Chapter 9
Case Study of Network-Based Unsupervised Learning: Stochastic Competitive Learning in Networks

Abstract Many business and day-to-day problems that arise in our lives must be dealt with under several constraints, such as the prohibition of external interventions of human beings. This may be due to high operational costs or physical or economical impossibilities that are inherently involved in the process. The unsupervised learning—one of the existing machine learning paradigms—can be employed to address these issues and is the main topic discussed in this chapter. For instance, a possible unsupervised task would be to discover communities in social networks, find out groups of proteins with the same biological functions, among many others. In this chapter, the unsupervised learning is investigated with a focus on methods relying on the complex networks theory. In special, a type of competitive learning mechanism based on a stochastic nonlinear dynamical system is discussed. This model possesses interesting properties, runs roughly in linear time for sparse networks, and also has good performance on artificial and real-world networks. In the initial setup, a set of particles is released into vertices of a network in a random manner. As time progresses, they move across the network in accordance with a convex stochastic combination of random and preferential walks, which are related to the offensive and defensive behaviors of the particles, respectively. The competitive walking process reaches a dynamic equilibrium when each community or data cluster is dominated by a single particle. 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. In this case, each vertex corresponds to a data item and pairwise connections are established using a suitable network formation process.

9.1 A Quick Overview of the Chapter

Competition is a natural process observed in nature and in many social systems that have limited resources, such as water, food, mates, territory, recognition, etc. Competitive learning is an important machine learning approach that is widely employed in artificial neural networks to realize unsupervised learning. Early developments include the famous self-organizing map (SOM—*Self-organizing Map*) [19], differential competitive learning [20], and adaptive resonance theory (ART—*Adaptive Resonance Theory*) [6, 14]. From then on, many competitive

T.C. Silva, L. Zhao, *Machine Learning in Complex Networks*,
DOI 10.1007/978-3-319-17290-3_9

learning neural networks have been proposed [1–3, 16, 17, 24, 25, 28, 31, 39] and a wide range of applications has been considered. Some of these application include data clustering, data visualization, pattern recognition, and image processing [4, 7, 9, 10, 22, 41]. Without a doubt, competitive learning represents one of the main successes of the unsupervised learning development.

The network-based unsupervised learning technique that we present here is one type of competitive learning process. In essence, the model relies on a competitive mechanism of multiple homogeneous particles originally proposed in [32]. Thereafter, the particle competition technique has been enhanced and formally modeled by a stochastic nonlinear dynamical system and applied to data clustering tasks in [35]. In this chapter, we explore the particle competition algorithm by providing several empirical and analytical analyses. In this investigation, we attempt to show the potentialities and shortcomings of the particle competition technique. Given that the models of interactive walking processes correspond to many natural and artificial systems, and due to the relative lack of theory for such systems, the analytical analysis of this model is an important step to understanding such systems.

Once the fundamental idea and the model definition are properly presented, several applications that use the particle competition model are discussed in various interesting problems indicated in the literature. One of these problems is the creation of efficient evaluation indices for estimating the most likely number of clusters or communities in data sets. We show that these indices explore dynamic variables that are constructed from the competitive behavior of the particles inside the network. In this way, the evaluation of these indices is embedded within the mechanics of the particle competition process. As a result, if one takes into account that the number of clusters is far less than the quantity of data items, the process of determining the most likely number of clusters does not increase the model's time complexity order. Since the determination of the actual number of clusters is an important issue in data clustering [38, 40], the particle competition model also presents a contribution to this topic.

Following the same line, an index for detecting overlapping cluster structures is also discussed, which, under some assumptions, may also not increase the model's time complexity order due to its embedded nature within the competitive process.

With all these tools at hand, the chapter is finalized by investigating how the model behaves in an application of handwritten digits and letters clustering. Therein, we see that the competitive model is able to satisfactorily cluster several variations and distortions of the same handwritten digits and letters into their corresponding clusters.

9.2 Description of the Stochastic Competitive Model

In this section, the competitive dynamical system consisting of multiple particles [35] is discussed.

Section 9.2.1 provides the intuition behind the mechanics of the model. Section 9.2.2 builds on the caveats for constructing the transition matrix of the stochastic dynamical system that the particle competition model relies on. Section 9.2.3 formally defines the corresponding dynamical system. Section 9.2.4 explores the application of estimating the most likely number of communities or groups in a data set. Section 9.2.5 introduces another application of detecting overlapping vertices and communities. Section 9.2.6 supplies a parameter sensitivity analysis of the model's parameters. Finally, Sect. 9.2.7 analyzes convergence issues of the particle competition algorithm.

9.2.1 Intuition of the Model

Consider a network $\mathscr{G} = \langle \mathscr{V}, \mathscr{E} \rangle$, where $\mathscr{V}$ is the set of vertices and $\mathscr{E} \subset \mathscr{V} \times \mathscr{V}$ is the set of links (or edges). There are $V = |\mathscr{V}|$ vertices and $E = |\mathscr{E}|$ edges in the network. In the competitive learning model, a set of particles $\mathscr{K} = \{1, \ldots, K\}$ is inserted into the vertices of the network in a random manner. Essentially, each particle can be conceived as a flag carrier whose goal is to conquer new vertices, while defending its current dominated vertices. Given that we have a finite number of vertices, competition among particles naturally occurs. Note that the vertices play the role as valuable resources in this competition process. When a particle visits an arbitrary vertex, it strengthens its own domination level on that vertex and, at the same time, weakens the domination levels of all of the other rival particles on the same vertex. Finally, it is expected that each particle will be confined within a subnetwork corresponding a community. In this way, the communities are uncovered. Figure 9.1a, b portray a possible initial condition, in which particles are randomly inserted into network vertices, and the expected long-run dynamic of the particle competition system for an artificial clustered network with three well-defined communities.

Due to the competition effect, a particle is either in the *active* or in the *exhausted* state. Whenever the particle is active, it navigates in the network guided by a combination of two orthogonal walking rules: the random and the preferential movements. The random walking term permits particles to randomly visit neighboring vertices regardless of their current conditions and the neighborhood. Therefore, the random walking term is an unconditional rule that depends only on the immutable network topology and hence is responsible for the particle's exploratory behavior. On the contrary, the preferential walking term accounts for the defensive behavior of the particles by favoring particles to revisit and reinforce their dominated territory rather than to visit non-dominated vertices. This walking term is a conditional rule that depends on the particles' domination levels on the neighborhood. Therefore, while the movement distribution that models the exploratory behavior is fixed, that distribution that describes the defensive behavior is mutable, being dependent both on the particles and the time dimension.

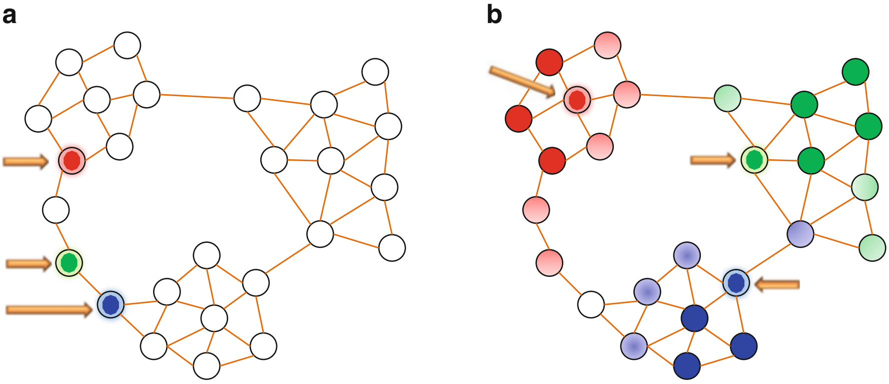

Fig. 9.1 Illustration of the initial conditions and long-run dynamic of the particle competition model. (**a**) Possible initial setup, (**b**) expected long-run dynamic

In the particle competition process, each particle carries a time-dependent energy variable that reflects its instantaneous exploration ability. The energy variable increases when the particle is visiting a vertex that it dominates, and decreases whenever it visits a vertex dominated by a rival particle. If the energy variable drops below a minimum threshold, the particle becomes exhausted and it is brought back to one of the vertices that it dominates. With this mechanism, the network always has a constant number of particles and frequent intrusions of particles to regions dominated by rival particles can be avoided. The exhaustion of particles in the learning process can be related to the smoothness assumption of unsupervised learning algorithms, because this process delimits the community borders that each particle dominates.

9.2.2 Derivation of the Transition Matrix

During the competition process, each particle $k \in \mathscr{K}$ performs two distinct types of movements:

- a *random movement term*, modeled by the matrix $\mathbf{P}^{(k)}_{\text{rand}}$, which allows the particle to venture throughout the network, without accounting for the defense of the previously dominated vertices; and
- a *preferential movement term*, modeled by the matrix $\mathbf{P}^{(k)}_{\text{pref}}$, which is responsible for inducing the particle to reinforce vertices that the particle dominates, effectively creating a preferential visiting rule to dominated vertices rather than random ones.

Consider the random vector $p(t) = [p^{(1)}(t), p^{(2)}(t), \ldots, p^{(K)}(t)]$, which denotes the location of the set of K particles presented to the network. Its k-th entry, $p^{(k)}(t)$, indicates the location of particle k in the network at time t, i.e., $p^{(k)}(t) \in \mathscr{V}, \forall k \in \mathscr{K}$. With the intent of keeping track of the current states of all particles, we introduce the random vector $S(t) = [S^{(1)}(t), \ldots, S^{(K)}(t)]$, where the k-th entry, $S^{(k)}(t) \in \{0, 1\}$, indicates whether particle k is active ($S^{(k)}(t) = 0$) or exhausted ($S^{(k)}(t) = 1$) at time t. When a particle is active, it performs the combined random-preferential movements; when it is exhausted, the particle switches its movement policy to a new transition matrix, here referred to as $\mathbf{P}^{(k)}_{\text{rean}}(t)$. This matrix is responsible for taking the particle back to its dominated territory, in order to reanimate the corresponding particle by recharging its energy. This sequence of steps is called the *reanimation procedure*. After the particle's energy has been properly recharged, it again walks in the network. With these notations at hand, we can define a transition matrix that governs the probability distribution of the movement of the particles to the immediate future state $p(t+1) = [p^{(1)}(t+1), p^{(2)}(t+1), \ldots, p^{(K)}(t+1)]$ as follows:

$$\mathbf{P}^{(k)}_{\text{transition}}(t) \triangleq (1 - S^{(k)}(t)) \left[\lambda \mathbf{P}^{(k)}_{\text{pref}}(t) + (1-\lambda)\mathbf{P}^{(k)}_{\text{rand}}\right] + S^{(k)}(t)\mathbf{P}^{(k)}_{\text{rean}}(t), \quad (9.1)$$

in which k is the particle index, $\lambda \in [0, 1]$ modulates the desired fraction of preferential and random movements. Larger values of λ favor preferential walks in detriment to random walks. The entry $\mathbf{P}^{(k)}_{\text{transition}}(i, j, t)$ indicates the probability that particle k performs a transition from vertex i to j at time t. Now we define the random and the preferential movement matrices.

Each entry $(i, j) \in \mathscr{V} \times \mathscr{V}$ of the random movement matrix is given by:

$$\mathbf{P}^{(k)}_{\text{rand}}(i, j) \triangleq \frac{\mathbf{A}_{ij}}{\sum_{u \in \mathscr{V}} \mathbf{A}_{iu}}, \quad (9.2)$$

in which $\mathbf{A}_{ij}$ denotes the (i, j)-th entry of the adjacency matrix $\mathbf{A}$ of the network. It means that the probability of an adjacent neighbor j to be visited from vertex i is proportional to the edge weight linking these two vertices. The matrix is time-invariant and it is the same for every particle in the network; therefore, whenever the context makes it clear, we drop the superscript k for convenience.

