X Congresso Brasileiro de Inteligência Computacional (CBIC'2011), 8 a 11 de Novembro de 2011, Fortaleza, Ceará
© Sociedade Brasileira de Inteligência Computacional (SBIC)

Uncovering Overlapping Structures via Stochastic Competitive Learning

Thiago C. Silva and Liang Zhao
Department of Computer Sciences, Institute of Mathematics and Computer Science (ICMC), University of São Paulo
{thiagoch, zhao}@icmc.usp.br

Abstract – Competitive learning is an important approach in Machine Learning. In this paper, we present a method for determining overlapping structures or vertices in the network using a stochastic competitive model, where several particles walk in the network and compete with each other to occupy as many nodes as possible, while attempting to reject intruder particles. The proposed measure for detecting overlapping structures is built from the rich information that is inherently embedded within the model description. Therefore, no extra processing is necessary to detect the overlapping structures in the data. Computer simulations reveal that the proposed overlapping index works well in real-world data sets.

Keywords – Stochastic competitive learning, overlapping vertices, unsupervised learning.

1. Introduction

Machine learning is an active research topic in artificial intelligence [1]. It concerns the development of algorithms that allow computers to adapt behaviors based on empirical input data. There are three main categories of learning schemes: supervised learning, unsupervised learning and semi-supervised learning. Supervised learning aims at mapping each input data item to its corresponding desired output (data labels or supervisors); unsupervised learning has the goal to map the input data, without a supervisor, from one space to another space revealing some intrinsic features, such as data clusters, among others. Between these two extremes reside semi-supervised learning, which combines both labeled and unlabeled examples to generate an appropriate function or classifier.

Competition is a natural process observed in nature and in many social systems sharing limited resources, such as water, food, mates, territory, recognition, etc. Competitive learning is an important approach of machine learning and is widely implemented in artificial neural networks to realize unsupervised learning. Early works include the development of the famous Self-Organizing Map (SOM) [2], Differential Competitive Learning [3], and Adaptive Resonance Theory (ART) [4]. From then on, many competitive learning neural networks have been developed [5] and a wide range of applications, such as data clustering, data visualization, pattern recognition, and image processing have been considered [6]. Without a doubt, competitive learning neural networks (CLNN) represent one of the main successes of neural network development.

In this paper, we study a new type of competitive learning mechanism. Consider a large scale graph (network), where several particles walk in the network and compete with each other to mark their own territory (occupy as many nodes as possible), while attempting to reject intruder particles. Each particle can perform a random walk by choosing any neighbor to visit, a preferential walk by choosing the node with the highest domination to visit or a combination of them. Straightforward applications are in community detection and data clustering. In essence, data clustering can be considered as a community detection problem once a network is constructed from the original data set, where each node corresponds to a data item and the connections are established by using a certain similarity measure. Considering such applications, the competitive walking process reaches dynamics equilibrium when each community or a data cluster is dominated by a single particle.

Interestingly, the particle competition process is rather similar to many natural and social processes, such as resource competition by animals, territory exploration by humans (animal), election campaigns, etc. Moreover, the combined random-preferential walking of competitive particles can largely improve the community detection rate. The model highlights the role of randomness in evolutionary systems, where it serves to automatically escape from some undesirable traps and to explore new spaces. Therefore, a certain level of randomness is essential for the learning process. Such randomness represents the "I don't know" state and serves as a novelty finder. It can also help learning agents, like particles in our model, to escape from traps in physical or learning spaces.

The particle competition model was originally proposed in [7], where only a procedure of particle competition is introduced without formal definition. In the present work, a rigorous definition is provided, where the particle competition is formally represented by a stochastic dynamical system. Since the models of several interactive walking processes correspond to many natural and artificial systems, the study of this topic is important. Due to the lack of theory for such models, our approach (the model definition) is an important step to understand and dominate such systems. Moreover, we have developed an efficient method for determining the overlapping nodes in the input data by using the dominance level information generated by the competition process itself, i.e., the detection procedure is already embedded in the model, which is a characteristic that the majority of the competing techniques do not have, since, usually, there is a dedicated or separated process to calculate the overlapping characteristics of the input data. As a result, our method does not increase the model's complexity order. Since the determination of overlapping nodes is an important issue in community detection [8], our method presents a contribution to this topic. Another interesting feature is that the underlying network is constructed directly from the input data set, therefore, the correspondence between the input data and the processing result (the final network) is maintained. In this way, we hope that our work is useful for developing the competitive learning theory based on particle walking in large scale networks.

