A Dynamical Model for Community Detection in Complex Networks

Marcos G. Quiles, Ezequiel R. Zorzal and Elbert E.N. Macau

Abstract—One important feature observed in several complex networks is the presence of modular structures, or communities. Detecting communities is still a big challenge for researchers, specially the development of models to deal with dynamic networks. Here, we propose a new method for detecting communities by using a dynamical model. The first step consists of generating a spatial representation, named particle, for each vertex in the network. With these two representation, network structure and the spatial particles, we define the model’s dynamics by means of two interactions types: the first is related to the network structure, or relational, and it is responsible for approaching particles representing neighbor vertices; the second, repulsive, is generated according to the spatial position of each particle and is responsible to make each unrelated particle, according to the network structure, to repel each other. Thus, after a couple of iteration, we observe the formation of groups of particles representing communities. On the other hand, distinct communities are separated according to the spatial positions of their particles. Simulation results show that our model achieves good results on the two benchmark models taken into account and that it can also deal with dynamic networks owing to its intrinsic dynamics.

I. INTRODUCTION

In the last decade of the twentieth century, with the publication of new findings related to the concept of networks, a new research topic, named complex networks, was established and has revolutionized not only the theories of networks (graphs) but also various sciences [1]–[4]. Complex network are everywhere, ranging from society to science, several complex systems such as social interaction, disease propagation, etc. can be modeled by a complex network. Vertices and links, representing the units of the system and their relation, respectively, compose these networks. For example, in a social network, each vertex represents a person, while links define the relation between people.

A remarkable feature observed in several complex networks is the presence of modular structures, or communities, [5]–[8]. Such communities can be defined as groups of densely connected vertices, while connections between vertices belonging distinct communities are, proportionally, more sparse. These communities represent interaction patterns between vertices of the network, and their identification is important in understanding the formation and growth mechanisms of the network [9]. Moreover, an important element regarding the structure of communities is the similarity of the characteristics of the vertices that compose them [10]. Thus, through the identification and study of communities it is possible to obtain information relevant to the network domain. For example, by observing the structure of links between pages of the world wide web one can see that pages describing related topics tend to be more densely connected among themselves than with the rest of the network [11].

This property is also shared by real networks from other domains, such as biological networks [12], metabolic networks [13], etc. This property has also been used to develop graph-based machine learning algorithms [14], [15]. In these algorithms, firstly, the dataset is represented into a network; after, the obtained community structure of the network is used to solve both, clustering and classification problems.

Detecting communities in a network is not a trivial task. For example, the problem of dividing a graph into two parts of equal size in such a manner that the number of links connecting these two parts is minimal is defined as a $NP$-Complete problem [6]. To complicate things further, this splitting problem can be seen as a simple case of the community detection problem, real networks may consist of an arbitrary number of communities and with several sizes. Moreover, communities themselves may also be defined by a hierarchical structures in which a community is formed by other sub-communities [8]. Given the importance of the problem, several models have been proposed recently [16]–[23]. For a review and comparative study of various algorithms, see [6], [8], [24].

Although several models have been proposed, the issue of dealing with dynamic communities has been neglected [8]. Real networks are not static, but their sizes and structures can evolve continually, i.e. if we take any social network, such as one provided by the facebook, new users (vertices) are added or removed daily, new connections are formed or eliminated, etc. In order to deal with these networks most authors use the following methodology: a static snapshot of the network at time $t$ is obtained and their communities detected. After any change in the network, another snapshot is taken at $t + \delta$ and the algorithm is reapplied. Thus, the former detection process is not taken into account. Basically, it can be seen as a continuous application of the algorithm to the network in several instants.

Here, we propose a new algorithm for detecting communities in complex network following a particle moving approach. In our model, for each vertex we create an associated spatial particle. By using these two representations, relational network and spatial particles, we develop a dynamical model capable of clustering the particles representing the commu-
Fig. 1. Illustration of the community detection process. (a) depicts the network used in the simulation; b) the spatial representation of each particle representing the vertices in a); c) a snapshot at $t = 3$; d) a snapshot at $t = 5$; and e) a snapshot at the equilibrium state.

The first step of our model consists of creating a spatial representation for each network vertex, named particle. Taking these two representations into account, the relational (the network) and spatial (particles), two types of interactions are defined. The **Relational Interaction**: it is defined according to relational structure of the network, it represents an attractive interaction responsible for approximating the particles into the space. The **Spatial Interaction**: it is a repulsive interaction defined as a function of the distance between particles representing not connected vertices. This interaction, in contrast to the relational one, is responsible for making each pair of unlinked particle to repel each other.