In order to derive the preferential movement matrix, $\mathbf{P}^{(k)}_{\text{pref}}(t)$, we introduce the following random vector:

$$\mathbf{N}_i(t) \triangleq [\mathbf{N}^{(1)}_i(t), \mathbf{N}^{(2)}_i(t), \ldots, \mathbf{N}^{(K)}_i(t)]^T, \quad (9.3)$$

in which $\dim(\mathbf{N}_i(t)) = K \times 1$, T denotes the transpose operator, and $\mathbf{N}_i(t)$ registers the number of visits received by vertex i up to time t by each of the particles in the network. Specifically, the k-th entry, $\mathbf{N}^{(k)}_i(t)$, indicates the number of visits made by particle k to vertex i up to time t. Then, the matrix that contains the number of visits made by each particle in the network to all the vertices is defined as:

$$\mathbf{N}(t) \triangleq [\mathbf{N}_1(t), \mathbf{N}_2(t), \ldots, \mathbf{N}_V(t)]^T, \tag{9.4}$$

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

$$\bar{\mathbf{N}}_i(t) \triangleq [\bar{\mathbf{N}}_i^{(1)}(t), \bar{\mathbf{N}}_i^{(2)}(t), \ldots, \bar{\mathbf{N}}_i^{(K)}(t)]^T, \tag{9.5}$$

in which $\dim(\bar{\mathbf{N}}_i(t)) = K \times 1$ and $\bar{\mathbf{N}}_i(t)$ denotes the relative frequency of visits of all particles in the network to vertex i at time t. In particular, the k-th entry, $\bar{\mathbf{N}}_i^{(k)}(t)$, indicates the relative frequency of visits performed by particle k to vertex i at time t. Then, the matrix form of the domination level of all vertices is defined as:

$$\bar{\mathbf{N}}(t) \triangleq [\bar{\mathbf{N}}_1(t), \bar{\mathbf{N}}_2(t), \ldots, \bar{\mathbf{N}}_V(t)]^T, \tag{9.6}$$

in which $\dim(\bar{\mathbf{N}}(t)) = V \times K$. Mathematically, each entry of $\bar{\mathbf{N}}_i^{(k)}(t)$ is defined as:

$$\bar{\mathbf{N}}_i^{(k)}(t) \triangleq \frac{\mathbf{N}_i^{(k)}(t)}{\sum_{u \in \mathscr{K}} \mathbf{N}_i^{(u)}(t)}. \tag{9.7}$$

With these notations at hand, the preferential movement rule can be defined as:

$$\mathbf{P}_{\text{pref}}^{(k)}(i, j, t) \triangleq \frac{\mathbf{A}_{ij} \bar{\mathbf{N}}_j^{(k)}(t)}{\sum_{u \in \mathscr{V}} \mathbf{A}_{iu} \bar{\mathbf{N}}_u^{(k)}(t)}. \tag{9.8}$$

Equation (9.8) defines 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. It can be observed that each particle has a different transition matrix associated to its preferential movement. Moreover, each matrix is time-varying with dependence on the domination levels of all of the vertices ($\bar{\mathbf{N}}(t)$) at time t. Since the preferential movement term of particles directly depends on their visiting frequency to a specific vertex, as more visits are performed by a particle to a determined vertex, the higher is the chance for that particle to repeatedly visit the same vertex. Furthermore, if the domination level of the visiting particle on a vertex is strengthened, the domination levels of all other particles on the same vertex are consequently weakened. This feature occurs on account of the normalization process in (9.7): if one domination level increases, all of the others must go down, so that the overall sum still produces 1.

For didactic purposes, we now summarize and consolidate the key concepts introduced so far in a simple example given in the following.

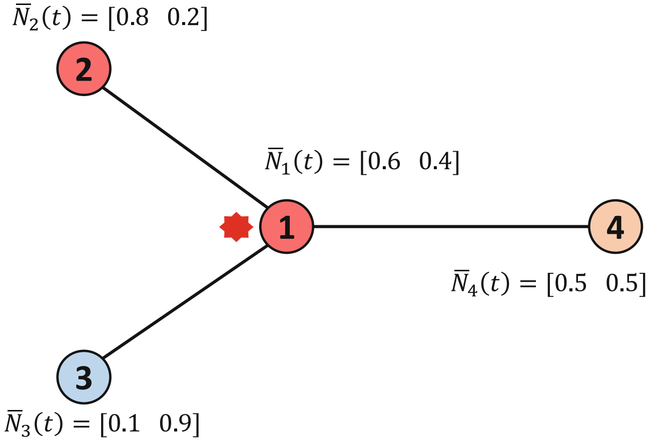
$\bar{N}_2(t) = [0.8 \;\; 0.2]$
2
$\bar{N}_1(t) = [0.6 \;\; 0.4]$
1
4
$\bar{N}_4(t) = [0.5 \;\; 0.5]$
3
$\bar{N}_3(t) = [0.1 \;\; 0.9]$

Fig. 9.2 A typical situation where the *red* (*dark gray*) particle, presently located at vertex 1, has to choose a neighbor to visit in the next iteration. For illustration purposes, the domination level vector for each vertex is displayed, in which the two entries represent the domination levels of the *red* (*dark gray*) and the *blue* (*gray*) particles, in that order. In this example, there are two particles, *red* (*dark gray*) and *blue* (*gray*). The beige (*light gray*) *color* denotes vertices that are not being dominated by any particles in the system at time *t*

Example 9.1. Consider the network portrayed in Fig. 9.2, where there are two particles, namely red (dark gray) and blue (gray), and four vertices. For illustrative purposes, we only depict the location of the red (dark gray) particle, which is currently visiting vertex 1. In this example, the role that the domination level plays in determining the resulting transition probability matrix is presented. Within the figure, we also didactically supply the domination level vector of each vertex at time t. Note that the ownership of the vertex (in the figure, the color of the vertex) is set according to the particle that is imposing the highest domination level on that specific vertex. For instance, in vertex 1, the red (dark gray) particle is imposing a domination of 60 %, and the blue (gray) particle, of only 40 %. The goal here is to derive the transition matrix of the red particle in agreement with (9.1). Suppose at time t the red particle is *active*, therefore, $S^{(\text{red})}(t) = 0$. Consequently, the second term of the convex combination in (9.1) vanishes. On the basis of (9.2), the random movement term of the red particle is given by:

$$\mathbf{P}_{\text{rand}}^{(\text{red})} = \begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \tag{9.9}$$

(continued)

Example 9.1 (continued)
and the preferential movement matrix at the immediate posterior time $t + 1$, according to (9.8), is given by:

$$\mathbf{P}_{\text{pref}}^{(\text{red})}(t+1) = \begin{bmatrix} 0 & 0.57 & 0.07 & 0.36 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \tag{9.10}$$

Finally, the transition matrix associated to the red particle is determined by a weighted combination of the random (time-invariant) and the preferential matrices at time $t + 1$, given that the particle is active (see (9.1)). If $\lambda = 0.8$, then such matrix is given by:

$$\mathbf{P}_{\text{transition}}^{(\text{red})}(t+1) = 0.2 \begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} + 0.8 \begin{bmatrix} 0 & 0.57 & 0.07 & 0.36 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0.52 & 0.12 & 0.36 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \tag{9.11}$$

Therefore, the red particle, which is currently in vertex 1, has a higher chance to visit vertex 2 (52 % chance of visiting) than the others. This behavior can be controlled by adjusting the λ parameter. A large value of λ induces particles to perform mostly preferential movements, i.e., particles keep visiting their dominated vertices in a frequent manner. A small value of λ, in contrast, provides a larger weight to the random movement term, making particles resemble traditional Markovian walkers as $\lambda \to 0$ [8]. In the extreme case, i.e., when $\lambda = 0$, the mechanism of competition is turned off and the model reduces to multiple non-interactive random walks. In this way, we can see that the particle competition model generalizes the dynamical system of multiple random walks, according to the parameter λ.

Now we define $\mathbf{P}_{\text{rean}}^{(k)}(t)$ matrix that is responsible for transporting an exhausted particle $k \in \mathscr{K}$ back to its dominated territory, with the purpose of recharging its energy (reanimation process). Suppose particle k is visiting vertex i when its energy is completely depleted. In this situation, the particle must regress to an arbitrary vertex j of its possession at time t, according to the following expression:

$$\mathbf{P}_{\text{rean}}^{(k)}(i,j,t) \triangleq \frac{\mathbb{1}_{\left[\arg\max\limits_{m\in\mathscr{K}}\left(\bar{\mathbf{N}}_j^{(m)}(t)\right)=k\right]}}{\sum_{u\in\mathscr{V}} \mathbb{1}_{\left[\arg\max\limits_{m\in\mathscr{K}}\left(\bar{\mathbf{N}}_u^{(m)}(t)\right)=k\right]}}, \tag{9.12}$$

in which $\mathbb{1}_{[.]}$ is the indicator function that yields 1 if the argument is logically true and 0, otherwise. The operator $\arg\max\limits_{m\in\mathbb{K}}(.)$ returns an index M, where $\bar{\mathbf{N}}_u^{(M)}(t)$ is the maximal value among all $\bar{\mathbf{N}}_u^{(m)}(t)$ for $m = 1, 2, \ldots, K$. We note that (9.12) reduces to a uniform distribution when we take the subset of vertices that are dominated by particle k. For all of the non-dominated vertices, the transition probability is zero. Observe also that the transition probability is independent of the network topology. If no vertex is being dominated by particle k at time t, we put it in any vertex of the network in a random manner (uniform distribution on the whole network).

Example 9.2. Figure 9.3 illustrates how the reanimation scheme takes place. Consider that the red (dark gray) particle is exhausted possibly because it has visited several non-dominated vertices, which led to the depletion of its energy. The reanimation procedure consists in transporting back that particle to one of its dominated vertices, regardless of the network topology. The intuition of this procedure is that, with a relatively high probability, its energy will be renewed in the next iterations, for the neighborhood is expected to be dominated by the same particle.

Let also the random vector $E(t) = [E^{(1)}(t), \ldots, E^{(K)}(t)]$ represent the energy that each particle holds. In special, its k-th entry, $E^{(k)}(t)$ denotes the energy level of particle k at time t. In view of these definitions, the energy update rule is given by:

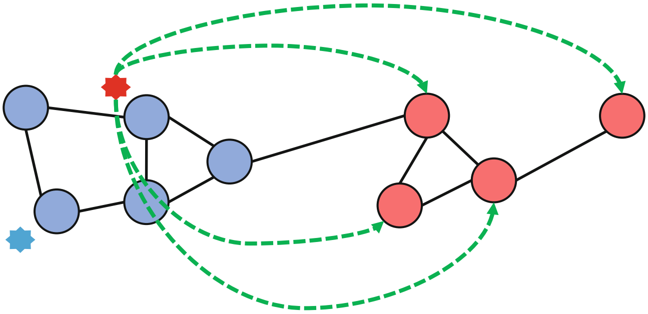

Fig. 9.3 Illustration of the reanimation scheme. The *red* (*dark gray*) particle is exhausted and is forced to be transported back to its dominated territory. The transition probability follows a uniform distribution on the dominated vertices

$$E^{(k)}(t) = \begin{cases} \min(\omega_{\max}, E^{(k)}(t-1) + \Delta), & \text{if owner}(k,t) \\ \max(\omega_{\min}, E^{(k)}(t-1) - \Delta), & \text{if } \neg\text{owner}(k,t) \end{cases} \tag{9.13}$$

in which the parameters $\omega_{\min}$ and $\omega_{\max}$ characterize the minimum and maximum energy levels, respectively, that a particle may possess. Therefore, $E^{(k)}(t) \in [\omega_{\min}, \omega_{\max}]$. The owner(k, t) is defined as:

$$\text{owner(k, t)} = \left(\arg \max_{m \in \mathscr{K}} \left(\bar{\mathbf{N}}^{(m)}_{p^{(k)}(t)}(t) \right) = k \right) \tag{9.14}$$

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 that same visiting particle, and results in a logical false otherwise; $\dim(E(t)) = 1 \times K$; $\Delta > 0$ symbolizes the increment or decrement of energy that each particle receives at time t. The first expression in (9.13) represents the increment of the particle's energy and occurs whenever particle k visits a vertex $p^{(k)}(t)$ that it dominates, i.e., $\arg \max_{m \in \mathbb{K}} \left(\bar{\mathbf{N}}^{(m)}_{p^{(k)}(t)}(t) \right) = k$. Similarly, the second expression in (9.13) indicates the decrement of the particle's energy that happens when it visits a vertex dominated by rival particles. Therefore, in this model, particles are given a penalty if they are wandering in rival territory, so as to minimize aimless navigation trajectories in the network.

