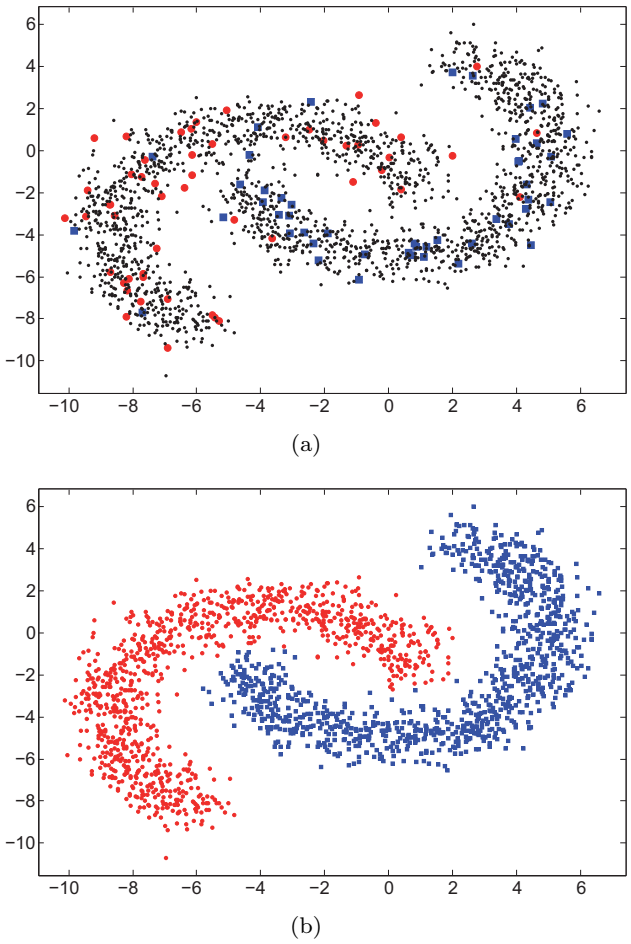


Fig. 3: Classification of data sets with some outliers: (a) artificial data set with some wrongly labeled nodes (b) classification by the proposed method

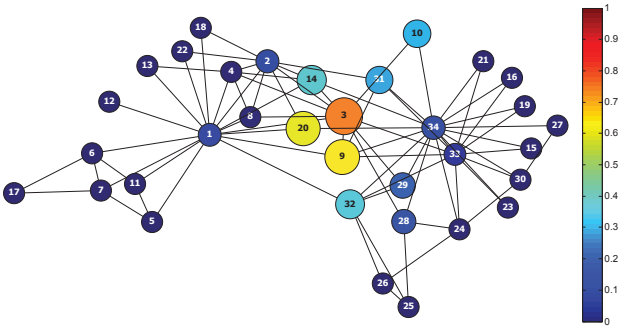


Fig. 4: The Karate Club Network. Nodes size and colors represent their respective overlapping index detected by the proposed method.

chine Learning Repository (Frank and Asuncion 2010): the Iris Data Set and the Wine Data Set. The Iris Data Set has 150 samples of 3 different types of Iris flower: Iris Setosa, Iris Versicolour and Iris Virginica. The first class (Iris Setosa) is quite different from the other two; while the latter are more similar and hard to separate from each other. There are 50 samples in each class and 4 real-valued attributes for each sample. The network is built from the data using (2) with $k = 5$, which was empirically set. We randomly select 10% of the samples to be presented to the algorithm with their respective labels, while the remaining are presented unlabeled. The degrees of membership from each sample to each class obtained by the proposed algorithm are presented in Table 1. A graph representation is showed in Fig. 5, in which the overlapping index of each node is indicated by their sizes. Notice that the linearly separable class becomes a disconnected subset of the graph nodes, and the membership degrees of these samples are complete to the respective class and zero for the others, as expected for clearly non-overlapping nodes. The other two classes are not linearly separable as they have some degree of overlapping. The different overlapping degrees of all these nodes and their respective pertinence to each of the classes are detected by the proposed algorithm. These experiments are repeated $1,000$ times and the mean standard deviation of the membership levels is 0.0803. Regarding the hard labels, the average correct classification rate is 0.9375 and the standard deviation is 0.0458.

Table 1: Degrees of membership from each sample to each class obtained by the proposed method for the Iris Data Set.