Generally speaking, the dynamics of these two interactions can be understood as follows. Particles representing densely connected vertices, or communities, tend to become closer into the space. On the other hand, the negative spatial interaction makes each unrelated particle, according to the network structure, to move away from each other. Thus, after a couple of iterations, we can observe the formation of several conglomerates of particles. These conglomerates
represent the communities of the network.

Figure 1 illustrates the whole process. Starting with a network (a), a spatial representation is created in which each vertex is randomly inserted into the space (b). Figures 1(c)-(d) depict snapshots of the process at iteration 3 and 5 while Fig. 1(e) shows the equilibrium state. Through visual inspection of Fig. 1(e), it is possible to observe four groups of particles, which represent the existent communities in (a). In order to detect the communities automatically, one needs to highlight each of these groups by using any clustering algorithm. It is worthy noting that due to the effective separation of the groups provided by our model, any simple clustering algorithm will succeed well.

Formally, for each vertex $v_i$ in the network, we generate a particle $x_i$ randomly positioned into a 3D Euclidean space:

$$x_i(0) = (\text{rand}(), \text{rand}(), \text{rand}())^T \forall v_i \in G$$  \hfill (1)

A 3D space was set empirically albeit other dimensions could be considered. The dynamics of each particle $x_i$ is governed by the following equation:

$$\frac{dx_i}{dt} = \frac{\beta R_i - \alpha A_i}{k_i}$$  \hfill (2)

where $k_i$ defined the degree of vertex $v_i$, $\alpha$ and $\beta$ are variables related to strength of the relational and spatial interactions, respectively. The Relational Interaction (attractive) is defined as follows:

$$A_i = \sum_{j \in \Omega_i} \frac{(x_i - x_j)}{\|x_i - x_j\|}$$  \hfill (3)

where $\Omega_i$ is the adjacency list of vertex $v_i$. The Spatial Interaction (repulsive) is modeled by the equation:

$$R_i = \sum_{j=1,j \neq i}^{N} \frac{(x_i - x_j)}{\|x_i - x_j\|} e^{(-\gamma \|x_i - x_j\|)}$$  \hfill (4)

which defines that the closer two particles are, the stronger is the repulsive interaction between them. Parameter $\gamma$ represents the decaying of the proximity function according to the distance between particles.

Governed by Equations (2)-(4), our model is able to group particles representing densely connected vertices (see Fig. 1). In order to detect the communities automatically, here, parallel to the model dynamics, we use a simple clustering algorithm based on a centroid-seed approach, as explained bellow. This clustering algorithm is similar to the $k$-means clustering algorithm, but with a $k$ varying dynamically. It can be explained as follows:

After a transient period, responsible the initial self-organization of the groups, we start adding seeds $s_k$ into the space:

$$s_k(t) = (\text{rand}(), \text{rand}(), \text{rand}())^T$$  \hfill (5)

These seeds $s_k$ are used to identify the communities according to their membership. To identify to which community each vertex belongs, a variable $y_i$ is defined as its label, i.e. if $y_i = 1$, it means that vertex $v_i$, associated to particle $x_i$, belongs to the community number 1, or associated to the seed $s_1$.

To assign each particle to its community, we evaluate the distance from the particle $x_i$ to all existent seeds at time $t$, as follows:

$$y_i(t) = \arg \min_k \|x_i(t) - s_k(t)\|$$  \hfill (6)

After, the error of each seed is calculated:

$$E(s_k) = \sum_{\forall \in \Delta_k} \|x_i(t) - s_k(t)\|$$  \hfill (7)

where $\Delta_k$ represent the set of particles associated to the seed $s_k$ and $|\Delta_k|$ inform the number of particles in $\Delta_k$. Here, two conditions are checked:

1) If any seed has error zero:

$$E(s_k) = 0$$

it represents that this seed is associated to zero or only one particle. Consequently, the seed is removed;

2) If any seed has error greater than a threshold $\theta_s$:

$$E(s_k) \geq \theta_s$$

it means that this seed is associated to a highly heterogeneous cluster, or group of particles, which indicates that a new cluster must be created. Thus, a new seed is inserted. The threshold $\theta_s$ defines the maximum heterogeneity allowed in each cluster.

If any seed is removed or added, the particles are reassign to the seeds (Eq. 6). Finally, the centroids are recomputed following Equation (8):

$$s_j(t) = \frac{1}{|\Delta_j|} \sum_{\forall \in \Delta_j} x_i(t)$$  \hfill (8)

The overall algorithm is summarized in Algorithm 1.