The term $S(t)$ is responsible for determining the movement policy of each particle at each time t. It is really a switching function and defined as follows:

$$S^{(k)}(t) = \mathbb{1}_{\left[E^{(k)}(t) \,=\, \omega_{\min}\right]}, \tag{9.15}$$

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

In the following, we apply the concepts introduced so far in a concise and simple example.

Example 9.3. Consider the network depicted in Fig. 9.4. Suppose there are two particles, namely, red (dark gray) and blue (gray), each of which located at vertices 13 and 1, respectively. As both particles are visiting vertices whose owners are rival particles, their energy levels drop. Consider, in this case, that both particles have reached the minimum allowed energy, i.e., $\omega_{\min}$, at time t. Therefore, according to (9.15), both particles are *exhausted*. Consequently, $S^{(\text{red})}(t) = 1$ and $S^{(\text{blue})}(t) = 1$, and the transition matrix associated to each particle reduces to the second term in the convex combination of (9.1). According to the mechanism of the dynamical system, these particles are

(continued)

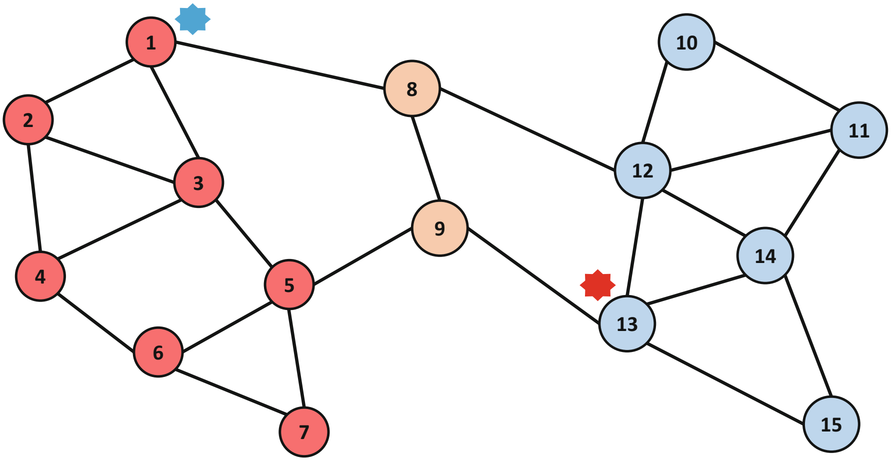

Fig. 9.4 Illustration of the reanimation procedure in a typical situation. There are two particles, namely *red* (*dark gray*) and *blue* (*gray*), located at vertices 13 and 1, respectively, at time *t*. The network encompasses 15 vertices. As both particles are visiting vertices whose owners are rival particles, their energy levels drop. In this example, suppose both energy levels of the particles reach the minimum possible value, ω_{min}. The vertex *color* represents the particle that is imposing the highest domination level at time *t*. The beige (*light gray*) denotes a non-dominated vertex

Example 9.3 (continued)
transported back to their dominated territory to recharge their energy levels. The transition occurs regardless of the network topology. This mechanism follows the distribution in (9.12). In view of that, the transition matrix for the red (dark gray) particle at time *t* is:

$$\mathbf{P}^{(\text{red})}_{\text{transition}}(i,j,t) = \frac{1}{7}, \forall i \in \mathscr{V}, j \in \{v_1, v_2, \ldots, v_7\}, \tag{9.16}$$

$$\mathbf{P}^{(\text{red})}_{\text{transition}}(i,j,t) = 0, \forall i \in \mathscr{V}, j \in \mathscr{V} \setminus \{v_1, v_2, \ldots, v_7\}, \tag{9.17}$$

and the transition matrix associated to the blue (gray) particle at time *t* is written as:

$$\mathbf{P}^{(\text{blue})}_{\text{transition}}(i,j,t) = \frac{1}{6}, \forall i \in \mathscr{V}, j \in \{v_{10}, v_{11}, \ldots, v_{15}\}, \tag{9.18}$$

$$\mathbf{P}^{(\text{blue})}_{\text{transition}}(i,j,t) = 0, \forall i \in \mathscr{V}, j \in \mathscr{V} \setminus \{v_{10}, v_{11}, \ldots, v_{15}\}. \tag{9.19}$$

One can verify that exhausted particles are transported back to their territory (set of dominated vertices) regardless of the network topology. The determination of which of the dominated vertices to visit follows a uniform distribution. In this way, vertices are equally probable to receive only those particles that dominate them.

Looking at (9.1), we see that each particle has a representative transition matrix. For compactness, we can join all of them together into a single representative transition matrix that we refer to $\mathbf{P}_{\text{transition}}(t)$, which models the transition of the random vector $p(t)$ to $p(t+1)$. This global matrix will prove useful in Sect. 9.3. Given the current system's state at time t, one can see that $p^{(k)}(t+1)$ and $p^{(u)}(t+1)$ are independent for every pair $(k, u) \in \mathscr{K} \times \mathscr{K}, k \neq u$. Another way of looking at this fact is that, given the immediate past position of each particle, it is clear that, via (9.1), the next particle's location is only dependent on the topology of the network (random term) and the domination levels of the neighborhood in the previous step (preferential term). In this way, $\mathbf{P}_{\text{transition}}(t)$ can be written as:

$$\mathbf{P}_{\text{transition}}(t) = \mathbf{P}^{(1)}_{\text{transition}}(t) \otimes \ldots \otimes \mathbf{P}^{(k)}_{\text{transition}}(t), \tag{9.20}$$

in which $\otimes$ denotes the Kronecker tensor product operator. In this way, Eq. (9.20) completely specifies the transition distribution matrix for *all* of the particles in the network.

Essentially, when $K \geq 2$, $p(t)$ is a vector and we would no longer be able to conventionally define the row $p(t)$ of matrix $\mathbf{P}_{\text{transition}}(t)$. Owing to this, we define an invertible mapping $f : \mathscr{V}^K \mapsto \mathbb{N}$. The function f simply maps the input vector to a scalar number that reflects the natural ordering of the tuples in the input vector. For example, $p(t) = [1, 1, \ldots, 1, 1]$ (all particles at vertex 1) denotes the first state; $p(t) = [1, 1, \ldots, 1, 2]$ (all particles at vertex 1, except the last particle, which is at vertex 2) is the second state; and so on, up to the scalar state V^K. Therefore, with this tool, we can fully manipulate the matrix $\mathbf{P}_{\text{transition}}(t)$.

Remark 9.1. The matrix $\mathbf{P}_{\text{transition}}(t)$ in (9.20) possesses dimensions $V^K \times V^K$, which are undesirably high. In order to save up space, one can use the individual transition matrices associated to each particle (therefore, we maintain a collection of K matrices), as shown in (9.1), each of which with dimensions $V \times V$, to model the dynamic of the particles' transition with no loss of generality, by using the following method: once every transition of the collection of K matrices has been performed, one could concatenate the new particle positions to assemble the random vector that denotes the particles' localization, $p(t+1)$, in an ordered manner. With this technique, the spatial complexity would not surpass $\mathscr{O}(KV)$, provided that we implement the matrices in a sparse mode.

9.2.3 Definition of the Stochastic Nonlinear Dynamical System

We can stack up all of the dynamic variables that have been introduced in the previous section to make up the dynamical system's state $\mathbf{X}(t)$ as follows:

$$\mathbf{X}(t) = \begin{bmatrix} p(t) \\ \mathbf{N}(t) \\ E(t) \\ S(t) \end{bmatrix}, \tag{9.21}$$

and the dynamical system that governs the particle competition model is given by:

$$\phi : \begin{cases} p^{(k)}(t+1) &= j, \quad j \sim \mathbf{P}^{(k)}_{\text{transition}}(t) \\ \mathbf{N}_i^{(k)}(t+1) &= \mathbf{N}_i^{(k)}(t) + \mathbb{1}_{[p^{(k)}(t+1)=i]} \\ E^{(k)}(t+1) &= \begin{cases} \min(\omega_{\max}, E^{(k)}(t) + \Delta), \text{if owner}(k,t) \\ \max(\omega_{\min}, E^{(k)}(t) - \Delta), \text{if } \neg\text{owner}(k,t) \end{cases} \\ S^{(k)}(t+1) &= \mathbb{1}_{[E^{(k)}(t+1)=\omega_{\min}]} \end{cases} \tag{9.22}$$

The first equation of system ϕ addresses the transition rules from i to a neighbor j, in which j is determined according to the time-varying transition matrix in (9.1). In other words, the acquisition of $p(t+1)$ is performed by generating random numbers following the distribution of the transition matrix $\mathbf{P}^{(k)}_{\text{transition}}(t)$. The second equation updates the number of visits that vertex i has received by particle k up to time t. The third equation is used to update the energy levels of all of the particles inserted in the network. Finally, the fourth equation indicates whether the particle is active or exhausted, depending on its actual energy level. Note that system ϕ is nonlinear. This occurs on account of the indicator function, which is nonlinear.

Observe that system ϕ can also be written in matrix form as:

$$\phi : \begin{cases} p(t+1) &= \mathbf{f}_p(p(t)), \quad \mathbf{f}_p(p(t)) \sim \mathbf{P}_{\text{transition}}(t) \\ \mathbf{N}(t+1) &= \mathbf{f}_{\mathbf{N}}(\mathbf{N}(t), p(t+1)) \\ E(t+1) &= f_E(\mathbf{N}(t+1), p(t+1)) \\ S(t+1) &= f_S(E(t+1)) \end{cases}, \tag{9.23}$$

in which $\mathbf{f}_p(.)$, $\mathbf{f}_{\mathbf{N}}(.)$, $f_E(.)$, and $f_S(.)$ are suitable random matrix functions, whose entries have been defined in (9.22). An important characteristic of system ϕ is its Markovian property (see Proposition 9.1).

Now we discuss how to settle the initial condition of the dynamical system's state $\mathbf{X}(0)$. Firstly, the particles are randomly inserted into the network, i.e., the values of $p(0)$ are randomly set. A desirable and interesting feature of the particle competition method is that the initial positions of the particles do not affect the community detection or data clustering results, due to the competition nature. This behavior occurs even when particles are put together at the beginning of the process.

