

Particle Competition and Cooperation for Uncovering Network Overlap Community Structure

Fabricio Breve¹, Liang Zhao¹, Marcos Quiles², Witold Pedrycz^{3,4}, and Jiming Liu⁵

¹ Department of Computation, Institute of Mathematics and Computer Science, University of São Paulo, São Carlos, SP, Brazil, 13560-970. E-mail: {fabricio,zhao}@icmc.usp.br

² Department of Science and Technology, Federal University of São Paulo, São José dos Campos, SP, Brazil. E-mail: quiles@unifesp.br

³ Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, T6R 2V4, Canada. E-mail: pedrycz@ee.ualberta.ca

⁴ Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland.

⁵ Computer Science Department, Hong Kong Baptist University, Kowloon, Hong Kong. E-mail: jiming@comp.hkbu.edu.hk

Abstract. Identification and classification of overlap nodes in communities is an important topic in data mining. In this paper, a new graph-based (network-based) semi-supervised learning method is proposed. It is based on competition and cooperation among walking particles in the network to uncover overlap nodes, i.e., the algorithm can output continuous-valued output (soft labels), which corresponds to the levels of membership from the nodes to each of the communities. Computer simulations carried out for synthetic and real-world data sets provide a numeric quantification of the performance of the method.

Keywords: Graph-based method, community detection, particle competition and cooperation, overlap nodes

1 Introduction

Community detection in networks is an important data mining problem that has received increasing interest in the last years. Many networks are found to be divided naturally into communities or modules, therefore discovering of these communities structure became an important research topic [6, 5, 3, 9, 2]. The problem of community detection is very hard and not yet satisfactorily solved, despite a large amount of efforts having been made over the past years [1].

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. However, in practice there are common cases where some nodes in a network can belong to more than one 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 *overlap nodes*, and most known community detection algorithms cannot detect them. Therefore, uncovering the overlapping community structure of complex networks becomes an important topic in data mining [11, 7, 12].

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 in [8], in which only hard labels can be produced by unsupervised learning. That model features walking particles in the network competing with each other in order to possess as many nodes as possible. In the proposed method, besides of the particle competition mechanism, we also introduce the cooperative behavior through the concept of teams of particles. Particles in the same team cooperate with their teammates and compete against 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 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 overlap community structure in networks.

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

Given a data set $\chi = \{x_1, x_2, \dots, x_l, x_{l+1}, \dots, x_n\} \subset \mathbb{R}^m$ and the corresponding label set $L = \{1, 2, \dots, c\}$, the first l points $x_i (i \leq l)$ are labeled as $y_i \in L$ and the remaining points $x_u (l < u \leq n)$ are left unlabeled, i.e, $y_u = \emptyset$. The goal of the algorithm is to provide a vector of membership degrees to each class for each of the unlabeled data, which corresponds to a node in the graph representation.

The first step of the algorithm is to build a network from a given data set. Define a graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$. $\mathbf{V} = \{v_1, v_2, \dots, v_n\}$ is the set of nodes, where each one v_i corresponds to a sample x_i . \mathbf{E} is the set of edges (v_i, v_j) , which can also be represented by an adjacency matrix \mathbf{W} :

$$W_{ij} = \begin{cases} 1 & \text{if } \|x_i - x_j\|^2 \leq \sigma \text{ and } i \neq j \\ 0 & \text{if } \|x_i - x_j\|^2 > \sigma \text{ or } i = j \end{cases}, \quad (1)$$

where W_{ij} specifies whether there is an edge between the pair of nodes x_i and x_j . σ is a threshold which defines the distance between x_i and x_j below which the nodes v_i and v_j are connected, and $\|\cdot\|$ is the Euclidean distance.

For each labeled data $x_i \in \{x_1, x_2, \dots, x_l\}$ or its corresponding node in the network $v_i \in \{v_1, v_2, \dots, v_l\}$, a particle $\rho_i \in \{\rho_1, \rho_2, \dots, \rho_l\}$ is generated and its initial position is at the node v_i . Thus, there is a particle for each labeled sample in the data set. 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 from samples with the same class labels form a *team* and cooperate with each other to compete with other teams.

Each particle ρ_j comes with two variables: $\rho_j^\omega(t)$ and $\rho_j^d(\mathbf{t})$. 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 $\rho_j^d(\mathbf{t}) = \{\rho_j^{d_1}(t), \rho_j^{d_2}(t), \dots, \rho_j^{d_n}(t)\}$, where each element $\rho_j^{d_i}(t) \in [0, n-1]$ 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 $\mathbf{v}_i^\omega(\mathbf{t}) = \{v_i^{\omega_1}(t), v_i^{\omega_2}(t), \dots, v_i^{\omega_c}(t)\}$ 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:

$$\sum_{\ell=1}^c v_i^{\omega_\ell} = 1. \quad (2)$$

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 $\mathbf{v}_i^\lambda(\mathbf{t}) = \{v_i^{\lambda_1}(t), v_i^{\lambda_2}(t), \dots, v_i^{\lambda_c}(t)\}$, and each element $v_i^{\lambda_\ell}(t) \in [0, \infty]$ 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 the initial value of its instantaneous domination vector \mathbf{v}_i^ω set as follows:

$$v_i^{\omega_\ell}(0) = \begin{cases} 1 & \text{if } y_i = \ell \\ 0 & \text{if } y_i \neq \ell \text{ and } y_i \in L \\ \frac{1}{c} & \text{if } y_i = \emptyset \end{cases}, \quad (3)$$

i.e., for each node corresponding to a labeled data item, 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; for each node corresponding to an unlabeled data item, the domination levels of all particle teams are set to the same value $\frac{1}{c}$, where c is the number of classes (number of teams). On the other hand, in all nodes, all long term domination levels $\mathbf{v}_i^\lambda(\mathbf{0})$ 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:

$$\rho_j^\omega(0) = 1, \quad (4)$$

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:

$$\rho_j^{d_i}(t) = \begin{cases} 0 & \text{if } i = j \\ n - 1 & \text{if } i \neq j \end{cases}, \quad (5)$$

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 $\rho_j^r(t)$ will be chosen with probabilities defined by Eq. (6) or Eq. (7), respectively.

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

$$p(v_i|\rho_j) = \frac{W_{qi}}{\sum_{\mu=1}^n W_{q\mu}}, \quad (6)$$

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:

$$p(v_i|\rho_j) = \frac{W_{qi} v_i^{\omega_\ell} \frac{1}{(1+\rho_j^{d_i})^2}}{\sum_{\mu=1}^n W_{q\mu} v_i^{\omega_\ell} \frac{1}{(1+\rho_j^{d_i})^2}}. \quad (7)$$

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:

$$v_i^{\omega_\ell}(t+1) = \begin{cases} \max\{0, v_i^{\omega_\ell}(t) - \frac{\Delta_v \rho_j^\omega(t)}{c-1}\} \\ \quad \text{if } y_i = \emptyset \text{ and } \ell \neq \rho_j^f \\ v_i^{\omega_\ell}(t) + \sum_{q \neq \ell} v_i^{\omega_q}(t) - v_i^{\omega_q}(t+1), \\ \quad \text{if } y_i = \emptyset \text{ and } \ell = \rho_j^f \\ v_i^{\omega_\ell}(t) \quad \text{if } y_i \in L \end{cases}, \quad (8)$$

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. (2). The instantaneous domination level of all labeled node v_i^ω are always fixed, as defined by the third case expressed by Eq. (8).

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::

$$v_i^{\lambda_\ell}(t+1) = v_i^{\lambda_\ell}(t) + \rho_j^\omega(t) \quad (9)$$

where ℓ is the class label of particle ρ_j . Eq. (9) 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.

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:

$$\rho_j^\omega(t+1) = v_i^{\omega_\ell}(t+1), \quad (10)$$

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:

$$\rho_j^{d_k}(t+1) = \begin{cases} \rho_j^{d_i}(t) + 1 & \text{if } \rho_j^{d_i}(t) + 1 < \rho_j^{d_k}(t) \\ \rho_j^{d_k}(t) & \text{otherwise} \end{cases}, \quad (11)$$

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.

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

$$f_i^\ell = \frac{v_i^{\lambda_\ell}(\infty)}{\sum_{q=1}^c v_i^{\lambda_q}(\infty)} \quad (12)$$

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 overlap measure in order to easily illustrate the application of the algorithm. Therefore, the overlap index o_i for a node v_i is defined as follow:

$$o_i = \frac{f_i^{\ell^{**}}}{f_i^{\ell^*}} \quad (13)$$

where $\ell^* = \arg \max_\ell f_i^\ell$ and $\ell^{**} = \arg \max_{\ell, \ell \neq \ell^*} f_i^\ell$, and $o_i \in [0, 1]$, 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.

3 Computer Simulations

In this section, we present some simulation results to evaluate the effectiveness of these modifications. The graphs are built from the data sets by using Eq. (1), with the parameter σ being empirically set for each problem, i.e., a set of simulations is executed by varying σ and the value leading to the best result is chosen. The algorithm parameters in this modified version are less sensitive than in the original version, and therefore they are empirically set to $\Delta_v = 0.1$ and $p_{\text{grd}} = 0.5$ for all the experiments in this subsection.

Figure 1a shows a data set with 4 classes with Gaussian distribution, generated by using PRTools [4] function `gendats` with 1,000 elements (250 per class) and 20 samples are labeled (5 per class), represented by the *red squares*, *blue triangles*, *green lozenges* and *purple stars*. The algorithm is applied to the data set and the detected overlap indexes are shown in Fig. 1b. We see that the

nodes in the interior of each class are small and dark blue, i.e., they are clearly non-overlapping nodes. Meanwhile, the nodes in the borders among classes have tonalities and larger sizes, which represent their different levels of overlap. These results are in agreement with our intuition.