Overlapping community structure has been widely studied [8, 9]. In [10], the authors combine the idea of the modularity function Q, spectral relaxation and fuzzy c-means clustering method in order to build a new modularity function based on generalizing Newman and Girvan's Q function, which is an approximation mapping of network nodes into Euclidean space and fuzzy c-means clustering. In [8], the community structure is uncovered by k-clique percolation and the overlaps between communities are guaranteed by the fact that one node can participate in more than one clique. However, the k-clique method gives rise to an incomplete cover of the network, i.e., some nodes may not belong to any community. In addition, the hierarchical structure cannot be revealed for a given k. In [9], it is presented an algorithm that finds both overlapping communities and the hierarchical structure concomitantly based on a fitness function and a resolution parameter given by the user. Recently, Evans et al [25] proposed a method to recognize the overlapping community structure by partitioning a line graph built from the original network. This method only allows the communities to overlap at nodes. A drawback of the majority of these techniques resides in the fact that the detection of the overlapping characteristics of the input data is performed as a separated or dedicated process.

The remainder of the paper is organized as follows. The proposed model definition is described in Section 2. In Section 3, computer simulations are performed to show how the proposed model solves network community detection and data clustering problems by using artificial and real data sets. Finally, Section 4 concludes the paper.

2. Model Description

In this section, the proposed competitive learning model pertaining to the unsupervised scheme is presented in details.

2.1 The Competitive Transition Matrix

Regarding the movement policy of each particle $k \in \mathcal{K}$, it is basically composed of two distinct types: (i) a random movement term, modeled by the matrix $\mathbb{P}^{(k)}_{\text{rand}}$, which permits the particle to adventure through the network, without accounting for the defense of the previously dominated vertices; (ii) a preferential movement term, modeled by the matrix $\mathbb{P}^{(k)}_{\text{pref}}$, which is responsible for inducing the particle to reinforce the vertices that are owned by itself, i.e., the particle will prefer visiting its dominated vertices, instead of a randomly selected one. In order to model such dynamics, consider the stochastic vector $p(t) = [p^{(1)}(t), p^{(2)}(t), \dots, p^{(K)}(t)]$, which denotes the localization of the set of K particles presented to the network, where the kth-entry, $p^{(k)}(t)$, indicates the location of the particle k in the network at time t, i.e., $p^{(k)}(t) \in \mathcal{V}, \forall k \in \mathcal{K}$. It is desirable to find a transition matrix that governs the probability distribution of the particles' movement to the immediate future state, $p(t+1) = [p^{(1)}(t+1), p^{(2)}(t+1), \dots, p^{(K)}(t+1)]$.

We also introduce energy levels for all particles in the following manner: if a particle visits a vertex that is being dominated by itself, then the corresponding energy of that particle increases. Likewise, if a particle visits a vertex that is being dominated by a rival particle, then the corresponding energy of that particle is drained. If the actual energy of a specific particle reaches a certain minimum threshold, then it is said that the particle has gotten exhausted at that step. In the subsequent step, that particle is automatically reanimated in a vertex that belongs to it in a random manner. At each step, the vertices belonging to each particle can be easily determined by checking the domination level variables of each vertex. With this behavior, we expect that the particles will no longer wander free in the network, possibly swapping territories with other particles several times. Thus, this characteristic is expected to restrain the particles' effective acting region.

With the intent of modeling such dynamics, we introduce the following stochastic vector $S^{(t)} = [S^{(1)}(t), \dots, S^{(K)}(t)]$, where the kth-entry, $S^{(k)}(t) \in \{0, 1\}$, indicates whether the particle k is active or exhausted at time t. Specifically, if $S^{(k)}(t) = 1$, then particle k is said to be exhausted. Likewise, when $S^{(k)}(t) = 0$, the particle is said to be active. Thus, if $S^{(k)}(t) = 0$, the particle navigates in the network according to a combined behavior of randomness and preferential movement towards the dominated vertices. However, if $S^{(k)}(t) = 1$, the particle switches its movement policy to a new transition matrix, here entitled $\mathbb{P}^{(k)}_{\text{rean}}(t)$, which is responsible for taking the particle back to its owned territory ("safe ground"), in order to reanimate the corresponding particle by recharging its energy. After the energy has been properly recharged, the particle can again perform the combined random-preferential movement in the network. In brief, $S(t)$ acts as a switch that determines the movement policy of all particles at time t. With all this information in mind, we are able to define the transition matrix associated to the particle k as:

$$\mathbb{P}^{(k)}_{\text{transition}}(t) \triangleq (1 - S^{(k)}(t)) \left[\lambda \mathbb{P}^{(k)}_{\text{pref}}(t) + (1-\lambda)\mathbb{P}^{(k)}_{\text{rand}}\right] + S^{(k)}(t)\mathbb{P}^{(k)}_{\text{rean}}(t) \tag{1}$$