Each entry of matrix $\mathbf{N}(0)$ is initialized according to the following expression:

$$\mathbf{N}_i^{(k)}(0) = \begin{cases} 2, & \text{if particle } k \text{ is generated at vertex } i. \\ 1, & \text{otherwise.} \end{cases} \tag{9.24}$$

Remark 9.2. The initialization of $\mathbf{N}(0)$ may be awkward, but there is a mathematical reason behind it. The domination level matrix, $\bar{\mathbf{N}}(0)$, is a row-normalization of $\mathbf{N}(0)$. Therefore, if all entries of a same row are zero, then (9.8) is undefined. In order to overcome this problem, all entries of matrix $\mathbf{N}(0)$ are evenly set to 1, with exception of those in which the particles are initially spawned, whose starting values are 2. In this setup, a consistent initial configuration for the competitive scheme is provided.

Since a fair competition among the particles is desired, all particles $k \in \mathscr{K}$ start out with the same energy level:

$$E^{(k)}(0) = \omega_{\min} + \left(\frac{\omega_{\max} - \omega_{\min}}{K}\right). \tag{9.25}$$

Lastly, all particles are active in the beginning of the competitive process, i.e.:

$$S^{(k)}(0) = 0. \tag{9.26}$$

9.2.4 Method for Estimating the Number of Communities

The particle competition algorithm described by the dynamical system ϕ produces a large quantity of useful information. Some of these dynamical variables can be used to solve other kinds of problems beyond community detection. In this section, we review the method for determining the most likely number of communities or clusters in a data set presented in [35]. In order to do so, an efficient evaluator index called average maximum domination level $\langle R(t)\rangle \in [0, 1]$ that monitors the information generated by the competitive model itself is constructed. Mathematically, this index is given by:

$$\langle R(t)\rangle = \frac{1}{V}\sum_{u\in\mathscr{V}} \max_{m\in\mathscr{K}} \left(\bar{\mathbf{N}}_u^{(m)}(t)\right), \tag{9.27}$$

in which $\bar{\mathbf{N}}_u^{(m)}(t)$ indicates the domination level that particle m is imposing on vertex u at time t (see (9.7)) and $\max\limits_{m\in\mathscr{K}} \left(\bar{\mathbf{N}}_u^{(m)}(t)\right)$ yields the maximum domination level imposed on vertex u at time t.

The basic idea can be described as follows. For a given network with K real communities, if we put exactly K particles in the network, each of them will dominate a community. Thus, one particle will not interfere much in the acting region of the other particles. As a consequence, $\langle R(t)\rangle$ will be large. In the extreme case when each vertex is completely dominated by a single particle, $\langle R(t)\rangle$ reaches 1. However, if we add more than K particles in the network, inevitably more than one particle will share the same community. Consequently, they will dispute the same group of vertices. In this case, one particle will lower the domination levels imposed by the other particles, and vice versa. As a result, $\langle R(t)\rangle$ will be small. Conversely, if we insert in the network a quantity of particles less than the number of real communities K, some particles will attempt to dominate more than one community. Again, $\langle R(t)\rangle$ will be small. In this way, the actual number of communities or clusters can be effectively estimated by checking for each K the index $\langle R(t)\rangle$ is maximized.

As it turns out, the optimal number of particles K to be inserted into a network is exactly the number of real communities that it has. In this way, the index $\langle R(t) \rangle$ is employed both to estimate the actual number of communities or clusters and also the number of particles K. The last point will be made precise in Sect. 9.2.6, where we study the sensitivity of the parameters that compose the particle competition model.

9.2.5 Method for Detecting Overlapping Structures

A measure that detects overlapping structures or vertices in a given network has been proposed in [36]. For this purpose, the domination level matrix $\bar{\mathbf{N}}(t)$ generated by the particle competition process is employed. The intuition is as follows. When the maximum domination level imposed by an arbitrary particle k on a specific vertex i is much larger 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 k and no other particle is influencing it in a relevant manner. Therefore, the overlapping nature of such vertex is minimal. In contrast, 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 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:

$$O_i(t) = 1 - (M_i(1, t) - M_i(2, t)) \, , \tag{9.28}$$

i.e., the overlapping index $O_i(t)$ measures the difference between the two greatest domination levels imposed by any pair of particles in the network on vertex i.

9.2.6 Parameter Sensitivity Analysis

The particle competition model requires a set of parameters to work. In special, we need to set the number of particles (K), the desired fraction of preferential movement (λ), the energy that each particle gains or loses (Δ), and a stopping factor (ϵ). In this section, we give the intuition on how to choose all of these parameters based on the type of data set we are dealing with.

In this section, we also discuss candidates as termination criteria.

9.2.6.1 Impact of the Parameter λ

Parameter λ is responsible for counterweighting the proportion of preferential and random walks performed by all particles in the network. Recall that the preferential

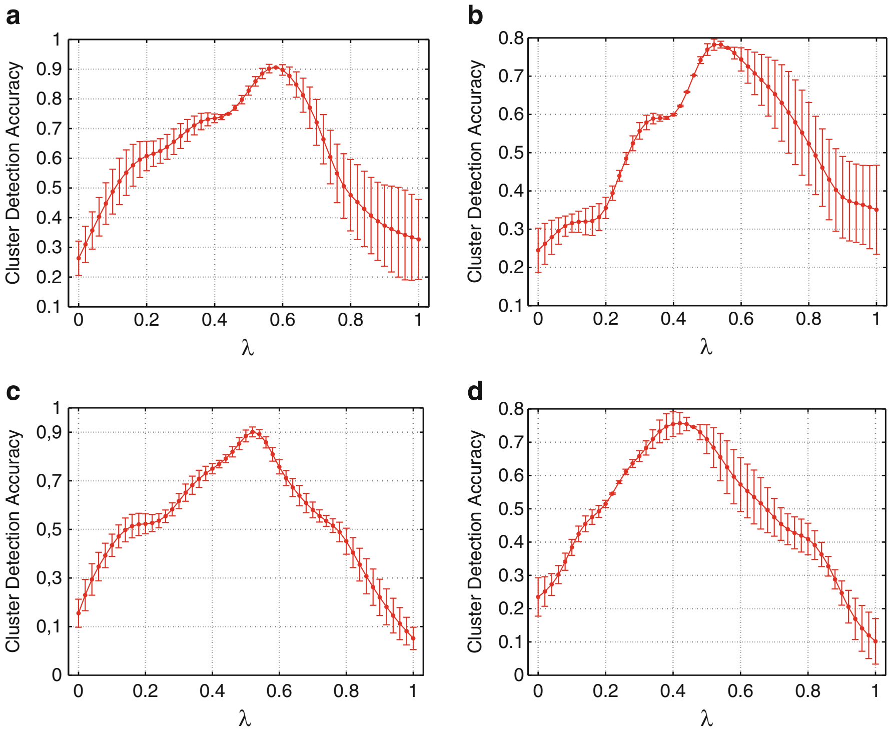

Fig. 9.5 Community detection rate vs. parameter λ. We fix $\Delta = 0.15$. Taking into account the steep peek that is verified and the large negative derivatives that surround it, one can see that the parameter λ is sensible to the overall model's performance. Results are averaged over 30 simulations. Reproduced from [36] with permission from Elsevier. (**a**) $\gamma = 2$ and $\beta = 1$, (**b**) $\gamma = 2$ and $\beta = 2$, (**c**) $\gamma = 3$ and $\beta = 1$, (**d**) $\gamma = 3$ and $\beta = 2$

term is related to the defensive behavior of the particles, while the random term is associated to the exploratory behavior. If we have small values for λ, we favor randomness over preferential visiting. As we increase λ, the tendency is to prefer reinforcing dominated territories instead of exploring new vertices. The two terms serve different and important roles in the community detection task and, in this section, we show that a combination of randomness and preferential behavior can really improve the performance of community detection tasks.

To study the role of λ in the learning process, we use artificial clustered networks that are generated following the benchmark of Lancichinatti et al. [21], which we have introduced in Sect. 6.2.4. We fix $V = 10{,}000$ vertices and the average network degree or network connectivity as $\bar{k} = 15$. Recall that the benchmark consists in varying the mixing parameter μ while evaluating the attained community detection rates.

Figure 9.5a–d portray the community detection rate of the particle competition model as a function of λ. We vary the counterweighting parameter λ from 0 (pure random walks) to 1 (pure preferential walks) for different values of γ and β, which are the exponents of the power-law degree and community size distributions.

We can observe that the community detection rate of the particle competition algorithm is very sensitive to parameter λ. Though choosing several different values for γ and β, we can see a very clear behavior from these pictures: when $\lambda = 0$ or $\lambda = 1$, the particle competition algorithm does not produce satisfactory results. These values correspond to walks with only random or preferential terms, respectively. This observation suggests that a mixture of these two terms can improve the algorithm's performance to a significant extent. One reason for that is because each of these terms serve a different role in the community detection process: while the random term expands community borders, the preferential term guarantees that community cores stay strongly dominated. The tradeoff between the speed of expanding community borders and guaranteeing the control of the subset of dominated vertices is performed by tuning parameter λ. By our results, we see that the particle competition algorithm provides good results when we have a sustainable increase and defense of the community borders, which happens when we select intermediate values for λ.

As a rule-of-thumb, the model gives good community detection rates in networks with well-defined communities when $0.2 \leq \lambda \leq 0.8$.

9.2.6.2 Impact of the Parameter Δ

Parameter Δ is responsible for updating the particles' energy levels as described in (9.13). We use the same type of artificial clustered networks as in the previous analysis. Figure 9.6a–d display the community detection rate of the particle competition model as a function of Δ. Again, we see that the competitive model does not behave well for extremal values of parameter Δ. The intuition for that is as follows. When Δ is very small, particles are not penalized enough and hence they do not get exhausted often. Consequently, particles are expected to frequently visit vertices that should belong to rival particles, possibly getting into the core of other communities. Therefore, all of the vertices in the network will be in constant competition and no community borders will be established and consolidated. As such, the algorithm's performance is expected to be poor. On the other extreme, when Δ is very large, particles are expected to be constantly exhausted once they visit vertices dominated by rival particles, thus frequently returning to their community core. In this setup, the initial positions of the particles become sensitive to the competitive model. Once we randomly put the particles inside the network at $t = 0$, they are expected to not venture far away from their initial positions due to the reanimation procedure. As such, whenever we put particles near each other, the community detection rate will be poor. In this way, it is unattainable for the particles to switch the ownership of already conquered vertices. We can conceive this phenomenon as an artificial "hard labeling."