Inst.	I. Set	I. Vers.	I. Virg.	Inst.	I. Set	I. Vers.	I. Virg.	Inst.	I. Set	I. Vers.	I. Virg.
1	1,0000	0,0000	0,0000	51	0,0000	0,9252	0,0748	101	0,0000	0,0084	0,9916
2	1,0000	0,0000	0,0000	52	0,0000	0,9245	0,0755	102	0,0000	0,2505	0,7495
3	1,0000	0,0000	0,0000	53	0,0000	0,8756	0,1244	103	0,0000	0,0072	0,9928
4	1,0000	0,0000	0,0000	54	0,0000	0,9688	0,0312	104	0,0000	0,0275	0,9725
5	1,0000	0,0000	0,0000	55	0,0000	0,9251	0,0749	105	0,0000	0,0111	0,9889
6	1,0000	0,0000	0,0000	56	0,0000	0,9569	0,0431	106	0,0000	0,0071	0,9929
7	1,0000	0,0000	0,0000	57	0,0000	0,7678	0,2322	107	0,0000	0,9384	0,0616
8	1,0000	0,0000	0,0000	58	0,0000	0,9769	0,0231	108	0,0000	0,0071	0,9929
9	1,0000	0,0000	0,0000	59	0,0000	0,9291	0,0709	109	0,0000	0,0243	0,9757
10	1,0000	0,0000	0,0000	60	0,0000	0,9492	0,0508	110	0,0000	0,0072	0,9928
11	1,0000	0,0000	0,0000	61	0,0000	0,9769	0,0231	111	0,0000	0,1042	0,8958
12	1,0000	0,0000	0,0000	62	0,0000	0,9784	0,0216	112	0,0000	0,0450	0,9550
13	1,0000	0,0000	0,0000	63	0,0000	0,9847	0,0153	113	0,0000	0,0118	0,9882
14	1,0000	0,0000	0,0000	64	0,0000	0,8696	0,1304	114	0,0000	0,2545	0,7455
15	1,0000	0,0000	0,0000	65	0,0000	0,9919	0,0081	115	0,0000	0,2492	0,7508
16	1,0000	0,0000	0,0000	66	0,0000	0,9300	0,0700	116	0,0000	0,0293	0,9707
17	1,0000	0,0000	0,0000	67	0,0000	0,9466	0,0534	117	0,0000	0,0298	0,9702
18	1,0000	0,0000	0,0000	68	0,0000	0,9924	0,0076	118	0,0000	0,0071	0,9929
19	1,0000	0,0000	0,0000	69	0,0000	0,7237	0,2763	119	0,0000	0,0071	0,9929
20	1,0000	0,0000	0,0000	70	0,0000	0,9792	0,0208	120	0,0000	0,3700	0,6300
21	1,0000	0,0000	0,0000	71	0,0000	0,5359	0,4641	121	0,0000	0,0078	0,9922
22	1,0000	0,0000	0,0000	72	0,0000	0,9924	0,0076	122	0,0000	0,2488	0,7512
23	1,0000	0,0000	0,0000	73	0,0000	0,4285	0,5715	123	0,0000	0,0071	0,9929
24	1,0000	0,0000	0,0000	74	0,0000	0,9488	0,0512	124	0,0000	0,2812	0,7188
25	1,0000	0,0000	0,0000	75	0,0000	0,9431	0,0569	125	0,0000	0,0104	0,9896
26	1,0000	0,0000	0,0000	76	0,0000	0,9318	0,0682	126	0,0000	0,0071	0,9929
27	1,0000	0,0000	0,0000	77	0,0000	0,8878	0,1122	127	0,0000	0,2847	0,7153
28	1,0000	0,0000	0,0000	78	0,0000	0,5506	0,4494	128	0,0000	0,3837	0,6163
29	1,0000	0,0000	0,0000	79	0,0000	0,9593	0,0407	129	0,0000	0,0220	0,9780
30	1,0000	0,0000	0,0000	80	0,0000	0,9862	0,0138	130	0,0000	0,0072	0,9928
31	1,0000	0,0000	0,0000	81	0,0000	0,9731	0,0269	131	0,0000	0,0071	0,9929
32	1,0000	0,0000	0,0000	82	0,0000	0,9752	0,0248	132	0,0000	0,0071	0,9929
33	1,0000	0,0000	0,0000	83	0,0000	0,9937	0,0063	133	0,0000	0,0184	0,9816
34	1,0000	0,0000	0,0000	84	0,0000	0,2363	0,7637	134	0,0000	0,2480	0,7520
35	1,0000	0,0000	0,0000	85	0,0000	0,9469	0,0531	135	0,0000	0,1000	0,9000
36	1,0000	0,0000	0,0000	86	0,0000	0,8618	0,1382	136	0,0000	0,0071	0,9929
37	1,0000	0,0000	0,0000	87	0,0000	0,9015	0,0985	137	0,0000	0,0160	0,9840
38	1,0000	0,0000	0,0000	88	0,0000	0,8396	0,1604	138	0,0000	0,0294	0,9706
39	1,0000	0,0000	0,0000	89	0,0000	0,9782	0,0218	139	0,0000	0,4552	0,5448
40	1,0000	0,0000	0,0000	90	0,0000	0,9541	0,0459	140	0,0000	0,0148	0,9852
41	1,0000	0,0000	0,0000	91	0,0000	0,9496	0,0504	141	0,0000	0,0084	0,9916
42	1,0000	0,0000	0,0000	92	0,0000	0,9450	0,0550	142	0,0000	0,0246	0,9754
43	1,0000	0,0000	0,0000	93	0,0000	0,9861	0,0139	143	0,0000	0,2506	0,7494
44	1,0000	0,0000	0,0000	94	0,0000	0,9769	0,0231	144	0,0000	0,0075	0,9925
45	1,0000	0,0000	0,0000	95	0,0000	0,9675	0,0325	145	0,0000	0,0080	0,9920
46	1,0000	0,0000	0,0000	96	0,0000	0,9856	0,0144	146	0,0000	0,0254	0,9746
47	1,0000	0,0000	0,0000	97	0,0000	0,9654	0,0346	147	0,0000	0,2266	0,7734
48	1,0000	0,0000	0,0000	98	0,0000	0,9725	0,0275	148	0,0000	0,0857	0,9143
49	1,0000	0,0000	0,0000	99	0,0000	0,9788	0,0212	149	0,0000	0,0257	0,9743
50	1,0000	0,0000	0,0000	100	0,0000	0,9853	0,0147	150	0,0000	0,2961	0,7039