where $\lambda \in [0, 1]$ indicates the desired fraction of preferential movement that all particles in the network will perform, $\mathbb{P}^{(k)}_{\text{pref}}(t)$ portrays the transition matrix with a probability distribution according to the preferential behavior described above and, likewise, $\mathbb{P}^{(k)}_{\text{rand}}$ describes the random behavior, $S^{(k)}(t)$ indicates whether particle k is active or exhausted, and $\mathbb{P}^{(k)}_{\text{rean}}(t)$ is responsible for the particle reanimation behavior. Specifically, $\mathbb{P}^{(k)}_{\text{transition}}(i, j, t)$ indicates the probability that particle k makes a transition from vertex i to j at time t. It is worth noting that (1) is a convex combination of two transition matrices (the first term is itself a combination of two transition matrices, too), since the sum of the coefficients is unitary, therefore, the resulting matrix is guaranteed to be another transition matrix. Now we proceed to define each matrix that appears in (1) in a detailed manner.

The derivation of the random movement matrix is straightforward, since this matrix is only dependent on the adjacency matrix of the graph, which is previously known. Then, each entry $(i, j) \in \mathcal{V} \times \mathcal{V}$ of the matrix $\mathbb{P}^{(k)}_{\text{rand}}$ is given by:

$$\mathbb{P}^{(k)}_{\text{rand}}(i, j) \triangleq \frac{a_{i,j}}{\sum_{u=1}^{V} a_{i,u}} \tag{2}$$

where $a_{i,j}$ denotes the (i, j)th-entry of the adjacency matrix A of the graph. Note that (2) resembles the traditional Markovian matrix for a single random walker, here symbolized as a particle [11]. Also note that matrix $\mathbb{P}^{(k)}_{\text{rand}}$ is time-invariant and is the same for every particle in the network; therefore, we will drop the superscript k whenever the situation makes it clear. In short terms, the probability of an adjacent neighbor to be visited using only the random movement behavior is proportional to the edge weight linking the vertex that a specific particle is visiting and that neighbor vertex.

In order to assist in the calculation of the matrix associated to the preferential movement term, $\mathbb{P}^{(k)}_{\text{pref}}(t)$, for a given particle $k \in \mathcal{K}$, we introduce the following stochastic vector:

$$N_i(t) \triangleq [N_i^{(1)}(t), N_i^{(2)}(t), \dots, N_i^{(K)}(t)] \tag{3}$$

where $\dim(N_i(t)) = 1 \times K$ and $N_i(t)$ stands for the number of visits received by vertex i up to time t by all the particles scattered throughout the network. Specifically, the kth-entry, $N_i^{(k)}(t)$, indicates the number of visits made by the particle k to vertex i up to time t. We now extend this notation to all vertices in the network, defining the global matrix that maintains the number of visits made by every particle in the network to all the vertices as:

$$N(t) \triangleq [N_1(t), N_2(t), \dots, N_V(t)]^T \tag{4}$$

where $\dim(N(t)) = V \times K$. Let us also formally define the domination level vector of vertex i, $\bar{N}_i(t)$, according to the following stochastic vector:

$$\bar{N}_i(t) \triangleq [\bar{N}_i^{(1)}(t), \bar{N}_i^{(2)}(t), \dots, \bar{N}_i^{(K)}(t)] \tag{5}$$

where $\dim(\bar{N}_i(t)) = 1 \times K$ and $\bar{N}_i(t)$ denotes the relative frequency of visits of all particles in the network to vertex i until the time t (included). Particularly, the kth-entry, $\bar{N}_i^{(k)}(t)$, indicates the relative frequency of visits performed by particle k to vertex i up to time t. Similarly to the previous case, we extend this notion to all vertices in the network, defining the domination level matrix that sustains all the domination levels imposed by every particle in the network to all the vertices as:

$$\bar{N}(t) \triangleq [\bar{N}_1(t), \bar{N}_2(t), \dots, \bar{N}_V(t)]^T \tag{6}$$

where $\dim(N(t)) = V \times K$. Mathematically, we define each entry of $\bar{N}_i^{(k)}(t)$ as:

$$\bar{N}_i^{(k)}(t) \triangleq \frac{N_i^{(k)}(t)}{\sum_{u=1}^{K} N_i^{(u)}(t)} \tag{7}$$

In view of that, we can define $\mathbb{P}^{(k)}_{\text{pref}}(i, j, t)$, which is the probability of a single particle k to perform a transition from vertex i to j at time t, using solely the preferential movement term, as follows:

$$\mathbb{P}^{(k)}_{\text{pref}}(i, j, t) \triangleq \frac{a_{i,j}\bar{N}_j^{(k)}(t)}{\sum_{u=1}^{V} a_{i,u}\bar{N}_u^{(k)}(t)} \tag{8}$$