Another interesting characteristic that can be extracted from the sensitivity curves in Fig. 9.6a–d is that the competitive model becomes robust against variations of Δ

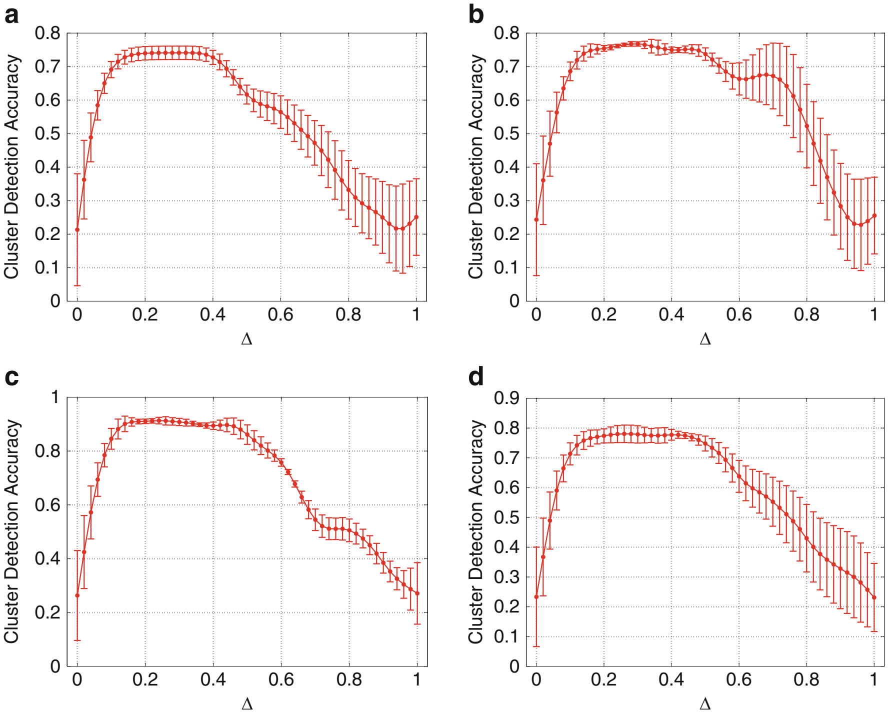

Fig. 9.6 Community detection rate vs. parameter Δ. We fix $\lambda = 0.6$. Taking into account the large steady region around $0.1 \leq \Delta \leq 0.4$, one can see that the parameter Δ is conditionally not sensible to the overall model's performance. Results are averaged over 30 simulations. Reproduced from [36] with permission from Elsevier. (**a**) $\gamma = 2$ and $\beta = 1$, (**b**) $\gamma = 2$ and $\beta = 2$, (**c**) $\gamma = 3$ and $\beta = 1$, (**d**) $\gamma = 3$ and $\beta = 2$

when we are in the region $0.1 \leq \Delta \leq 0.4$, i.e., it is conditionally not sensitive to Δ in this relatively wide interval. In this way, as a rule-of-thumb, we should choose intermediate but small values of Δ.

9.2.6.3 Impact of the Parameter *K*

Parameter K quantifies the number of particles that are inserted into the network to perform the community detection process. In comparison to all of the other parameters of the particle competition algorithm, K is the most sensitive parameter for the model's performance. Hence, the correct determination or at least estimation of K stands as an important problem when employing the particle competition model. Considering that, in the long-run dynamic, each particle dominates a single community, the heuristic presented for estimating the actual number of clusters or communities in Sect. 9.2.4 is a perfect candidate for estimating the proper value for the K parameter. That is, we estimate the number of particles K as the estimated number of communities in the data set using the index $\langle R(t)\rangle$. Mathematically, we choose a candidate K, $K_{\text{candidate}}$, such as to maximize the measure $\langle R(t)\rangle$, $\langle R(t)_{\max}\rangle$, as follows:

$$K = \{K_{\text{candidate}} \in \mathbb{N} : \langle R(t)\rangle = \langle R_{\max}(t)\rangle\}, \tag{9.29}$$

in which $\langle R(t)_{\max}\rangle$ is given by:

$$\begin{aligned}\langle R(t)_{\max}\rangle &= \arg \max_{|\mathscr{K}|\in\mathbb{N}} \frac{1}{V} \sum_{u\in\mathscr{V}} \max_{m\in\mathscr{K}} \left(\bar{\mathbf{N}}_u^{(m)}(t)\right) \\ &\propto \arg \max_{|\mathscr{K}|\in\mathbb{N}} \sum_{u\in\mathscr{V}} \max_{m\in\mathscr{K}} \left(\bar{\mathbf{N}}_u^{(m)}(t)\right).\end{aligned} \tag{9.30}$$

In computational terms, we iterate the particle competition algorithm using $K = 2$ up to a small positive number, while maintaining the best K associated to the maximum achieved $\langle R(t)_{\max}\rangle$. We do not need to try large values for K because the number of communities is often far less than the number of data items.

9.2.7 Convergence Analysis

In this section, we present two possible stopping criteria for the particle competition model. Both of them assume that the particle competition converges. The termination criteria stands as an important issue as we are dealing with a dynamical system that can evolve indefinitely. In essence, we investigate the properties of the indices $\langle R(t)\rangle$ and $|\bar{\mathbf{N}}(t+1) - \bar{\mathbf{N}}(t)|_\infty$ when employed as stopping criteria. We inspect their behavior as a function of time and conclude for the convergence of the dynamical system using an empirical analysis. Based on convergence issues, we give evidences favoring $|\bar{\mathbf{N}}(t+1) - \bar{\mathbf{N}}(t)|_\infty$ in detriment to $\langle R(t)\rangle$.

In our analysis in this section, we use the synthetic data sets shown in Fig. 9.7a–c, which is composed of two communities: the red or “circle” and the green or “square” communities. The two groups in Fig. 9.7a are well-posed as their distributions are distinct and do not overlap. Figure 9.7b portrays an intermediate situation, in which the two groups slightly overlap. Finally, Fig. 9.7c depicts an ill-posed situation, in which the groups largely overlap. In the latter, the clustering task is extremely difficult since the smoothness and cluster assumptions do not hold.

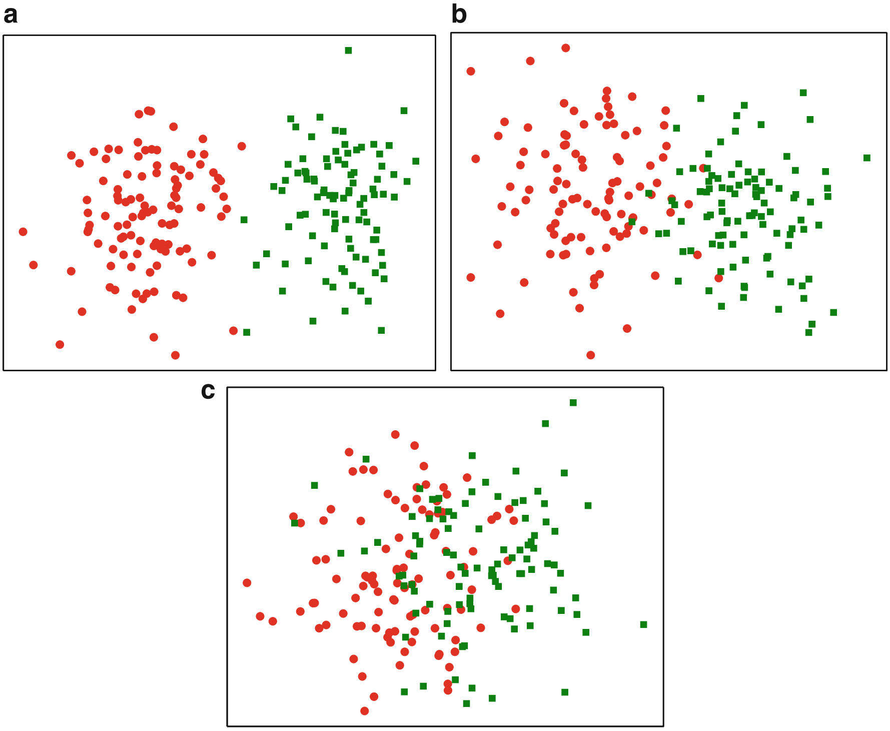

Fig. 9.7 Scatter plot of artificial databases constituted by two groups. The data is constructed using two bi-dimensional Gaussian distributions with varying mean and unitary covariance. Reproduced from [36] with permission from Elsevier. (**a**) Well-posed groups, (**b**) somewhat well-posed groups, (**c**) not well-posed groups

Now, we investigate how the index $\langle R(t) \rangle$, which has been introduced in Sect. 9.2.4, behaves as the competitive dynamical system progresses in time. The simulation results with regard to the synthetic data sets displayed in Fig. 9.7a–c are depicted in Fig. 9.8a–c, respectively. In all of these plots, we have explicitly indicated two important dynamical properties: (1) t_s, which is the time to reach the "almost-stationary" state of the model and (2) the diameter of the region in which the almost-stationary state is confined within. Note that, since the competition is always taking place, the model never reaches a perfect stationary state. Rather, the dynamic variables float around quasi-stationary states because of the constant visits that particles perform on vertices of the network. These fluctuations are expected, since the random walk behavior of the particles, which is denoted by the second term in Eq. (9.1), compels particles to visit vertices that they do not dominate. This behavior creates oscillations in the domination levels between rival particles. However, if we conduct walks with no random behavior, i.e., with only preferential movements, these fluctuations are expected to be eliminated, since the exploratory behavior of the particles would cease to exist. In this case, only the defensive behavior would

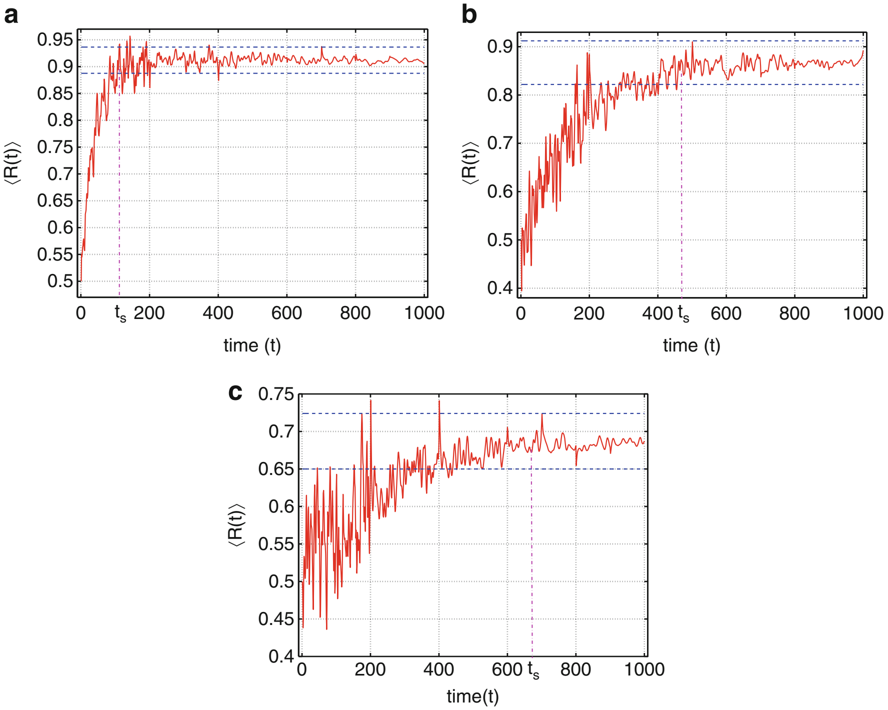

Fig. 9.8 Convergence analysis of the particle competition algorithm when $\langle R(t)\rangle$ is used. The algorithm is run against the binary artificial databases in Fig. 9.7. Here, we inspect how the index $\langle R(t)\rangle$ varies as a function of time. Reproduced from [36] with permission from Elsevier. (**a**) $\langle R(t)\rangle$ for Fig. 9.7a, (**b**) $\langle R(t)\rangle$ for Fig. 9.7b, (**c**) $\langle R(t)\rangle$ for Fig. 9.7c

be used by particles. However, each of the two kinds of movements (random and preferential) has its role in the competition process, in a such a way that disabling one or another would drastically affect the community or cluster detection. As such, good values for λ must reside between 0 and 1 and not in the extremes.

From Fig. 9.8a–c, we see that the time to reach the almost-stationary state t_s lingers to be established as the overlapping region of the groups gets larger. In this respect, t_s is roughly $150, 430, 650$ for Fig. 9.8a–c, respectively. This is because competition in the community border regions gets stronger as the overlap width increases. As a consequence, the dominance of each particle takes longer to be established. Another interesting phenomenon is that of the diameter of the confinement region of $\langle R(t)\rangle$, which grows larger as the overlap width increases. In these simulations, the diameters of such regions are roughly $0.06, 0.07, 0.08$. This is expected by the same reasons stated before.