The Wine Data Set has 178 samples of 3 different types of Wine. There are 59, 71 and 48 samples in classes 1, 2, and 3, respectively. Each sample has 13 integer and real-valued attributes. The network is also built from the data using (2) with $k = 5$. Once more, we randomly select 10% of the samples to be presented to the algorithm with their respective labels, while the remaining are presented unlabeled. The degrees of membership from each sample to each class obtained by the proposed algorithm are presented in Table 2; and a graph representation is showed in Fig. 6, in which the overlapping index of each node is indicated by their sizes. By analyzing the results, we can notice that there are some overlapping nodes between classes 1 and 2; and some more between classes 2 and 3. On the other hand, classes 1 and 3 are almost completely separated. These experiments are repeated $1,000$ times and the mean standard deviation of the membership levels is 0.0794. Regarding the hard labels, the average correct classification rate is 0.9326 and the standard deviation is 0.0364.

Finally, the proposed method is evaluated with the benchmark for undirected and unweighted networks with overlapping communities proposed by Lancichinetti and Fortunato (2009a) (LFR benchmark). This benchmark uses the normalized mutual information between the planted and the recovered partition, in its generalized form for overlapping communities proposed by Lancichinetti et al (2009), as the performance measure. The benchmark method does not use the overlap degrees, only hard labels. Therefore, we adapted the proposed method to produce one single hard label using Eq. (15) when the overlap index is low ($o_i \leq 0.5$), and two hard labels when the overlap index is high ($o_i > 0.5$). In this last case, the first hard label is given by Eq. (15), and the second hard label (y_i^{**}) is given by:

$$y_i^{**} = \arg \max_{\ell, \ell \neq y_i} f_i^{\ell}, \tag{16}$$

i.e., the node second hard label is assigned after the team with the second largest domination level on that node. In all cases, 10% of the nodes are randomly chosen to be presented to the algorithm with their respective label.

The proposed method benchmark output is shown in Figures 7 and 8. Each data point in these figures

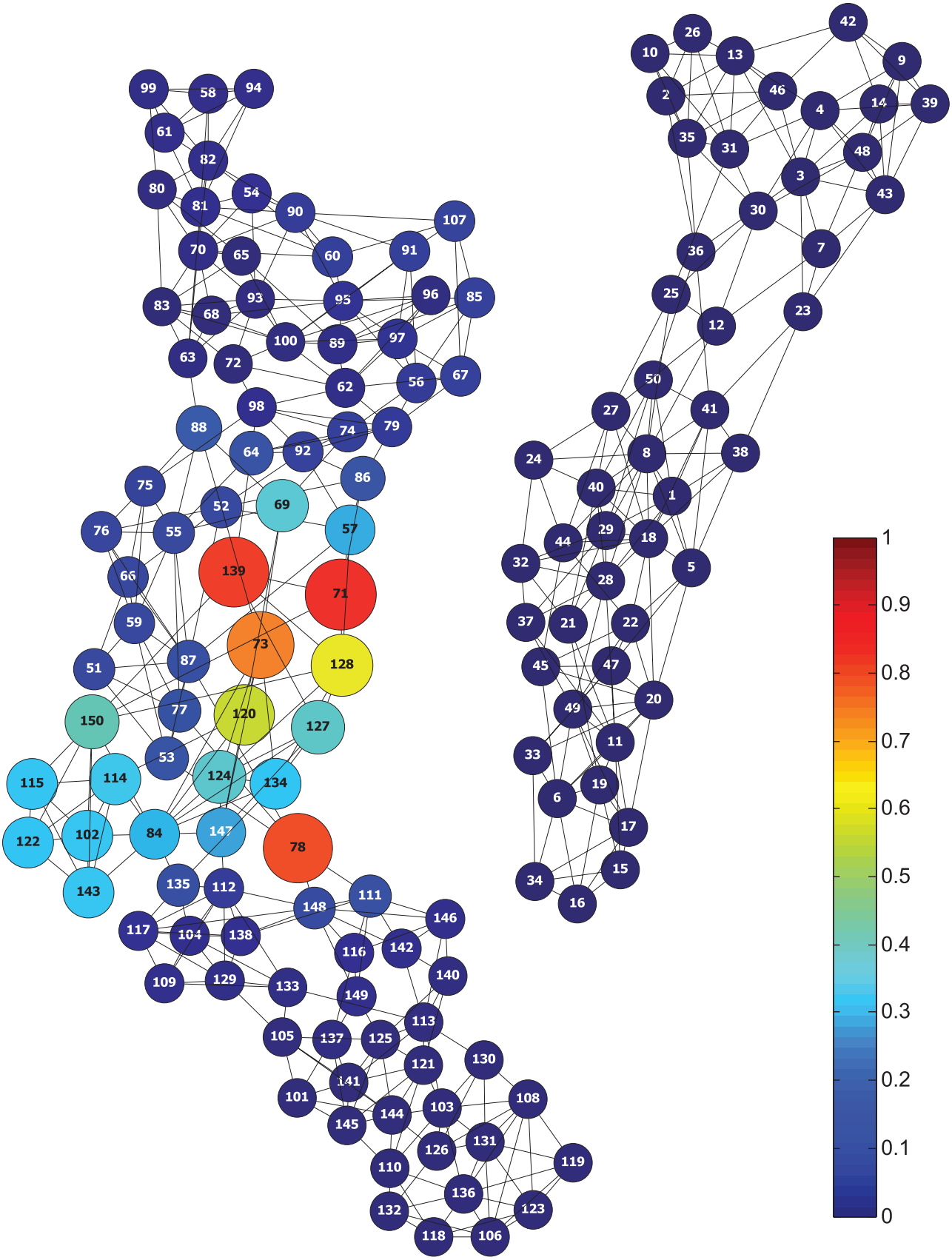


Fig. 5: The Iris Data Set. Nodes size and colors represent their respective overlapping index detected by the proposed method.