**Algorithm 1 Community Detection Algorithm**

Load network  
Start particles  
Set parameters

while $(\Delta R(t) > \theta_r)$ do  
Run Equations (2)-(4)  
if $t > $ transient then  
Assign each particle to its closest seed:  
Check removal condition (1)  
Check insertion condition (2)  
Recompute position of the seeds with Eq. (8)  
end if  
end while

The stop criterion used in Alg. 1, $\Delta R > \theta_r$, is defined by the following equation:
III. COMPUTER SIMULATIONS

For the sake of comparison, we evaluate our model by using the same methodology considered in [24]. First, a set of experiments using the Girvan-Newman benchmark networks [5] is conducted. After, we evaluate its performance on the LFR benchmark [25] which provides more realistic networks. The community structure of networks in these two benchmarks are controlled by a parameter, named mixing parameter $\mu$, which defines the portion of links of each vertex that connect it to communities belonging to other communities. It means, if $\mu = 0.0$, the communities are completely isolated, or there are no inter-links. On the other hand, if $\mu = 0.5$, half of its links connects the vertex to vertices belonging to its own community, while the remaining links connect the vertex to vertices of other communities. To calculate the accuracy of each algorithm we use the normalized mutual information ($I$) measure presented in [6]. This quality measure is calculated by using a confusion matrix $N$, where rows correspond to the expected community result and the columns correspond to the obtained community structure. The normalized mutual information measure is calculated according to the following equation:

$$I = \frac{-2 \sum_{i=1}^{M_R} \sum_{j=1}^{M_F} N_{ij} \log (N_{ij}/N_{i}N_{j})}{\sum_{i=1}^{M_R} N_{i} \log (N_{i}/N) + \sum_{j=1}^{M_F} N_{j} \log (N_{j}/N)}$$

where $M_R$ and $M_F$ correspond to the number of expected and found communities, respectively. $N_{ij}$ represents the number of vertices belonging to the real community $i$ but clustered with community $j$ according to the algorithm outcome. $N_{i}$ defines the sum over row $i$ of matrix $N$, while $N_{j}$ corresponds to the sum over the column $j$ of matrix $N$. $N$ it self, represent the confusion matrix and also the number of vertices in the network (see Eq. (10)).

Before presenting the simulations and results of the community detection process, we will present some considerations about the implementation of our model and the parameters used in the simulations.

A. Model Parameters

To integrate the Equation (2) numerically, we follow a coarse discretization by taking $\frac{d\tau}{dt} \approx \frac{\Delta t}{\Delta t}$ with $\Delta t = 1$. 

\[
\Delta R(t) = \frac{1}{N} \left| \sum_i R_i(t) - \sum_i R_i(t-1) \right| 
\]

it considers the instantaneous variation of the average of the negative interaction of particles. If the variation of the negative interaction of particles at $t$ is below to a certain threshold $\theta_e$, thus an equilibrium state have been reached and the dynamics can be stopped, providing the community structure of the network. The transient period is used to guarantee that the detection process starts only when a certain equilibrium has been achieved and not at the initial random condition ($t = 0$).
Basically, we use the Euler method with step $\Delta t = 1.0$. It is worthy noting that the results obtained with this discrete version is totally compatible to the original model albeit much faster. To implement our model we use the igraph package [26].

![Graph showing normalized mutual information as a function of the mixing parameter.](image)

Fig. 3. Simulations on the GN benchmark. Each plot depicts the normalized mutual information as a function of the mixing parameter $\mu$ for all algorithms considered in this work. Each point represents the average of 200 realizations.

Parameters $\alpha$ and $\beta$ represents the strength of the attractive and repulsive interactions, respectively. To set these parameters, exhaustive experiments were performed taking several network topologies into account. We observe that there are no single pair of values for these parameters, but a set of values. For each value of $\beta$ one can set a good value of $\alpha$. To find suitable values of $\alpha$ for a given value of $\beta$, $\alpha$ can be set in order to minimize the mean error of the seeds (see Eq. (7)). For instance, by fixing $\beta = 0.2$, we found that good community detection rates to all topologies taken into account can be achieved by setting $\alpha \in [0.7, 0.9]$. It is worthy noting that the outcome is not so sensitive to these values.

In all experiments presented in this work we hold $\alpha$ constant at 0.8. If much lower values are assumed, the attractive interaction is reduced and groups of particles are barely observed owning to the repulsive interaction. On the other hand, if $\alpha$ is increased to values higher than 0.9, the attraction between particles is boosted and communities start merging. Thus, by varying parameter $\alpha$ while holding $\beta$ fixed, we can detect the hierarchical structure of the network, whether it exists. In Figure 2, we depict the community detection outcome for the network (adjacency matrix) presented in Fig. 2(a) for several values of $\alpha$. One can see that for small values of $\alpha$, eight groups (communities) were detected (c). By increasing $\alpha$, the model detects four, two and one communities, depicted in (d)-(f), respectively.

The other parameters, $\gamma$, $\theta_r$, and $\theta_s$ were held constant at: 1.0, $10^{-2}$, and 0.5, respectively. Those parameters do not need further adjustment for specific simulations.