Clearly, from (8), it can be observed that each particle has a different transition matrix associated to its preferential movement and that, unlike the matrix associated to the random movement, this matrix is time-variant with dependence on the domination levels of all the vertices ($\bar{N}(t)$) in the network at the time t. It is worth remarking that the approach taken here to characterize the preferential movement of the particles is the visiting frequency of each particle to a specific vertex, in such a way that as more visits are performed from a specific particle to an arbitrary vertex, the higher will be the chance of the same particle repeatedly visit the same vertex. Also it is important to emphasize that (8) produces two distinct features presented by a natural competition model: (i) the strengthening of the domination level of the visiting particle to a vertex; (ii) the consequent weakening of the domination levels of all other particles on that same vertex. This behavior is naturally represented by the model due to the frequency approach taken.

Now we define each entry of $\mathbb{P}^{(k)}_{\text{rean}}(t)$ that is accounted for teleporting an exhausted particle $k \in \mathcal{K}$ back to its owned territory, with the purpose of recharging its energy (reanimation process). Suppose that particle k is visiting vertex i when its energy is completely depleted. In this situation, the particle teleports back to an arbitrary vertex j of its possession at time t according to the probability given by:

$$\mathbb{P}^{(k)}_{\text{rean}}(i,j,t) \triangleq \frac{\mathbb{1}_{\left\{\arg\max\limits_{m\in\mathcal{K}}\left(\bar{N}_j^{(m)}(t)\right)=k\right\}}}{\sum_{u=0}^{V} \mathbb{1}_{\left\{\arg\max\limits_{m\in\mathcal{K}}\left(\bar{N}_u^{(m)}(t)\right)=k\right\}}} \tag{9}$$

where arg $\max\limits_{m\in\mathbb{K}}(.)$ returns the index m which maximizes the argument and $\mathbb{1}_{\{.\}}$ is the indicator functions that yields 1 if the argument is logically true and 0, otherwise. Indeed, a careful analysis of the expression in (9) reveals that the probability of returning to an arbitrary vertex j dominated by the particle k follows a uniform distribution. Moreover, all rows of this matrix are equal, showing that this movement does not depend on which vertex a specific particle is. This provides a compact way of computationally representing such structure. With that in mind, (9) only results in non-zero transition probabilities for vertices j that are being dominated by particle k at time t, regardless of the existence of a connection between i and j in the adjacency matrix. In essence, once the particle is exhausted, the switch is enabled, which, in turn, compels the particle k to return to its previously owned territory to be recharged, no matter whether there is a physical connection or not in the adjacency matrix. If no vertex is being dominated by particle k at time t, we deliberately put it in any vertex of the network in a random manner, using a uniform distribution.

Now we proceed to the development of the particle's energy update rule. Firstly, it is useful to introduce the stochastic vector $E(t) = [E^{(1)}(t), \ldots, E^{(K)}(t)]$, where the kth-entry, $E^{(k)}(t) \in [\omega_{\min}, \omega_{\max}]$, $\omega_{\max} \geq \omega_{\min}$, denotes the energy level of particle k at time t, whose update rule is given by:

$$E^{(k)}(t) = \begin{cases} \min(\omega_{\max}, E^{(k)}(t-1) + \Delta), & \text{if } \mathrm{owner}(k,t) \\ \max(\omega_{\min}, E^{(k)}(t-1) - \Delta), & \text{if } \neg\ \mathrm{owner}(k,t) \end{cases} \tag{10}$$

where $\mathrm{owner(k,t)} = \left(\arg\max\limits_{m\in\mathcal{K}}\left(\bar{N}^{(m)}_{p^{(k)}(t)}(t)\right) = k\right)$ is a logical expression that essentially yields true if the vertex that particle k visits at time t (i.e., vertex $p^{(k)}(t)$) is being dominated by the visiting particle, and false otherwise; $\dim(E(t)) = 1 \times K$; $\Delta > 0$ symbolizes the increment or decrement of energy that each particle will receive at time t. Indeed, the first expression in (10) represents the increment of the particle's energy and occurs when the particle k visits a vertex $p^{(k)}(t)$ which is dominated by itself, i.e., arg $\max\limits_{m\in\mathbb{K}}\left(\bar{N}^{(m)}_{p^{(k)}(t)}(t)\right) = k$. Similarly, the second expression in (10) portrays the decrement of the particle's energy and occurs when particle k visits a vertex $p^{(k)}(t)$ which is not dominated by itself, i.e., there is a domination level on that vertex that is higher than the one imposed by particle k. Hence, in this model, particles will be given a penalty if they are wandering in rival territory, so as to minimize aimless navigation of the particles in the network which would only reduce the speed of convergence of the dynamical system. By the same reasons, we expect this behavior to improve the cluster and community detection rates of the algorithm.