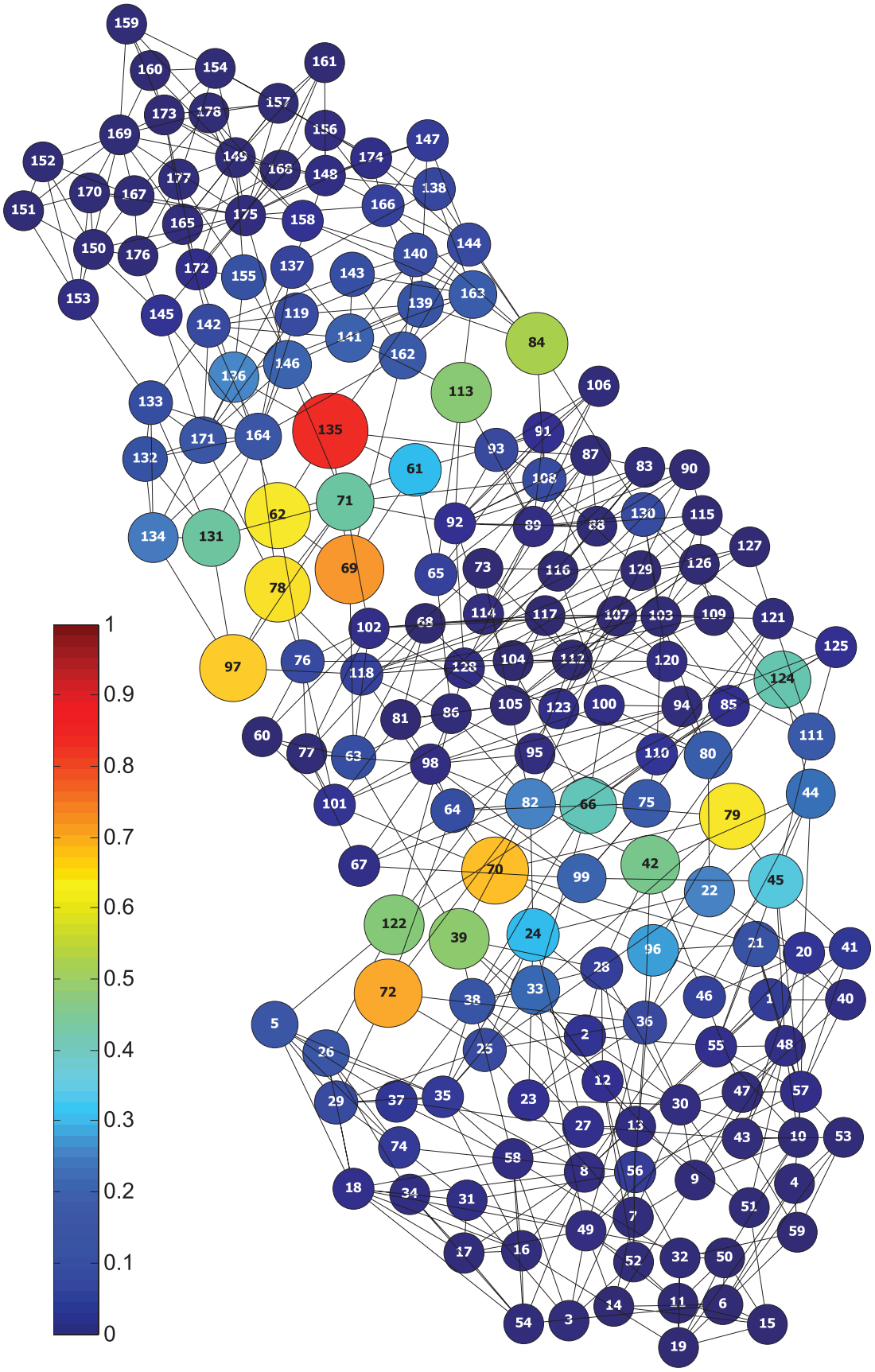


Fig. 6: The Wine Data Set. Nodes size and colors represent their respective overlapping index detected by the proposed method.

Table 2: Degrees of membership from each sample to each class obtained by the proposed method for the Wine Data Set.