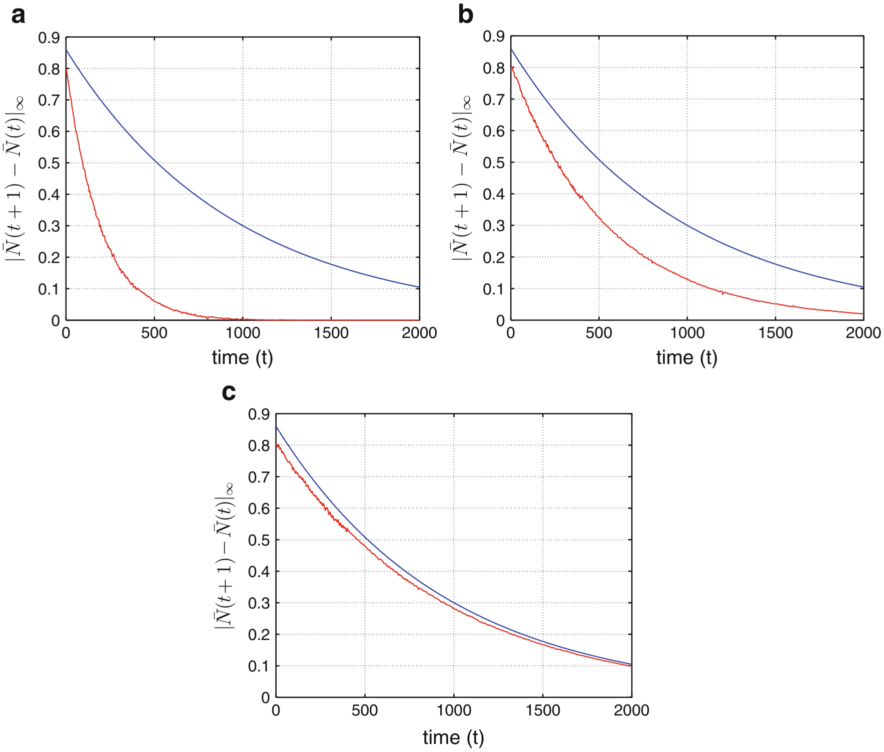

Fig. 9.9 Convergence analysis of the particle competition algorithm when $|\bar{\mathbf{N}}(t+1) - \bar{\mathbf{N}}(t)|_\infty$ is used. The upper theoretical limit is shown in the *blue curve* when $c = K$. The algorithm is run against the binary artificial databases in Fig. 9.7. Here, we inspect how $|\bar{\mathbf{N}}(t+1) - \bar{\mathbf{N}}(t)|_\infty$ varies as a function of time. Reproduced from [36] with permission from Elsevier. (**a**) $|\bar{\mathbf{N}}(t+1) - \bar{\mathbf{N}}(t)|_\infty$ for Fig. 9.7b, (**b**) $|\bar{\mathbf{N}}(t+1) - \bar{\mathbf{N}}(t)|_\infty$ for Fig. 9.7b, (**c**) $|\bar{\mathbf{N}}(t+1) - \bar{\mathbf{N}}(t)|_\infty$ for Fig. 9.7b

Figure 9.9a–c shows the variation term $|\bar{\mathbf{N}}(t+1) - \bar{\mathbf{N}}(t)|_\infty$ as a function of time when applied to the data sets in Fig. 9.7a–c. We see that the variation of $\bar{\mathbf{N}}(t+1)$ in relation to $\bar{\mathbf{N}}(t)$ reduces as time evolves. This happens because the total number of visits performed by particles always increases, since each particle must visit at least a vertex in any given time. Looking at Eq. (9.7), we see that the denominator always increases faster than the numerator. Therefore, it provides an upper limit for $\bar{\mathbf{N}}(t)$. In view of this, the variations from one iteration to another, i.e., $|\bar{\mathbf{N}}(t+1) - \bar{\mathbf{N}}(t)|_\infty$, tend to vary less and less. Analytically, we can verify that, when the particles start to walk, i.e., when $t = 1$, the maximum variation of $|\bar{\mathbf{N}}(t+1) - \bar{\mathbf{N}}(t)|_\infty$ is given by:

$$| \bar{\mathbf{N}}(1) - \bar{\mathbf{N}}(0) |_\infty \leq c \left(\frac{2}{V} - \frac{1}{V+1} \right), \tag{9.31}$$

in which c is a real positive constant which depends on the level of competitiveness, which in turn is directly proportional to λ. Such upper limit expression translates the maximum variation that occurs from time $t = 0$ to $t = 1$, which happens when, at $t = 0$, a vertex is not receiving any visits, but, at time $t = 1$, it is being visited by exactly one particle. Generalizing this equation for an arbitrary t, $|\bar{\mathbf{N}}(t+1) - \bar{\mathbf{N}}(t)|_\infty$ is always bounded by the following expression:

$$| \bar{\mathbf{N}}(t+1) - \bar{\mathbf{N}}(t) |_\infty \leq c \left(\frac{t+2}{V+t} - \frac{t}{V+t+1} \right), \tag{9.32}$$

which demonstrates that, as the previous case, for any $t \leq \infty$, the model presents fluctuations around a quasi-stationary state. From this analysis, it is clear that the $|\bar{\mathbf{N}}(t+1) - \bar{\mathbf{N}}(t)|_\infty$ can be used as a termination criterion.

In summary, we find that the particle competition algorithm does not converge to a fixed point, but the dynamics of the system gets confined within a small finite sub-region in the space. The intuition behind that is, though in the long-run the communities have already being established, the competition among particles is always occurring. In this way, the domination levels of vertices keep changing, though with less magnitude as time progresses due to the accumulative effect that the number of visits plays in establishing the vertices' domination levels.

9.3 Theoretical Analysis of the Model

In this section, a mathematical analysis of the competitive system is supplied. Also, we show that the competitive system reviewed in the previous section reduces to multiple independent random walks when a special situation occurs. Some of the results have been presented in [37] and others are new results. In this book, we present the mathematical analysis in a self-contained manner.

9.3.1 Mathematical Analysis

To estimate the long-run dynamic of the stochastic competitive learning model, we first need to derive the transition probabilities between the different states in the dynamical system. Let the transition probability function of system ϕ be $P(\mathbf{X}(t+1) \mid \mathbf{X}(t))$. Observe that the marginal probability of the system's state $P(\mathbf{X}(t))$ can be written in terms of the joint probability of each of the components of the system's state, meaning $P(\mathbf{X}(t)) = P(\mathbf{N}(t), p(t), E(t), S(t))$. Thus, applying the product rule on the transition probability function, we have:

$$\begin{aligned}
&P(\mathbf{X}(t+1) \mid \mathbf{X}(t)) \\
&\quad = P(\mathbf{N}(t+1), p(t+1), E(t+1), S(t+1) \mid \mathbf{N}(t), p(t), E(t), S(t)) \\
&\quad = P(S(t+1) \mid \mathbf{N}(t+1), p(t+1), E(t+1), \mathbf{N}(t), p(t), E(t), S(t)) \\
&\qquad \times P(\mathbf{N}(t+1), p(t+1), E(t+1) \mid \mathbf{N}(t), p(t), E(t), S(t)) \\
&\quad = P_{S(t+1)} P(E(t+1) \mid \mathbf{N}(t+1), p(t+1), \mathbf{N}(t), p(t), E(t), S(t)) \\
&\qquad \times P(\mathbf{N}(t+1), p(t+1) \mid \mathbf{N}(t), p(t), E(t), S(t)) \\
&\quad = P_{S(t+1)} P_{E(t+1)} P(\mathbf{N}(t+1) \mid p(t+1), \mathbf{N}(t), p(t), E(t), S(t)) \\
&\qquad \times P(p(t+1) \mid \mathbf{N}(t), p(t), E(t), S(t)) \\
&\quad = P_{S(t+1)} P_{E(t+1)} P_{\mathbf{N}(t+1)} P_{p(t+1)}.
\end{aligned} \tag{9.33}$$

in which:

$$P_{S(t+1)} = P(S(t+1) \mid \mathbf{N}(t+1), p(t+1), E(t+1), \mathbf{X}(t)), \tag{9.34}$$

$$P_{E(t+1)} = P(E(t+1) \mid \mathbf{N}(t+1), p(t+1), \mathbf{X}(t)), \tag{9.35}$$

$$P_{\mathbf{N}(t+1)} = P(\mathbf{N}(t+1) \mid p(t+1), \mathbf{X}(t)), \tag{9.36}$$

$$P_{p(t+1)} = P(p(t+1) \mid \mathbf{X}(t)). \tag{9.37}$$

Next, the algebraic derivations of these four quantities are explored.

9.3.1.1 Discovering the Factor $P_{p(t+1)}$

Observing that the random vector $p(t+1)$ is directly evaluated from $\mathbf{P}_{\text{transition}}(t)$ given in (9.20), which in turn only requires $p(t)$ and $\mathbf{N}(t)$ to be constructed ($\mathbf{X}(t)$ is given), then the following equivalence holds:

$$P_{p(t+1)} = P(p(t+1) \mid \mathbf{X}(t)) = \mathbf{P}_{\text{transition}}(\mathbf{N}(t), p(t)). \tag{9.38}$$

Here, we have used $\mathbf{P}_{\text{transition}}(\mathbf{N}(t), p(t))$ to emphasize the dependence of the transition matrix on $\mathbf{N}(t)$ and $p(t)$.

9.3.1.2 Discovering the Factor $P_{\mathbf{N}(t+1)}$

In this case, taking a close look at $P_{\mathbf{N}(t+1)} = P(\mathbf{N}(t+1) \mid p(t+1), \mathbf{X}(t))$, we can verify that, besides the previous state $\mathbf{X}(t)$, we also know the value of the random vector $p(t+1)$. By a quick analysis of the update rule given in the second expression of system ϕ, it is possible to completely determine $\mathbf{N}(t+1)$, since $p(t+1)$ and $\mathbf{N}(t)$ are known. Owing to that, the following equation holds:

$$\begin{aligned} P_{\mathbf{N}(t+1)} &= P(\mathbf{N}(t+1) \mid p(t+1), \mathbf{X}(t)) \\ &= \mathbb{1}_{[\mathbf{N}(t+1)=\mathbf{N}(t)+\mathbf{Q}_N(p(t+1))]}, \end{aligned} \tag{9.39}$$

in which $\mathbf{Q}_N(p(t+1))$ is a matrix with $\dim(\mathbf{Q}_N) = V \times K$ and dependent on $p(t+1)$. The (i,j)-th entry of $\mathbf{Q}_N(p(t+1))$ is given by:

$$\mathbf{Q}_N(p(t+1))(i,k) = \mathbb{1}_{\left[p^{(k)}(t+1)=i\right]}, \tag{9.40}$$

The argument in the indicator function shown in (9.39) is essentially the first expression of system ϕ, but in a matrix notation. In brief, Eq. (9.39) results in 1 if the computation of $\mathbf{N}(t+1)$ is correct, given $p(t+1)$ and $\mathbf{N}(t)$, i.e., it is in compliance with the dynamical system rules; and 0, otherwise.