Now we advance to the update rule that governs $S(t)$, which is responsible for determining the movement policy of each particle. As we have stated, an arbitrary particle k will be transported back to its domain only if its energy drops to a threshold $\omega_{\min}$. With that in mind, it is natural that each entry of $S^{(k)}(t)$ has to monitor the current energy value of its corresponding particle k, i.e., if it ever drops to the given threshold, the switch must be enabled; analogously, if the particle still has an energy value greater than the threshold, then the switch should be disabled. Mathematically, the kth-entry of $S(t)$ can be precisely written as:

$$S^{(k)}(t) = \mathbb{1}_{\{E^{(k)}(t)=\omega_{\min}\}} \tag{11}$$

where $\dim(S(t)) = 1 \times K$. Specifically, $S^{(k)}(t) = 1$ if $E^{(k)}(t) = \omega_{\min}$ and 0, otherwise. As there is an upper limit for the random variable $E^{(k)}(t)$, it is clear that if particle k frequently visits vertices owned by rival particles, its energy will decrease in such a way that it could reach the minimum energy $\omega_{\min}$ and, hence, become exhausted. The upper limit, $\omega_{\max}$, is established to prevent any particle in the network to keep increasing its energy to an undesirably high value (by constantly visiting vertices in its territory), and, once this energy is high enough, it can go far away from its territory and visit a substantial number of vertices belonging to rival particles before becoming exhausted, thus, considerably decreasing the convergence time and the community and cluster detection rates of the dynamical system.

2.2 The Unsupervised Competitive Learning Model

In light of all we have obtained in the previous section, we are ready to enunciate the proposed dynamical system which models the competition of particles in a given network. The internal state of the dynamical system is denoted as $X(t) = [N(t)\ \ p(t)\ \ E(t)\ \ S(t)]$ and the proposed competitive dynamical system as:

$$\phi : \begin{cases} N_i^{(k)}(t+1) & = N_i^{(k)}(t) + \mathbb{1}_{\{p^{(k)}(t+1)=i\}} \\ E^{(k)}(t+1) & = \begin{cases} \min(\omega_{\max}, E^{(k)}(t) + \Delta), \text{if owner}(k,t) \\ \max(\omega_{\min}, E^{(k)}(t) - \Delta), \text{if } \neg \text{ owner}(k,t) \end{cases} \\ S^{(k)}(t+1) & = \mathbb{1}_{\{E^{(k)}(t+1)=\omega_{\min}\}} \end{cases} \tag{12}$$

where, by the considerations that we have previously stated, $\dim(N(t)) = V \times K$, $\dim(p(t)) = 1 \times K$, $\dim(E(t)) = 1 \times K$, and $\dim(S(t)) = 1 \times K$, resulting that $\dim(X(t)) = (V+3) \times K$, with $N_i^{(k)}(t) \in [1, \infty)$, $(i,k) \in \mathcal{S}$, where $\mathcal{S}$ is the space spawned by $\mathcal{V} \times \mathcal{K}$. Observe that $p(t+1)$ has no closed form because it is qualified as a distribution with dependence on $p(t)$ and $N(t)$, therefore its acquisition is merely by random number generation. Succinctly, the internal state of system ϕ carries the current total number of visits made by each particle to each vertex in the network, the current localization of all particles in the network, the current energy that each particle holds, and the information about each particle whether it is current active or exhausted.

Note that system ϕ is nonlinear. This occurs on account of the indicator function, which is nonlinear. The first equation of system ϕ is responsible for updating the number of visits at vertex i by particle k up to time t; the second equation is used to maintain the current energy levels of all the particles inserted in the network; and the third equation is used to indicate whether the particle is active or exhausted, depending on its actual energy level. It is valuable to emphasize that the first expression of system ϕ must be applied for every $(i,k) \in \mathcal{S}$ and the second and third expressions must be performed for every $k \in \mathcal{K}$ with the intention of one properly derive the full state $X(t)$ of the system ϕ. One can also see that system ϕ is clearly Markovian, since it only depends on the present state to derive the future state.

