UNCOVERING OVERLAPPING STRUCTURES VIA STOCHASTIC COMPETITIVE LEARNING

Thiago C. Silva1 and Liang Zhao1

1Department of Computer Sciences, Institute of Mathematics and Computer Science (ICMC), University of São Paulo
{thiagoch, zhao}@icmc.usp.br

Resumo: 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 this overlapping index works well in real-world data sets.


1. INTRODUCTION

Overlapping community structure has been widely studied [1]. In [2], 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. Therefore, the detection of the overlapping characteristics of the input data is performed as a separated or dedicated process. The particle competition model was originally proposed in [3], 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. Moreover, we have developed an efficient method for determining the overlapping characteristics of the vertices by using solely the dominance level information generated by the competition process itself, i.e., the determination procedure is already embedded in the model. As a result, it does not increase the model’s complexity order.

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 detection of overlapping structures in real-world 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 $k \in K$, it is basically composed of two distinct types: (i) a random movement term, modeled by the matrix $P^{(k)}_{\text{rand}}$, which permits the particle to adventure through the network, with-
out accounting for the defense of the previously dominated vertices; (ii) a preferential movement term, modeled by the matrix $\mathbb{P}_{\text{pred}}^{(k)}$, 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 $p(t) = [p^{(1)}(t), p^{(2)}(t), ..., p^{(K)}(t)]$, which denotes the localization of the set of $K$ particles presented to the network, where the $k$th-entry, $p^{(k)}(t)$, indicates the location of the particle $k$ in the network at time $t$, i.e., $p^{(k)}(t) \in V, \forall k \in K$.

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, otherwise it is active. In the subsequent step, that particle is automatically reanimated in a vertex that belongs to it in a random manner.

With the intent of modeling such dynamics, we introduce the following stochastic vector $\mathbb{S}^{(t)} = [S^{((1)}(t), ..., S^{(K)}(t))]$, where the $k$th-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, if $S^{(k)}(t) = 0$, 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 $\mathbb{P}_{\text{transition}}^{(k)}(t)$, 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.

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

(1)

where $\lambda \in [0, 1]$ indicates the desired fraction of preferential movement that all particles in the network will perform. $\mathbb{P}_{\text{pref}}^{(k)}(t)$ portrays the transition matrix with a probability distribution according to the preferential behavior described above and, likewise, $\mathbb{P}_{\text{rand}}^{(k)}$ describes the random behavior, $S^{(k)}(t)$ indicates whether particle $k$ is active or exhausted, and $\mathbb{P}_{\text{rean}}^{(k)}(t)$ is responsible for the particle reanimation behavior. Specifically, $\mathbb{P}_{\text{transition}}^{(k)}(i, j, t)$ indicates the probability that particle $k$ makes a transition from vertex $i$ to $j$ at time $t$.

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 $(i, j) \in V \times V$ of the matrix $\mathbb{P}_{\text{rand}}^{(k)}$ is given by:

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

(2)

where $a_{i,j}$ denotes the $(i, j)$th-entry of the adjacency matrix $A$ of the graph.

In order to assist in the calculation of the matrix associated to the preferential movement term, $\mathbb{P}_{\text{pred}}^{(k)}$, for a given particle $k \in K$, we introduce the following stochastic vector: $N_i(t) = [N_i^{(1)}(t), N_i^{(2)}(t), ..., N_i^{(K)}(t)]$, 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 $k$th-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: $N(t) = [N_1(t), N_2(t), ..., N_V(t)]^T$, 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: $\bar{N}_i(t) = [\bar{N}_i^{(1)}(t), \bar{N}_i^{(2)}(t), ..., \bar{N}_i^{(K)}(t)]$, 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 $k$th-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: $\bar{N}(t) = [\bar{N}_1(t), \bar{N}_2(t), ..., \bar{N}_V(t)]^T$, where dim$(\bar{N}(t)) = V \times K$. Mathematically, we define each entry of $\bar{N}_i^{(k)}(t)$ as:

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

(3)

In view of that, we can define $\mathbb{P}_{\text{pref}}^{(k)}(i, j, t)$, 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:

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

(4)

Clearly, from (4), 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 $t$.