Inst.	Class 1	Class 2	Class 3	Inst.	Class 1	Class 2	Class 3	Inst.	Class 1	Class 2	Class 3
1	0,9519	0,0480	0,0002	61	0,0007	0,7603	0,2390	121	0,0103	0,9886	0,0011
2	0,9656	0,0344	0,0000	62	0,0007	0,6056	0,3937	122	0,3327	0,6665	0,0008
3	0,9918	0,0082	0,0000	63	0,0928	0,8615	0,0457	123	0,0188	0,9800	0,0012
4	0,9951	0,0049	0,0000	64	0,0907	0,9075	0,0018	124	0,3004	0,6992	0,0003
5	0,8581	0,1418	0,0000	65	0,0008	0,9466	0,0525	125	0,0288	0,9706	0,0006
6	0,9969	0,0030	0,0001	66	0,2946	0,7048	0,0007	126	0,0017	0,9974	0,0009
7	0,9922	0,0078	0,0000	67	0,0212	0,9770	0,0019	127	0,0024	0,9964	0,0012
8	0,9866	0,0129	0,0005	68	0,0009	0,9983	0,0008	128	0,0161	0,9802	0,0037
9	0,9944	0,0056	0,0000	69	0,0008	0,5794	0,4198	129	0,0009	0,9971	0,0020
10	0,9937	0,0063	0,0000	70	0,4067	0,5930	0,0003	130	0,0138	0,9002	0,0861
11	0,9976	0,0024	0,0000	71	0,0006	0,6912	0,3082	131	0,0010	0,3101	0,6889
12	0,9720	0,0280	0,0000	72	0,5857	0,4135	0,0008	132	0,0010	0,1112	0,8878
13	0,9847	0,0153	0,0000	73	0,0008	0,9973	0,0018	133	0,0008	0,0904	0,9088
14	0,9970	0,0030	0,0000	74	0,9584	0,0416	0,0000	134	0,0010	0,2033	0,7957
15	0,9977	0,0023	0,0000	75	0,1593	0,8391	0,0015	135	0,0005	0,4696	0,5299
16	0,9900	0,0095	0,0006	76	0,0010	0,9206	0,0785	136	0,0005	0,2124	0,7870
17	0,9921	0,0073	0,0006	77	0,0044	0,9932	0,0024	137	0,0004	0,0608	0,9388
18	0,9752	0,0244	0,0004	78	0,0008	0,6035	0,3957	138	0,0000	0,0689	0,9311
19	0,9969	0,0029	0,0002	79	0,6053	0,3946	0,0000	139	0,0003	0,1273	0,8724
20	0,9648	0,0352	0,0000	80	0,1655	0,8342	0,0003	140	0,0000	0,0849	0,9151
21	0,8798	0,1202	0,0000	81	0,0020	0,9974	0,0006	141	0,0002	0,1748	0,8251
22	0,7910	0,2090	0,0000	82	0,2093	0,7899	0,0008	142	0,0006	0,0761	0,9233
23	0,9694	0,0306	0,0000	83	0,0007	0,9942	0,0051	143	0,0002	0,0968	0,9030
24	0,7613	0,2383	0,0004	84	0,0001	0,3549	0,6451	144	0,0001	0,0823	0,9176
25	0,9177	0,0822	0,0001	85	0,0274	0,9719	0,0007	145	0,0002	0,0332	0,9667
26	0,8603	0,1395	0,0002	86	0,0058	0,9927	0,0015	146	0,0005	0,1752	0,8243
27	0,9745	0,0255	0,0000	87	0,0007	0,9931	0,0062	147	0,0000	0,0433	0,9567
28	0,9516	0,0484	0,0000	88	0,0006	0,9944	0,0049	148	0,0000	0,0156	0,9844
29	0,9208	0,0792	0,0001	89	0,0005	0,9859	0,0135	149	0,0000	0,0050	0,9950
30	0,9885	0,0115	0,0000	90	0,0007	0,9968	0,0025	150	0,0000	0,0046	0,9954
31	0,9811	0,0188	0,0002	91	0,0005	0,9702	0,0293	151	0,0000	0,0038	0,9962
32	0,9950	0,0050	0,0000	92	0,0005	0,9701	0,0294	152	0,0000	0,0038	0,9962
33	0,8139	0,1861	0,0000	93	0,0004	0,9253	0,0743	153	0,0001	0,0166	0,9833
34	0,9824	0,0169	0,0007	94	0,0087	0,9906	0,0007	154	0,0000	0,0038	0,9962
35	0,9592	0,0406	0,0002	95	0,0111	0,9877	0,0012	155	0,0004	0,1057	0,8939
36	0,9258	0,0742	0,0000	96	0,7751	0,2249	0,0000	156	0,0000	0,0137	0,9863
37	0,9625	0,0375	0,0000	97	0,0010	0,5963	0,4028	157	0,0000	0,0043	0,9957
38	0,8767	0,1233	0,0000	98	0,0125	0,9863	0,0012	158	0,0000	0,0321	0,9679
39	0,6611	0,3384	0,0005	99	0,1788	0,8194	0,0018	159	0,0000	0,0035	0,9965
40	0,9801	0,0199	0,0000	100	0,0195	0,9797	0,0008	160	0,0000	0,0035	0,9965
41	0,9626	0,0374	0,0000	101	0,0337	0,9645	0,0017	161	0,0000	0,0035	0,9965
42	0,6785	0,3214	0,0000	102	0,0008	0,9788	0,0204	162	0,0002	0,1577	0,8420
43	0,9917	0,0083	0,0000	103	0,0048	0,9943	0,0010	163	0,0000	0,1638	0,8362
44	0,8045	0,1955	0,0000	104	0,0009	0,9981	0,0009	164	0,0008	0,1396	0,8596
45	0,7310	0,2686	0,0005	105	0,0031	0,9966	0,0004	165	0,0001	0,0100	0,9899
46	0,9461	0,0537	0,0002	106	0,0002	0,9924	0,0074	166	0,0000	0,0519	0,9481
47	0,9904	0,0096	0,0000	107	0,0031	0,9960	0,0009	167	0,0000	0,0036	0,9964
48	0,9813	0,0187	0,0000	108	0,0003	0,9128	0,0869	168	0,0000	0,0041	0,9959
49	0,9885	0,0114	0,0001	109	0,0024	0,9974	0,0002	169	0,0000	0,0035	0,9965
50	0,9975	0,0025	0,0000	110	0,0357	0,9634	0,0010	170	0,0000	0,0038	0,9962
51	0,9949	0,0051	0,0000	111	0,1555	0,8442	0,0003	171	0,0008	0,1383	0,8610
52	0,9942	0,0058	0,0000	112	0,0021	0,9970	0,0008	172	0,0002	0,0222	0,9776
53	0,9946	0,0054	0,0000	113	0,0004	0,6633	0,3364	173	0,0000	0,0034	0,9966
54	0,9928	0,0065	0,0007	114	0,0108	0,9846	0,0046	174	0,0000	0,0217	0,9783
55	0,9758	0,0241	0,0001	115	0,0009	0,9970	0,0022	175	0,0000	0,0099	0,9901
56	0,9551	0,0449	0,0000	116	0,0017	0,9954	0,0029	176	0,0000	0,0046	0,9954
57	0,9794	0,0206	0,0000	117	0,0008	0,9987	0,0004	177	0,0001	0,0075	0,9924
58	0,9844	0,0154	0,0002	118	0,0247	0,9243	0,0510	178	0,0000	0,0039	0,9961
59	0,9967	0,0033	0,0000	119	0,0006	0,0818	0,9176				
60	0,0016	0,9955	0,0029	120	0,0136	0,9862	0,0002				