2.3 The Initial Conditions of the System

In order to run system ϕ, we need a set of initial conditions. Firstly, the particles are randomly inserted in the network, i.e., the values of $p(0)$ are randomly set. The initial positions of the particles do not affect the community detection or data clustering results, because each of them will be confined into a different community due to the competition nature, even if they are put together at the beginning. Regarding the initial condition of the system ϕ, $X(0)$, it is valuable to stress that, with the purpose of (7) to be well-defined, all terms in $\sum_{k=1}^{K} N_i^{(k)}(t)$ cannot be zero simultaneously, $\forall t \geq 0$. In this way, we arbitrary set the initial value of $N_i^{(k)}(0)$ to 1, $\forall (i,k) \in \mathcal{S}$, with no loss of fairness in the competition process. However, we also have to distinguish two types of vertices: (i) vertices from which the particles have generated at time $t = 0$; (ii) all other vertices. In view of that, we suggest the following initial condition to the matrix $N(0)$:

$$N_i^{(k)}(0) = \begin{cases} 2, & \text{if particle } k \text{ is generated at vertex } i \\ 1, & \text{otherwise} \end{cases} \tag{13}$$

Regarding the initial condition of $E(0)$, we desire a fair competition amongst the particles, so we place isonomy in their initial energy values, i.e., all particles $k \in \mathcal{K}$ start out with the same energy level given by:

$$E^{(k)}(0) = \omega_{\min} + \left(\frac{\omega_{\max} - \omega_{\min}}{K} \right) \tag{14}$$

Lastly, the variable that accounts for indicating whether the particle k is active or exhausted at the initial step, $S^{(k)}(0)$, $\forall k \in \mathcal{K}$, is given by:

$$S^{(k)}(0) = 0 \tag{15}$$

i.e., we deliberately set as active all particles in the network at the beginning of the process.

2.4 The Method for Detecting Overlapping Structures

The model that we have proposed in this paper carries a rich set of information throughout time. With the aid of such information, we are going to derive a measure that detects overlapping structures or vertices in a given network. For this matter, it is worth noticing that the domination level matrix $\bar{N}(t)$ can be used to indicate whose vertices are members of just one or several groups or communities in the following way: if the maximum domination level imposed by an arbitrary particle k on a specific vertex i is much greater than the second maximum domination level imposed by another particle on the same vertex, then we can conclude that this vertex is being strongly dominated by particle j and no other particle is influencing it in a relevant manner. Therefore, the overlapping nature of such vertex is minimal. On the other hand, when these two quantities are similar, then we can infer that the vertex in question holds an inherently overlapping characteristic. In light of these considerations, we can mathematically model this behavior as follows: let $M_i(x,t)$ denote the xth greatest domination level value imposed on vertex i at time t, in this way, the overlapping index of vertex i, $O_i(t) \in [0,1]$, is given by:

$$O_i(t) = 1 - (M_i(1,t) - M_i(2,t)) \tag{16}$$

i.e., the overlapping index $O_i(t)$ measures the gap between the two greatest domination levels imposed by any pair of particles in the network on vertex i. Succinctly, when this gap is high, a strong domination is taking place on that vertex and, hence, $O_i(t)$ yields a low value. On the other hand, when the competition is fiercely occurring on vertex i, all the domination levels of the particles on that vertex are expected to reside near each other. Therefore, the gap between the two greatest domination levels is hoped for being low, producing a large value for the overlapping index $O_i(t)$.

3. Computer Simulations

In this section, we present simulation results in order to show the effectiveness of the proposed competitive model and the proposed overlapping index. For all simulations hereon, we will fix $\Delta = 0.05$, $\omega_{\min} = 0$, and $\omega_{\max} = 1$, which are not sensitive to the community detection process.

3.1 Simulations for Community Detection on the Newman's Benchmark

A widely used standard benchmark test [12] for evaluating the robustness of a proposed algorithm in community detection tasks is to make use of artificial random clustered networks. In this case, the robustness of the proposed technique can be measured by analyzing how the community detection rate Ψ behaves as ${}^{z_{out}}/_{\langle k \rangle}$ is increased, since this quantity measures the community interconnectivity (a low value indicates well-defined communities, while a high value means highly mixed communities). Indeed, Fig. 1 shows this analysis. Observe that the proposed technique reaches good detection rates, even for high values of ${}^{z_{out}}/_{\langle k \rangle}$. Furthermore, from a comparison between the results shown in Fig. 2 of [12], which depicts the outcome of 9 different representative community detection techniques with the same experimental setup, and our result shown in Fig. 1, one can state that our technique has certainly obtained one of the highest community detection rates among all the techniques under comparison.