9.3.1.3 Discovering the Factor $P_{E(t+1)}$

For the third term, $P_{E(t+1)}$, we have knowledge of the previous state $\mathbf{X}(t)$, as well as of $p(t+1)$ and $\mathbf{N}(t+1)$. By (9.7), we see that $\bar{\mathbf{N}}(t+1)$ can be directly calculated from $\mathbf{N}(t+1)$, i.e., having knowledge of $\mathbf{N}(t+1)$ permits us to evaluate $\bar{\mathbf{N}}(t+1)$, which, probabilistically speaking, is also a given information. In light of this, together with (9.13), one can see that $E(t+1)$ can be evaluated if we have information of $E(t)$, $p(t+1)$, and $\bar{\mathbf{N}}(t+1)$, which we actually do. On account of that, $P_{E(t+1)}$ can be surely determined and, analogously to the calculation of the $P_{\mathbf{N}(t+1)}$, is given by:

$$\begin{aligned} P_{E(t+1)} &= P(E(t+1) \mid \mathbf{N}(t+1), p(t+1), \mathbf{X}(t)) \\ &= \mathbb{1}_{[E(t+1)=E(t)+\Delta \times Q_E(p(t+1),\mathbf{N}(t+1))]}, \end{aligned} \tag{9.41}$$

in which $Q_E(p(t+1), \mathbf{N}(t+1))$ is a random vector with $\dim(Q_E) = 1 \times K$ and dependence on $\mathbf{N}(t+1)$ and $p(t+1)$. The k-th entry, $k \in \mathscr{K}$, of such matrix is calculated as:

$$Q_E^{(k)}(p(t+1), \mathbf{N}(t+1)) = \mathbb{1}_{[\text{owner}(k,t+1)]} - \mathbb{1}_{[\neg\text{owner}(k,t+1)]}. \tag{9.42}$$

Note that the argument of the indicator function in (9.42) is essentially (9.13) in a compact matrix form. Indicator functions were employed to describe the two types of behavior that this variable can have: an increment or decrement of the particle's energy. Suppose that particle $k \in \mathscr{K}$ is visiting a vertex that it dominates, then only the first indicator function in (9.42) is enabled; hence, $Q_E^{(k)}(p(t+1), \mathbf{N}(t+1)) = 1$. Similarly, if particle k is visiting a vertex that is being dominated by a rival particle, then the second indicator function is enabled, yielding $Q_E^{(k)}(p(t+1), \mathbf{N}(t+1)) = -1$. This behavior together with (9.41) is exactly the expression given by (9.13) in a compact matrix form.

9.3.1.4 Discovering the Factor $P_{S(t+1)}$

Lastly, for the fourth term, $P_{S(t+1)}$, we have knowledge of $E(t+1)$, $\mathbf{N}(t+1)$, $p(t+1)$, and the previous internal state $\mathbf{X}(t)$. By a quick analysis of (9.15), one can verify that the calculation of the k-th entry of $S(t+1)$ is completely characterized once $E(t+1)$ is known. In this way, one can surely evaluate $P_{S(t+1)}$ in this scenario as follows:

$$\begin{aligned} P_{S(t+1)} &= P(S(t+1) \mid E(t+1), \mathbf{N}(t+1), p(t+1), \mathbf{X}(t)) \\ &= \mathbb{1}_{[S(t+1)=Q_S(E(t+1))]}, \end{aligned} \tag{9.43}$$

in which $Q_S(E(t+1))$ is a matrix with $\dim(Q_S) = 1 \times K$ and has dependence on $E(t+1)$. The k-th entry, $k \in \mathscr{K}$, of such matrix is calculated as:

$$Q_S^{(k)}(E(t+1)) = \mathbb{1}_{\left[E^{(k)}(t+1)=\omega_{\min}\right]}. \tag{9.44}$$

9.3.1.5 The Transition Probability Function

Substituting (9.38), (9.39), (9.41), and (9.43) into (9.33), we are able to encounter the transition probability function of the competitive dynamical system:

$$\begin{aligned} P(\mathbf{X}(t+1) \mid \mathbf{X}(t)) = {} & \mathbb{1}_{[\mathbf{N}(t+1)=\mathbf{N}(t)+\mathbf{Q}_N(p(t+1))]} \\ & \times \mathbb{1}_{[S(t+1)=Q_S(E(t+1))]} \\ & \times \mathbb{1}_{[E(t+1)=E(t)+\Delta Q_E(p(t+1),\mathbf{N}(t+1))]} \\ & \times \mathbf{P}_{\text{transition}}(\mathbf{N}(t), p(t)) \\ = {} & \mathbb{1}_{[\text{Compliance}(t)]} \mathbf{P}_{\text{transition}}(\mathbf{N}(t), p(t)), \end{aligned} \tag{9.45}$$

in which Compliance(t) is a logical expression given by:

$$\begin{aligned} \text{Compliance}(t) = {} & [\mathbf{N}(t+1) = \mathbf{N}(t) + \mathbf{Q}_N(p(t+1))] \\ & \wedge\ [S(t+1) = Q_S(E(t+1))] \wedge [E(t+1) \\ & = E(t) + \Delta Q_E(p(t+1), \mathbf{N}(t+1))], \end{aligned} \tag{9.46}$$

i.e., Compliance(t) encompasses all the rules that have to be satisfied in order to all the indicator functions in (9.45) produce 1. If all the values provided to (9.45) are in compliance with the dynamic of the system, then Compliance(t) = true and the indicator function $\mathbb{1}_{[\text{Compliance}(t)]}$ yields 1; otherwise, if there is at least one measure that does not satisfy the system, then, from (9.46), the chain of logical-AND produces false. As a consequence, Compliance(t) = false and the indicator function $\mathbb{1}_{[\text{Compliance}(t)]}$ in (9.45) yields 0, resulting in a zero-valued transition probability.

9.3.1.6 Discovering the Distribution $P(\mathbf{N}(t))$

With the transition probability function derived in the previous section, we now turn our attention to determining the marginal distribution $P(\mathbf{N}(t))$ for a sufficiently large t. First, the Markovian property of system ϕ is demonstrated as follows.

Proposition 9.1. $\{\mathbf{X}(t) : t \geq 0\}$ *is a Markovian process.*

Proof. We seek to infer that system ϕ is completely characterized by only the acquaintance of the present state, i.e., it is independent of all the past states. Having that in mind, the probability expression to make a transition to a specific event X_{t+1} (a set with an element representing an arbitrary next state) in time $t+1$, given the complete history of the state trajectory, is denoted by:

$$P\left(\mathbf{X}(t+1) \in X_{t+1} \mid \mathbf{X}(t), \ldots, \mathbf{X}(0)\right)$$
$$= P\left(p_{t+1} : \begin{bmatrix} f_N(\mathbf{N}(t), p_{t+1}) \\ f_E(\mathbf{N}(t+1), p_{t+1}) \\ f_S(E(t+1)) \end{bmatrix} \in X_{t+1} \mid \mathbf{X}(t), \ldots, \mathbf{X}(0)\right). \tag{9.47}$$

Noting that the determination of p_{t+1} only depends on $\mathbf{N}(t)$ and $p(t)$, then:

$$P\left(p_{t+1} : \begin{bmatrix} f_N(\mathbf{N}(t), p_{t+1}) \\ f_E(\mathbf{N}(t+1), p_{t+1}) \\ f_S(E(t+1)) \end{bmatrix} \in X_{t+1} \mid \mathbf{X}(t), \ldots, \mathbf{X}(0)\right)$$
$$= P\left(p_{t+1} : \begin{bmatrix} f_N(\mathbf{N}(t), p_{t+1}) \\ f_E(\mathbf{N}(t+1), p_{t+1}) \\ f_S(E(t+1)) \end{bmatrix} \in X_{t+1} \mid \mathbf{X}(t)\right)$$
$$= P\left(\mathbf{X}(t+1) \in X_{t+1} \mid \mathbf{X}(t)\right). \tag{9.48}$$

Therefore, in view of (9.48), $\{\mathbf{X}(t) : t \geq 0\}$ is a Markovian process, since it only depends on the present state to specify the next state and, hence, the past history of the system's trajectory is irrelevant. ∎

The strategy to calculate the distribution $P(\mathbf{N}(t))$ is to marginalize the joint distribution of the system's states, i.e., $P(\mathbf{X}(0), \ldots, \mathbf{X}(t))$, with respect to $\mathbf{N}(t)$ (a component of $\mathbf{X}(t)$). Mathematically, using Proposition 1 on this joint distribution $P(\mathbf{X}(0), \ldots, \mathbf{X}(t))$, we get:

$$\begin{aligned} P(\mathbf{X}(0), \ldots, \mathbf{X}(t)) = & P(\mathbf{X}(t) \mid \mathbf{X}(t-1)) \\ & \times P(\mathbf{X}(t-1) \mid \mathbf{X}(t-2)) \\ & \times \ldots \times P(\mathbf{X}(1) \mid \mathbf{X}(0)) P(\mathbf{X}(0)). \end{aligned} \tag{9.49}$$

Using the transition function that governs system ϕ, as illustrated in (9.45), to each shifted term in (9.49), we get:

$$P(\mathbf{X}(0),\ldots,\mathbf{X}(t)) = P(\mathbf{X}(0)) \prod_{u=1}^{t-1} \left[\mathbb{1}_{[\text{Compliance}(u)]} \mathbf{P}_{\text{transition}}(\mathbf{N}(u), p(u)) \right], \tag{9.50}$$

in which $P(\mathbf{X}(0)) = P(\mathbf{N}(0), p(0), E(0), S(0))$. But, we are interested in knowing the marginal distribution $\mathbf{N}(t)$ as $t \to \infty$. We can obtain it from the joint distribution calculated in (9.50), summing over all the possible values of random variables with no relevance in the analysis, i.e., $\mathbf{N}(t-1),\ldots,\mathbf{N}(0), p(t),\ldots,p(0), E(t),\ldots,E(0), S(t),\ldots,S(0)$. In doing so, it is worth studying the limits of $\mathbf{N}(t)$ for an arbitrary t, because the domain that an entry of $\mathbf{N}(t)$ can take is $[1,\infty)$. With this study, we expect to find the reachable values of every entry of matrix $\mathbf{N}(t)$ for any t. In this way, values which exceed these limits are guaranteed to happen with probability 0. Lemma 9.1 precisely supplies this analysis.

Lemma 9.1. *The maximum reachable value of* $\mathbf{N}_i^{(k)}(t)$, $\forall (i,k) \in \mathscr{V} \times \mathscr{K}, t \in \mathbb{N}$, *is:*

$$\mathbf{N}_{i_{\max}}^{(k)}(t) = \begin{cases} \left\lceil \frac{t+1}{2} \right\rceil + 1, & \mathit{if}\ t \geq 0\ \mathit{and}\ a_{ii} = 0 \\ t+2, & \mathit{if}\ t \geq 0\ \mathit{and}\ a_{ii} > 0 \end{cases}. \tag{9.51}$$

Proof. The proof is based on encountering the particle's trajectory that increases $\mathbf{N}_i^{(k)}(t)$ in the quickest manner. In this situation, we suppose particle k is generated in vertex i; otherwise, the maximum theoretical value would never be reached in view of the second expression in (9.24). For the sake of clarity, consider two specific cases, both depicted in Fig. 9.10: (1) networks without self-loops and (2) networks with self-loops.

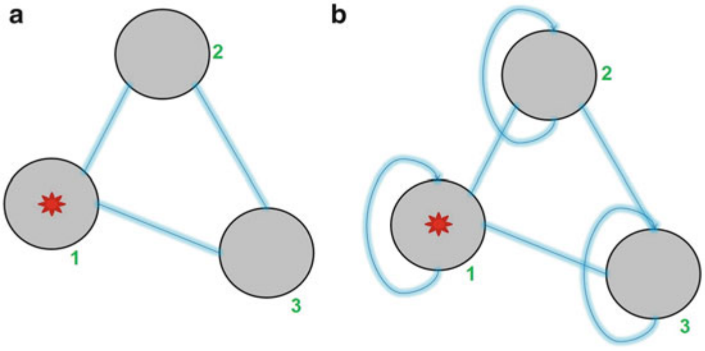

Fig. 9.10 An arbitrary network constructed with the purpose of obtaining the largest feasible entry of $\mathbf{N}(t)$ for a given t. (**a**) A network without the presence of self-loops; (**b**) a network with self-loops

For the first case, $\forall i \in \mathscr{V} : a_{ii} = 0$. By hypothesis, particle k starts in vertex i at $t = 0$. The quickest manner to increase $\mathbf{N}_i^{(k)}(t)$ happens when particle k visits a neighbor of vertex i and immediately returns to vertex i. Repeating this trajectory until time t, $\mathbf{N}_i^{(k)}(t)$ precisely matches the first expression in (9.51).