is obtained by an average of 50 executions with different generated networks. The vertical bars indicate the standard deviation. These figures makes it easier to compare the proposed method benchmark performance with those obtained by Lancichinetti and Fortunato (2009a,b) for the *Cfinder* method, proposed by Palla et al (2005), as we generate the test networks using exactly the same parameters they have used. The proposed method performed better than *Cfinder* when the communities are larger (Figures 7c, 7d, 8c, and 8d).

4 Conclusions

This paper presents a new semi-supervised learning graph-based method for uncovering the network overlapping community structure. The method combines cooperation and competition among particles in order to generate a fuzzy output (soft label) for each node in the network. The fuzzy output correspond to the levels of membership of the nodes to each class. An overlapping measure is derived from these fuzzy output, and it can be considered as a confidence level on the output label. This mechanism has been used to determine outliers in data sets too. The fuzzy output and outlier detection realized by our algorithm provide mechanisms to help stopping error propagation during the semi-supervised learning process, thus avoiding label propagation risk at certain level. It is also able to reclassify incorrectly labeled data items.

Computer simulations were performed with both synthetic and real-world data sets, including vector-based data-sets, network-based data sets, and an evaluation with the LFR benchmark. Their results show that the proposed model is a promising method for classification of data sets with overlapping structure and/or a considerable amount of outliers, as well as detecting and quantifying an overlapping measure for each node in the network.

Acknowledgements This work was supported by the São Paulo State Research Foundation (FAPESP), the Brazilian National Research Council (CNPq), and the Foundation for the Development of Unesp (Fundunesp).

References

Belkin M, Matveeva I, Niyogi P (2004) Regularization and semisupervised learning on large graphs. In: Conference on Learning Theory, Springer, pp 624–638

Belkin M, P N, Sindhwani V (2005) On manifold regularization. In: Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTAT 2005), Society for Artificial Intelligence and Statistics, New Jersey, pp 17–24

Blum A, Chawla S (2001) Learning from labeled and unlabeled data using graph mincuts. In: Proceedings of the Eighteenth International Conference on Machine Learning, Morgan Kaufmann, San Francisco, pp 19–26

Breve FA, Zhao L, Quiles MG (2009) Uncovering overlap community structure in complex networks using particle competition. In: International Conference on Artificial Intelligence and Computational Intelligence (AICI'09), vol 5855, pp 619–628

Breve FA, Zhao L, Quiles MG, Pedrycz W, Liu J (2011) Particle competition and cooperation in networks for semi-supervised learning. IEEE Transactions on Knowledge and Data Engineering (PrePrints), DOI 10.1109/TKDE.2011.119, URL `http://doi.ieeecomputersociety.org/10.1109/TKDE.2011.119`

Chapelle O, Schölkopf B, Zien A (eds) (2006) Semi-Supervised Learning. Adaptive Computation and Machine Learning, The MIT Press, Cambridge, MA

Danon L, Díaz-Guilera A, Duch J, Arenas A (2005) Comparing community structure identification. Journal of Statistical Mechanics: Theory and Experiment 9:P09,008