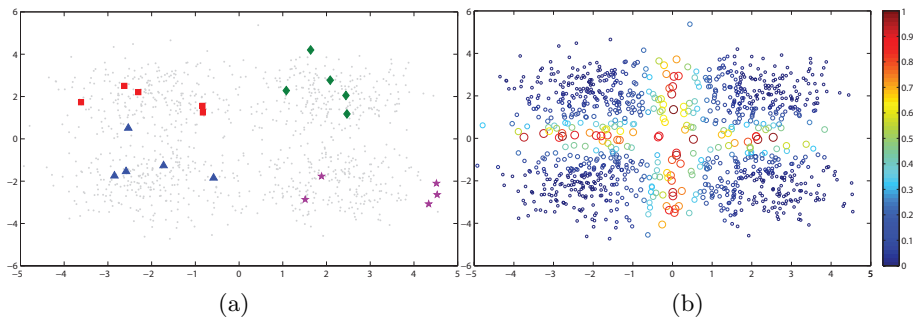


Fig. 1: Classification of normally distributed classes (Gaussian distribution). (a) toy data set with 1,000 samples divided in four classes, 20 samples are labeled, 5 from each class (red squares, blue triangles, green lozenges and purple stars). (b) nodes size and colors represent their respective overlap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

The proposed algorithm is also applied to a real-world data set: the Zachary's Karate Club Network [10]. 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. 2, and the overlap index of each node is indicated by their sizes and colors. Our visual inspection indicates that this is a good result as well. Notice that although the two labeled nodes exhibit some degree of overlap, the algorithm still produced a good result, even detecting these overlap degrees in the labeled nodes (notice the slightly larger size and the lighter blue color). This is also a desirable feature, since we do not need to choose a non-overlap node to represent a class.

4 Conclusions

This paper presents a new semi-supervised learning graph-based method for uncovering the network overlap 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 overlap measure is derived from these fuzzy output, and it can be considered as a confidence level on the output label.

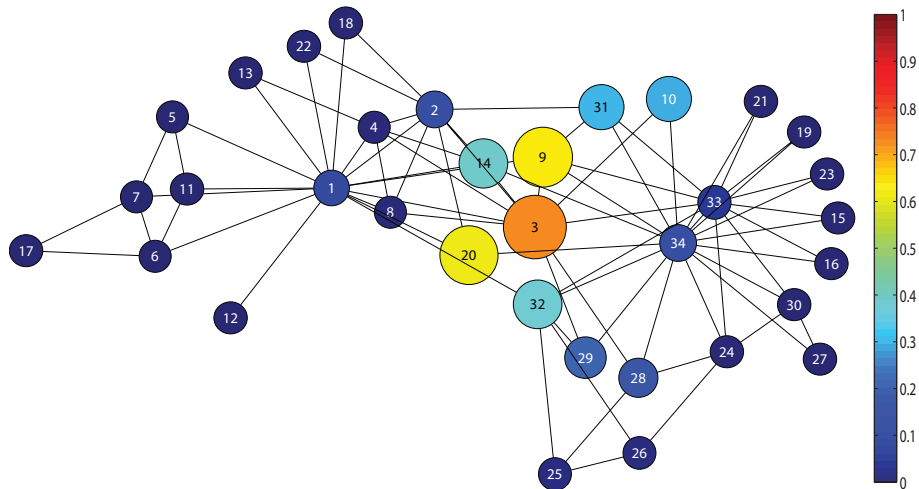


Fig. 2: The karate club network. Nodes size and colors represent their respective overlap index detected by the proposed method.

Computer simulations with both synthetic and real-world data sets show that the proposed model is a promising method for classification of data sets with overlap structure, as well as detecting and quantifying an overlap measure for each node in the network.

Acknowledgments. This work is supported by the State of São Paulo Research Foundation (FAPESP) and the Brazilian National Council of Technological and Scientific Development (CNPq).

References

1. Community detection in graphs. *Physics Reports* 486(3-5), 75 – 174 (2010)
2. Danon, L., Díaz-Guilera, A., Duch, J., Arenas, A.: Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment* 9, P09008 (2005)
3. Duch, J., Arenas, A.: Community detection in complex networks using extremal optimization. *Physical Review E* 72, 027104 (2005)
4. Duin, R., Juszczak, P., Paclik, P., Pekalska, E., de Ridder, D., Tax, D., Verzakov, S.: Prtools4.1, a matlab toolbox for pattern recognition
5. Newman, M.: 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 (2006)
6. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. *Physical Review E* 69, 026113 (2004)
7. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. *Nature* (7043), 814–818 (2005)

8. Quiles, M.G., Zhao, L., Alonso, R.L., Romero, R.A.F.: Particle competition for complex network community detection. *Chaos* 18(3), 033107 (2008)
9. Reichardt, J., Bornholdt, S.: Detecting fuzzy community structures in complex networks with a potts model. *Physical Review Letters* 93(21), 218701 (2004)
10. Zachary, W.W.: An information flow model for conflict and fission in small groups. *Journal of Anthropological Research* 33, 452–473 (1977)
11. Zhang, S., Wang, R.S., Zhang, X.S.: Identification of overlapping community structure in complex networks using fuzzy c-means clustering. *Physica A Statistical Mechanics and its Applications* (2007)
12. Zhang, S., Wang, R.S., Zhang, X.S.: Uncovering fuzzy community structure in complex networks. *Physical Review E* 76(4), 046103 (2007)