B. Community Detection

The first set of simulations was carried out with the GN networks. Figure 3(a) shows the results obtained with our model by varying the mixing parameter $\mu = 0.0, 0.6$. Fig. 3(b), depict the results provided by state of art algorithms: GN [5], CNM [27], InfoMAP [28], RAK [29], PL [30], and EigenV [31]. It is possible to observe that our model overcome most of the algorithms taken into account.

Next, we apply all algorithms used in the previous simulation to the LFR benchmark [25]. As mentioned before, we use the same parameters considered in [24] to set the LFR networks: the average degree at 20, the maximum degree at 50, the exponent of the degree distribution at $-2.0$, and the exponent of the community size distribution at $-1.0$. With these parameters, four scenarios are considered: a) Networks with 1000 vertices and community sizes varying from 10 to 50 vertices, named small (S); b) Networks with 1000 vertices and communities varying from 20 to 100, named big (B); c) and d) follow the same range for the community size of a) and b) but with networks with 5000 vertices.

Figure 4 depicts the results obtained with all considered algorithms by varying the mixing parameter from 0.1 to 0.8. Results are presented to the four scenarios, except to the GN algorithm, which has a high computational cost making simulations with 5000 vertices prohibitive. Contrasting Fig. 4(a) to (b)-(f), it is possible to observe that our model produces a good results, specially when the communities are barely observed, which means, to higher values of the mixing parameter $\mu \approx 0.8$. In this range, our model overcome all algorithms on average, except the InfoMAP [28] on the scenario c) 5000 vertices and small communities. The InfoMAP [28] and the PL [30] algorithms produced better results than our model with $\mu$ set to lower values.

A final simulation is carried out in order to illustrate our model dealing with a dynamical situation. We start with a GN Network with 128 vertices and the mixing parameter $\mu = 0.1$, see Figure 5(a). Fig. 5(b) depicts the initial spatial configuration. Fig. 5(c) shows the spatial configuration in the equilibrium state, where four communities can be observed. In this simulation we did not stop the model dynamics when the equilibrium state was reached. It was done purposely to
check how does the model behave when facing a change in the structure of the network. At \( t = 100 \), we add 200 new links between communities “blue” and “cyan” with the main purpose of balancing their number of intra and inter community links. Thus, after this modification on the network structure, both communities, “blue” and “cyan”, can be considered as a single merged community with 64 vertices, see Fig. 5(d). Figures 5(e)-(i) show the spatial configuration at \( t = 101 \), \( t = 103 \), \( t = 107 \), \( t = 110 \), and \( t = 120 \), respectively. Thus, it is possible to observe that a new equilibrium is reached after a network change. Moreover, the new community structure detected by our
model is not obtained by running the algorithm from the initial condition, but starting from the previous equilibrium observed at $t = 100$. It is worthy noting that our model was not applied to a sequence of static network, but it was kept running on real-time during the network modification.

IV. CONCLUDING REMARKS

In summary, we have proposed a new dynamical model to detect communities in complex networks. The dynamics is governed by two interaction types, named relational and spatial. These two interactions are modeled according to the network structure and to the spatial particle representation proposed here. Specifically, the relational interaction is defined according to the network structure and tends to make neighbor vertices to become closer into the space. The spatial interaction, on the other hand, is responsible to repel unlinked particles. Our model reaches an equilibrium state when the variation of the repulsive interaction is reduced to values below a certain threshold. At this point, particles representing densely connected vertices, or communities, have converged to regions of the space forming groups of related particles. By applying a simple clustering algorithm into the space, as the one that has been proposed here, we were able to detect the community structure of the network with a high accuracy.

The main contributions of our model are: 1) it is dynamical, it means, if any change occurs in the network, such as the inclusion or removal of any vertex or link, it can be interpreted as a perturbation into the dynamical model, which will reach a new equilibrium state after a new transient. In this way, we do not need to reapply our model to detect
the community structure of network when changes happen. Most of community models found in the literature are not able to deal with time-varying networks dynamically and the algorithm must be reapplied every time a change is observed; 2) by adjusting the relational (positive) strength parameter, \( \alpha \), we are able to detect community hierarchically, whether they exist, which is another interesting outcome of the proposed model.

As a future work, we intend to conduct a formal analysis of our model, by studying its convergence and stability as also a deeper study of its parameters, \( \alpha \) and \( \beta \), and their relations to the model dynamics. Moreover, we will also carry out a new set of simulations with synthetic and real dynamic networks and also with weighted directed and undirected networks.

ACKNOWLEDGMENTS

This research was supported by São Paulo Research Foundation (FAPESP) and by the the Brazilian National Research Council (CNPq). The authors thank Dr. Eudes Fileti for his critical review and comments.

REFERENCES