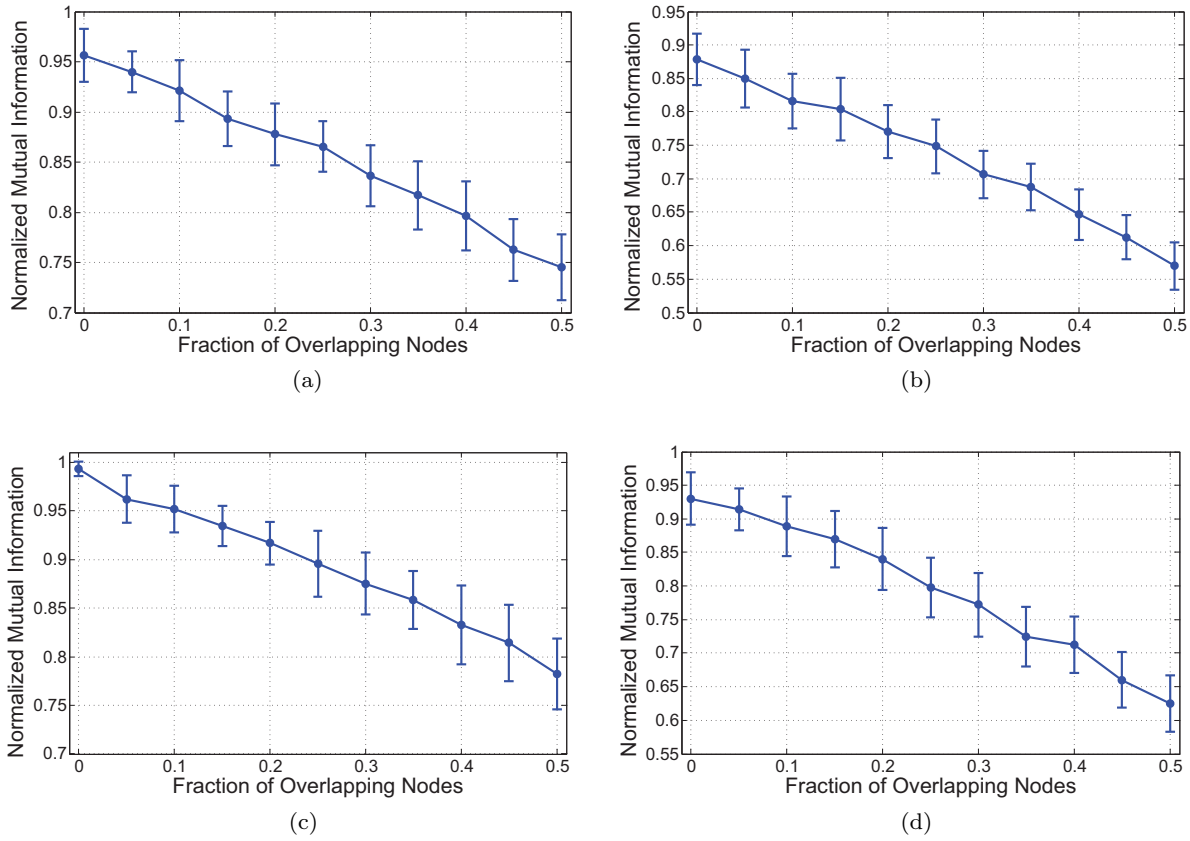


Fig. 7: Test of the proposed method on the benchmark for undirected and unweighted networks with overlapping communities (Lancichinetti and Fortunato 2009a). The plot shows the variation of the normalized mutual information between the planted and the recovered partition, in its generalized form for overlapping communities (Lancichinetti et al 2009), with the fraction of overlapping nodes. The error bars indicate standard deviation. The networks have 1000 nodes, the other parameters are $\tau_1 = 2$, $\tau_2 = 1$, $\langle k \rangle = 20$, and $k_{\max} = 50$. (a) $S_{\min} = 10$, $S_{\min} = 50$, $\mu_t = 0.1$; (b) $S_{\min} = 10$, $S_{\max} = 50$, $\mu_t = 0.3$; (c) $S_{\min} = 20$, $S_{\max} = 100$, $\mu_t = 0.1$; (d) $S_{\min} = 20$, $S_{\max} = 100$, $\mu_t = 0.3$.

Duch J, Arenas A (2005) Community detection in complex networks using extremal optimization. Physical Review E 72:027,104

Duin R, Juszczak P, Paclik P, Pekalska E, de Ridder D, Tax D, Verzakov S (2007) Prtools4.1, a matlab toolbox for pattern recognition

Fortunato S (2010) Community detection in graphs. Physics Reports 486(3-5):75 – 174, DOI DOI:10.1016/j.physrep.2009.11.002

Frank A, Asuncion A (2010) UCI machine learning repository. URL `http://archive.ics.uci.edu/ml`

Joachims T (2003) Transductive learning via spectral graph partitioning. In: Proceedings of International Conference on Machine Learning, AAAI Press, pp 290–297

Karypis G, Han EH, Kumar V (1999) Chameleon: hierarchical clustering using dynamic modeling. IEEE Computer 32(8):68 – 75

Lancichinetti A, Fortunato S (2009a) Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. Phys Rev E 80:016,118, DOI 10.1103/PhysRevE.80.016118, URL `http://link.aps.org/doi/10.1103/PhysRevE.80.016118`

Lancichinetti A, Fortunato S (2009b) Community detection algorithms: A comparative analysis. Phys Rev E 80:056,117, DOI 10.1103/PhysRevE.80.056117, URL `http://link.aps.org/doi/10.1103/PhysRevE.80.056117`

Lancichinetti A, Fortunato S, Kertész J (2009) Detecting the overlapping and hierarchical community structure in complex networks. New Journal of Physics 11(3):033,015, URL `http://stacks.iop.org/1367-2630/11/i=3/a=033015`

Newman M (2006) Modularity and community structure in networks. In: Proceedings of the National Academy of Science of the United States of America, vol 103, pp 8577–8582

Newman MEJ, Girvan M (2004) Finding and evaluating community structure in networks. Physical Review E 69:026,113

Palla G, Derényi I, Farkas I, Vicsek T (2005) Uncovering the overlapping community structure of complex networks in