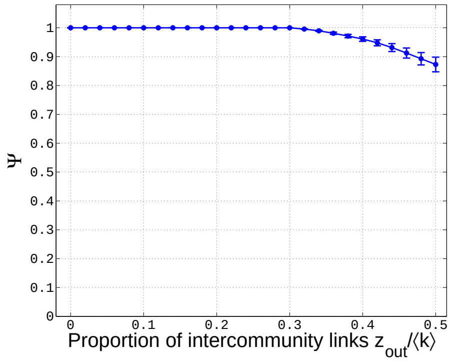

Figure 1: Community detection rate using the proposed technique applied to a random clustered network with $V = 128$, $M = 4$ equally sized communities and $\langle k \rangle = 16$ (Newman's community detection benchmark). The abscissa axis represents the proportion of intercommunity links presented in the network, whereas the ordinate axis, the fraction of nodes correctly grouped. For each point in the trajectory, 200 independent runs were performed. The error bars indicate the standard deviations.

3.2 Simulations for Detecting Overlapping Vertices and Communities

Firstly, we apply our technique to a real world data set entitled Zachary's "karate club" network [13]. This is a well-known network from the social science literature, which has become a benchmark test for community detection algorithms. This network exhibits the pattern of friendship among the 34 members of a club at an American University in the 1970s. The members are represented by vertices and an edge exists if both members know each other. Shortly after the observation of the network, the club dismembered in two as a consequence of an internal dispute, making it an interesting problem for detecting communities. Figure 2 shows the outcome of the simulation. The red (dark gray) and blue (gray) colors denote the communities detected by the algorithm. Only the vertex number 3 (the yellow or light gray vertex) is incorrectly grouped as belonging to the blue (gray) community, when, in reality, it is a member of the red (dark gray) community, as suggested by the real problem. In the literature, the vertices 3 (e.g., see [14]) and 10 (e.g., see [15]) are recurrently misclassified by many community detection algorithms. This happens because the number of edges that they share between the two communities are the same, i.e., they are inherently overlapping, making their clustering a hard problem. In our technique, the overlapping vertex 10 is correctly classified. Thus, good community detection results have been obtained. Now, we apply our overlapping index, as indicated in (16), on every vertex of the Zachary's "karate club" network. This result is shown in Fig. 3. One can see that the highest overlapping indices are yielded by vertices 3 and 10, which confirm our previous analysis. Moreover, vertices 9, 14, 20, 29, and 32 also presents high level of overlapping characteristics, which we can clearly verify from Fig. 2, since these are placed in the borders of each community.

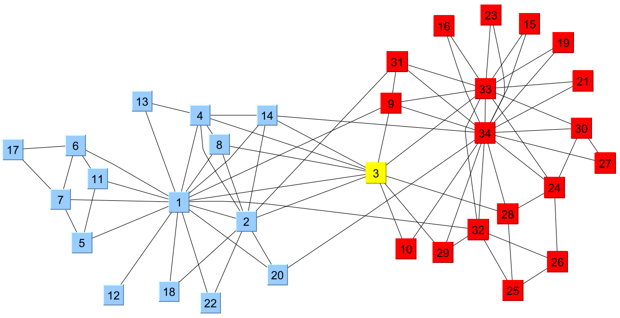

Figure 2: Community detection result of the Zachary's karate club network by using the proposed method. $K = 2$ and $\lambda = 0.6$. We have iterated the dynamical system for 100 steps. The red (dark gray) and blue (gray) colors denote the detected communities. Only the yellow or light gray vertex (vertex 3 in the original database) is incorrectly grouped.

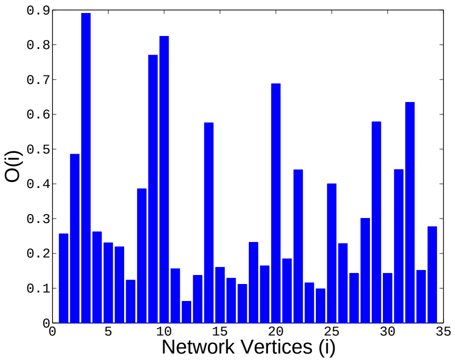

Figure 3: Result of the calculation of the overlapping index for all vertices in the Zachary's "karate club" network.

In order to further verify the effectiveness of our overlapping index measure, we have chosen a non-human social network entitled Dolphin Social Network [16], which is composed of 62 bottlenose dolphins living in Doubtful Sound, New Zealand. In this case, the dolphins are the vertices, with edges between dolphin pairs being established by observation of statistically significant frequent association. Figure 4 indicates the community detection outcome of the proposed technique, along with the 5 most overlapping vertices depicted in larger sizes. In this case, the number of particles that maximizes $\langle R(t) \rangle$ is $K = 2$, which corresponds to the division of the real problem indicated by Lusseau. The communities are identified by the vertices' colors, in this case, blue (gray) and red (dark gray). The split into two groups seems to match the known division of the dolphin community, except for the dolphin "PL", which is member of the blue (gray) community. Lusseau reports that for a period of about two years during observation of the dolphins' behavior, they segregated into two communities, apparently due to the fact of the disappearance of dolphins on the boundary between the communities. When some of these dolphins later reappeared, the two halves of the network joined together once more. Surprisingly, these border dolphins are the ones that the algorithm captured as being the 5 most overlapping nodes, as we can verify in Fig. 4 from the larger vertices, i.e., "DN63", "Knit", "PL", "SN89", and "SN100". As Lusseau draws attention to, developments of this kind illustrate that the dolphin network is not merely a scientific curiosity but, like human social networks, is closely tied to the evolution of the community.