Now we define each entry of $\mathbb{P}_{\text{rean}}^{(k)}(t)$ that is accounted for teleporting an exhausted particle $k \in K$ 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:

\[
\mathbb{P}(k_{\text{own}}(t, j, t) = \mathbb{1}\{\arg \max_{m \in \mathbb{K}} \left( N^m_i(t) = k \right)\})
\]

where \( \arg \max (.) \) 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 (5) reveals that the probability of returning to an arbitrary vertex \( j \) dominated by the particle \( k \) follows a uniform distribution. 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.

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

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

where \( \text{owner}(k, t) = \left( \arg \max_{m \in \mathbb{K}} \left( N^m_i(t) = k \right) \right) \) 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 \( t \). 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.

Now we advance to the update rule that governs \( S(t) \). As we have stated, an arbitrary particle \( k \) will be transported back to its domain only if its energy drops to a threshold \( \omega_{\text{min}} \). With that in mind, it is natural that each entry of \( S^{(k)}(t) \) 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 \( k \)th-entry of \( S(t) \) can be precisely written as:

\[
S^{(k)}(t) = \mathbb{1}\{E^{(k)}(t) = \omega_{\text{min}}\}
\]

where \( \dim(S(t)) = 1 \times K \). Specifically, \( S^{(k)}(t) = 1 \) if \( E^{(k)}(t) = \omega_{\text{min}} \) and 0, otherwise.

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:

\[
N^{(k)}(t+1) = N^{(k)}(t) + \mathbb{1}(p^{(k)}(t+1) = i)
\]

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

\[
S^{(k)}(t+1) = \mathbb{1}(E^{(k)}(t+1) = \omega_{\text{min}})
\]

where, by the considerations that we have previously stated, \( \dim(N(t)) = V \times K \), \( \dim(p(t)) = 1 \times K \), and \( \dim(S(t)) = 1 \times K \); and \( \dim(X(t)) = (V + 3) \times K \), with \( N^{(k)}(t) \in [1, \infty), (i, k) \in S \), where \( S \) is the space spawned by \( V \times K \). 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.

2.3. The Initial Conditions of the System

In order to run system \( \phi \), 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. Moreover, \( N(0) \) is initialized following the rule:

\[
N^{(k)}(0) = \begin{cases} 
2, & \text{if} \ \text{particle} \ k \ \text{is generated at vertex} \ i \\
1, & \text{otherwise}
\end{cases}
\]

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 \( k \in \mathbb{K} \) start out with the same energy level given by:

\[
E^{(k)}(0) = \omega_{\text{min}} + \left( \frac{\omega_{\text{max}} - \omega_{\text{min}}}{K} \right)
\]

Lastly, the variable that accounts for indicating whether the particle \( k \) is active or exhausted at the initial step, \( S^{(k)}(0), \forall k \in \mathbb{K} \), is simply given by \( S^{(k)}(0) = 0 \), 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 \( N(t) \) can be used to indicate whose vertices are members of just one or several groups or communities by only analyzing the gap between the two highest domination levels imposed on that vertex. Mathematically, we model this behavior as follows: let \( M_i(x, t) \) denote the \( x \)th 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:
\[ O_i(t) = 1 - (M_i(1, t) - M_i(2, t)) \]  \hspace{1cm} (11)

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 \( O_i(t) \).

2.5. Simulations for Detecting Overlapping Vertices and Communities

We apply our technique to a real-world data set entitled Zachary’s “karate club” network [4]. This is a well-known network from the social science literature, which has become a benchmark test for community detection algorithms. Figure 1 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 [5]) and 10 (e.g., see [6]) 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 (11), on every vertex of the Zachary’s “karate club” network. This result is shown in Fig. 2. 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. 1, since these are placed in the borders of each community.

3. CONCLUSIONS

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. Computer simulations show promising results obtained from a real-world data set.

ACKNOWLEDGEMENT

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

REFERENCES