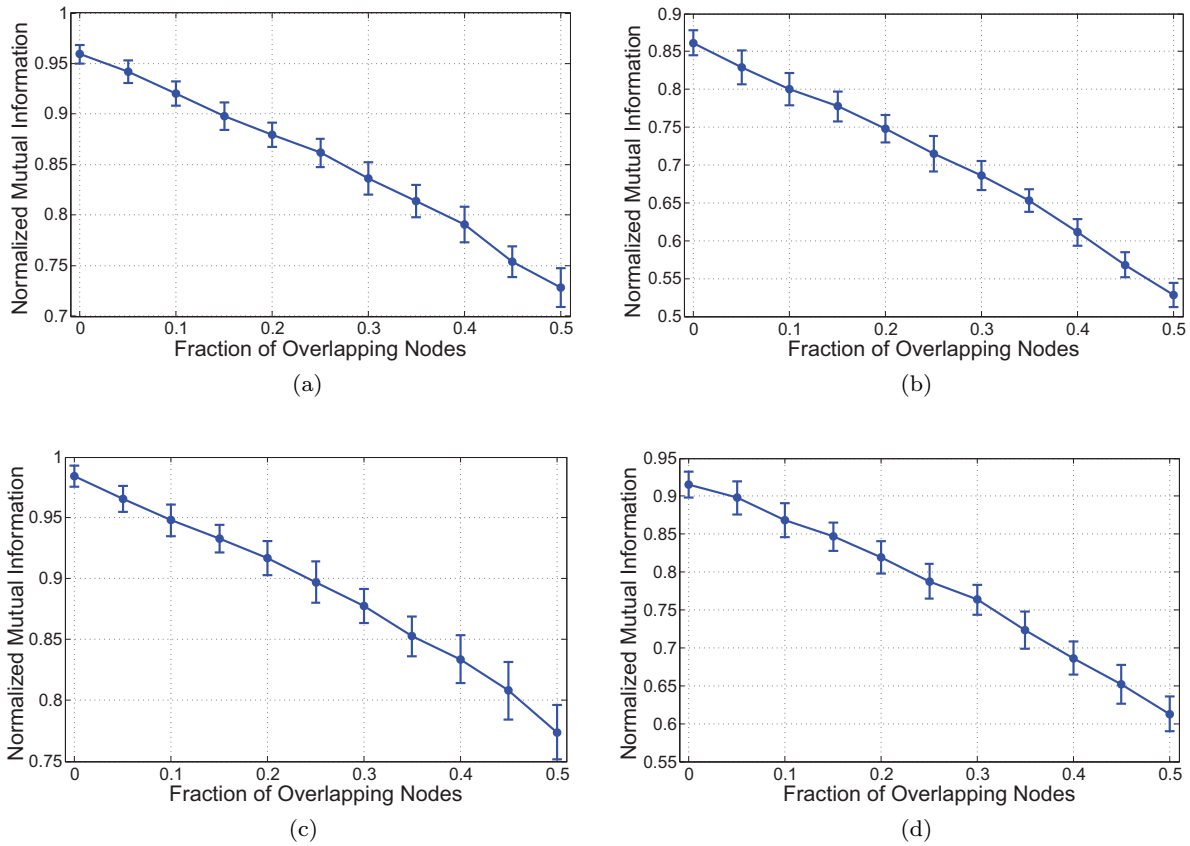


Fig. 8: Test of the proposed method on the benchmark for undirected and unweighted networks with overlapping communities (Lancichinetti and Fortunato 2009a). The plot shows the variation of the normalized mutual information between the planted and the recovered partition, in its generalized form for overlapping communities (Lancichinetti et al 2009), with the fraction of overlapping nodes. The error bars indicate standard deviation. The networks have 5000 nodes, the other parameters are $\tau_1 = 2$, $\tau_2 = 1$, $\langle k \rangle = 20$, and $k_{\max} = 50$. (a) $S_{\min} = 10$, $S_{\min} = 50$, $\mu_t = 0.1$; (b) $S_{\min} = 10$, $S_{\max} = 50$, $\mu_t = 0.3$; (c) $S_{\min} = 20$, $S_{\max} = 100$, $\mu_t = 0.1$; (d) $S_{\min} = 20$, $S_{\max} = 100$, $\mu_t = 0.3$.

nature and society. Nature 435(7043):814–818, DOI http://dx.doi.org/10.1038/nature03607

Quiles MG, Zhao L, Alonso RL, Romero RAF (2008) Particle competition for complex network community detection. Chaos 18(3):033,107, DOI 10.1063/1.2956982

Reichardt J, Bornholdt S (2004) Detecting fuzzy community structures in complex networks with a potts model. Physical Review Letters 93(21):218,701

Zachary WW (1977) An information flow model for conflict and fission in small groups. Journal of Anthropological Research 33:452–473

Zhang S, Wang RS, Zhang XS (2007a) Identification of overlapping community structure in complex networks using fuzzy c-means clustering. Physica A Statistical Mechanics and its Applications 374:483–490, DOI 10.1016/j.physa.2006.07.023

Zhang S, Wang RS, Zhang XS (2007b) Uncovering fuzzy community structure in complex networks. Physical Review E 76(4):046103, DOI 10.1103/PhysRevE.76.046103

Zhou D, Bousquet O, Lal TN, Weston J, Schölkopf B (2004) Learning with local and global consistency. In: Advances in Neural Information Processing Systems, MIT Press, vol 16, pp 321–328

Zhu X (2005) Semi-supervised learning literature survey. Tech. Rep. 1530, Computer Sciences, University of Wisconsin-Madison

Zhu X, Ghahramani Z, Lafferty J (2003) Semi-supervised learning using gaussian fields and harmonic functions. In: Proceedings of the Twentieth International Conference on Machine Learning, pp 912–919