4. Conclusion

This paper proposes a new measure for detecting overlapping structures or vertices in a network via a non-linear stochastic dynamical system, which is biologically inspired by the competition process taking place in many nature and social systems. Furthermore, such measure is embedded in the model, allowing its calculation efficiently. Particularly in this model, several

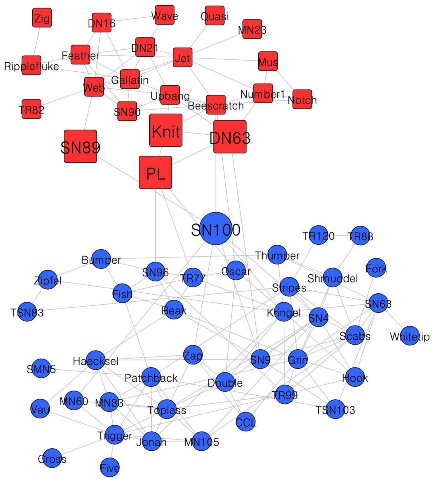

Figure 4: Dolphin Social Network observed by Lusseau. $K = 2$ and $\lambda = 0.6$. We have iterated the dynamical system for 210 steps. The 15 vertices with the highest overlapping structure are depicted with larger size.

particles navigate in the network to explore their territory and, at the same time, attempt to defend their territory from rival particles. Simulations are carried out to show the effectiveness of the model and satisfactory are obtained for real-world data sets.

References

[1] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[2] T. Kohonen. "The self-organizing map". *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[3] B. Kosko. "Stochastic Competitive Learning". *IEEE Trans. Neural Networks*, vol. 2, no. 5, pp. 522–529, 1991.

[4] S. Grossberg. "Competitive learning: From interactive activation to adaptive resonance". *Cognitive Science*, vol. 11, pp. 23–63, 1987.

[5] L. C. Jain, B. Lazzerini and U. H. (eds.). *Innovations in ART Neural Networks (Studies in Fuzziness and Soft Computing)*. Physica-Verlag, Heidelberg, 2010.

[6] G. Deboeck and T. K. (eds.). *Visual Explorations in Finance: with Self-Organizing Maps*. Springer, 2010.

[7] M. G. Quiles, L. Zhao, R. L. Alonso and R. A. F. Romero. "Particle competition for complex network community detection". *Chaos*, vol. 18, no. 3, pp. 033107, 2008.

[8] G. Palla, I. Derényi, I. Farkas and T. Vicsek. "Uncovering the overlapping community structure of complex networks in nature and society". *Nature*, vol. 435, pp. 814–818, 2005.

[9] A. Lancichinetti, S. Fortunato and J. Kertész. "Detecting the overlapping and hierarchical community structure in complex networks". *New Journal of Physics*, vol. 11, no. 3, pp. 033015, 2009.

[10] S. Zhang, R. Wang and X. Zhang. "Identification of overlapping community structure in complex networks using fuzzy cc-means clustering". *Physica A: Statistical Mechanics and its Applications*, vol. 374, no. 1, pp. 483–490, 2007.

[11] E. Çinlar. *Introduction to Stochastic Processes*. Prentice-Hall, Englewood Cliffs, N. J., 1975.

[12] L. Danon, A. Díaz-Guilera, J. Duch and A. Arenas. "Comparing community structure identification". *J. Stat. Mech.*, p. P09008, 2005.

[13] W. W. Zachary. "An information flow model for conflict and fission in small groups". *Journal of Anthropological Research*, vol. 33, pp. 452–473, 1977.

[14] M. Girvan and M. Newman. "Community structure in social and biological networks". *Proc. Natl. Acad. Sci. USA*, vol. 99, no. 12, pp. 7821–7826, 2002.

[15] M. Newman. "Fast algorithm for detecting community structure in networks". *Phys. Rev. E*, vol. 69, no. 6, pp. 066133, 2004.

[16] D. Lusseau. "The emergent properties of a dolphin social network." *Proc Biol Sci*, vol. 270 Suppl 2, pp. S186–S188.