For the second case, $\exists i \in \mathscr{V} : a_{ii} > 0$. By hypothesis, particle k starts in vertex i at $t = 0$. It is clear that the quickest manner to increase $\mathbf{N}_i^{(k)}(t)$ occurs when particle k always travels through the self-loop for all t. In view of this, the maximum value that $\mathbf{N}_i^{(k)}(t)$ can reach is given by the second expression in (9.51). The "+2" factor appears because the particle initially spawns at vertex i, according to the second expression in (9.24). ■

In what follows, we analyze the properties of the random vector $E(t)$. The upper limit of the k-th entry of $E(t)$ is always $\omega_{\max}$. Thus, provided that $\omega_{\max} < \infty$, the upper limit is always well-defined. However, this entry does not only accept integer values in-between $\omega_{\min}$ and $\omega_{\max}$. Lemma 9.2 provides all reachable values of $E(t)$ within this interval.

Lemma 9.2. *The reachable domain of* $E^{(k)}(t), \forall k \in \mathscr{K}, t \in \mathbb{N}$, *is:*

$$\begin{aligned}\mathscr{D}_E \triangleq &\left\{\omega_{\min} + \frac{\omega_{\max} - \omega_{\min}}{K} + n\Delta, n = \{-\lfloor n_i \rfloor, \ldots, \lfloor n_m \rfloor\}\right\} \\ &\cup \left\{\omega_{\min} + n\Delta, n = \left\{1, 2, \ldots, \left\lfloor \frac{\omega_{\max} - \omega_{\min}}{\Delta} \right\rfloor\right\}\right\} \\ &\cup \left\{\omega_{\max} - n\Delta, n = \left\{1, 2, \ldots, \left\lfloor \frac{\omega_{\max} - \omega_{\min}}{\Delta} \right\rfloor\right\}\right\},\end{aligned} \tag{9.52}$$

in which $n_i = \frac{\omega_{\max} - \omega_{\min}}{K\Delta} \geq 0$ *and* $n_m = \frac{\omega_{\max} - \omega_{\min}}{\Delta}\left(1 - \frac{1}{K}\right) \geq 0$.

Proof. We divide this proof in three main steps, namely the three sets that appear in the expression in the *caput* of this lemma. The first set accounts for supplying all values that are multiples of Δ with the offset $E^{(k)}(0) = \omega_{\min} + \left(\frac{\omega_{\max} - \omega_{\min}}{K}\right), \forall k \in \mathscr{K}$ (see (9.25)). The minimum reachable value is given when $n = n_i$:

$$n_i = \frac{\left(\omega_{\min} + \frac{\omega_{\max} - \omega_{\min}}{K}\right) - \omega_{\min}}{\Delta} = \frac{\omega_{\max} - \omega_{\min}}{K\Delta}, \tag{9.53}$$

whereas the maximum reachable value is given when $n = n_m$:

$$n_m = \frac{\omega_{\max} - \left(\omega_{\min} + \frac{\omega_{\max} - \omega_{\min}}{K}\right)}{\Delta} = \frac{\omega_{\max} - \omega_{\min}}{\Delta}\left(1 - \frac{1}{K}\right). \tag{9.54}$$

After some time, the particle k might reach one of the two possible extremes of energy value: $\omega_{\min}$ or $\omega_{\max}$. On account of the max(.) operator in (9.13), it is also necessary to list all multiple numbers of Δ with these two offsets. The second and third sets precisely fulfill this aspect when the offsets are $\omega_{\min}$ and $\omega_{\max}$, respectively. Once the particle enters one of these sets, it never leaves them. Hence, all values have been properly mapped. ■

Lastly, the upper limit of an arbitrary entry of $S(t)$ is 1, since it is a boolean-valued variable. Observing now that $P(\mathbf{X}(0), \ldots, \mathbf{X}(t)) = P(\mathbf{N}(0), p(0), E(0), S(0), \ldots, \mathbf{N}(t), p(t), E(t), S(t))$, we marginalize this joint distribution with respect to $\mathbf{N}(t)$ as follows:

$$P(\mathbf{N}(t)) = \sum_{\sim \mathbf{N}(t)} P(\mathbf{X}(0), \ldots, \mathbf{X}(t)), \tag{9.55}$$

in which $\sim \mathbf{N}(t)$ means that we sum over all the possible values of $\mathbf{X}(0), \ldots, \mathbf{X}(t)$, except for $\mathbf{N}(t)$ which is inside $\mathbf{X}(t) = [\mathbf{N}(t)\ p(t)\ E(t)\ S(t)]^T$. Using (9.50) in (9.55), we are able to obtain $P(\mathbf{N}(t))$ as follows:

$$P(\mathbf{N}(t)) = \sum_{\sim \mathbf{N}(t)} \left\{ P(\mathbf{X}(0)) \prod_{u=1}^{t-1} \left[\mathbb{1}_{[\text{Compliance}(u)]} \mathbf{P}_{\text{trans}}(\mathbf{N}(u), p(u)) \right] \right\}. \tag{9.56}$$

Expanding (9.56) using Lemmas 9.1 and 9.2, we have:

$$\begin{aligned} P(\mathbf{N}(t)) = & \sum_{p^{(1)}(0) \in \mathscr{V}} \sum_{p^{(2)}(0) \in \mathscr{V}} \cdots \sum_{p^{(K)}(0) \in \mathscr{V}} \cdots \sum_{p^{(K)}(t) \in \mathscr{V}} \\ & \times \sum_{N_1^{(1)}(0)=1}^{N_{1\max}^{(1)}(0)} \sum_{N_1^{(2)}(0)=1}^{N_{1\max}^{(2)}(0)} \cdots \sum_{N_V^{(K)}(0)=1}^{N_{V\max}^{(K)}(0)} \cdots \sum_{N_V^{(K)}(t-1)=1}^{N_{V\max}^{(K)}(t-1)} \\ & \times \sum_{E^{(1)}(0) \in \mathscr{D}_\mathrm{E}} \sum_{E^{(2)}(0) \in \mathscr{D}_\mathrm{E}} \cdots \sum_{E^{(K)}(0) \in \mathscr{D}_\mathrm{E}} \cdots \sum_{E^{(K)}(t) \in \mathscr{D}_\mathrm{E}} \\ & \times \sum_{S^{(1)}(0)=0}^{1} \sum_{S^{(2)}(0)=0}^{1} \cdots \sum_{S^{(K)}(0)=0}^{1} \cdots \sum_{S^{(K)}(t)=0}^{1} \\ & \left\{ P(\mathbf{X}(0)) \prod_{u=1}^{t-1} \left[\mathbb{1}_{[\text{Compliance}(u)]} \mathbf{P}_{\text{trans}}(\mathbf{N}(u), p(u)) \right] \right\}. \end{aligned} \tag{9.57}$$

The summations in the first line of (9.57) account for passing through all possible values of $p(0), \ldots, p(t)$. The summations in the second line are responsible for passing through all reachable values of $\mathbf{N}(0), \ldots, \mathbf{N}(t-1)$, where the upper limit is set with the aid of Lemma 9.1. The third line supplies the summation over all possible values of $E(0), \ldots, E(t)$, in which it is utilized the set $\mathscr{D}_E$ defined in Lemma 9.2. Lastly, the fourth line summations cover all the values of $S(0), \ldots, S(t)$. Note that the logical expression Compliance(u) and the transition matrix inside the product are built up from all these summation indices previously fixed.

Remark 9.3. An interesting feature added by this theoretical analysis is that the particle competition model can also accept uncertainty revolving around its initial state, i.e., $P(\mathbf{X}(0)) = P(\mathbf{N}(0), p(0), E(0), S(0))$. In other terms, the particles' initial locations can be conceptualized as a true distribution itself.

9.3.1.7 Discovering the Distribution of the Domination Level Matrix $P(\mathbf{N}(t))$

The distribution of the domination level matrix $\bar{\mathbf{N}}(t)$ is the fundamental information needed to group up the vertices. First, one can observe that positive integer multiples of $\mathbf{N}(t)$ compose the same $\bar{\mathbf{N}}(t)$. Therefore, the mapping $\mathbf{N}(t) \rightarrow \bar{\mathbf{N}}(t)$ is not injective; hence, not invertible. Below, an illustrative example shows this property.

Example 9.4. Consider a network with 3 particles and 2 vertices. At time t, suppose that the random process is able to produce two distinct configurations for $\mathbf{N}(t)$, as follows:

$$\begin{aligned} \mathbf{N}(t) &= \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix}, \\ \mathbf{N}'(t) &= \begin{bmatrix} 2 & 2 & 2 \\ 2 & 4 & 6 \end{bmatrix}. \end{aligned} \tag{9.58}$$

Then, the setups in (9.58) applied to (9.7) make clear that both configurations yield the same $\bar{\mathbf{N}}(t)$ given by:

$$\bar{\mathbf{N}}(t) = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/6 & 1/3 & 1/2 \end{bmatrix}. \tag{9.59}$$

In view of this, the mapping $\mathbf{N}(t) \rightarrow \bar{\mathbf{N}}(t)$ cannot be injective nor invertible.

Before proceeding further with the deduction of how to calculate $\bar{\mathbf{N}}(t)$ from $\mathbf{N}(t)$, let us present some helpful auxiliary results.

Lemma 9.3. *For any given time t, the following assertions hold* $\forall (i,k) \in \mathscr{V} \times \mathscr{K}$*:*

(a) The minimum value of $\bar{\mathbf{N}}_i^{(k)}(t)$ *is:*

$$\bar{\mathbf{N}}_{i_{\min}}^{(k)}(t) = \frac{1}{1 + \sum_{u \in \mathscr{K} \setminus \{k\}} \mathbf{N}_{i_{\max}}^{(u)}(t)}. \tag{9.60}$$

(b) The maximum value of $\bar{\mathbf{N}}_i^{(k)}(t)$ *is:*

$$\bar{\mathbf{N}}_{i_{\max}}^{(k)}(t) = \frac{\mathbf{N}_{i_{\max}}^{(k)}(t)}{\mathbf{N}_{i_{\max}}^{(k)}(t) + (K-1)}. \tag{9.61}$$

Proof. (a) According to (9.7), the minimum value occurs if three conditions are met: (i) particle k is not initially spawned at vertex i; (ii) particle k never visits vertex i; and (iii) all other $K-1$ particles $u \in \mathscr{K} \setminus \{k\}$ visit vertex i in the quickest possible manner, i.e., they follow the trajectory given in Lemma 9.1. In this way, vertex i is visited $\sum_{u\in\mathscr{K}\setminus\{k\}} \mathbf{N}_{i_{\max}}^{(u)}(t)$ times by other particles. However, having in mind the initial condition of $\mathbf{N}(0)$ shown in the second expression of (9.24), we must add 1 to the total number of visits received by vertex i. By virtue of that, it is expected that the total number of visits to be $1 + \sum_{u\in\mathscr{K}\setminus\{k\}} \mathbf{N}_{i_{\max}}^{(u)}(t)$. In view of this scenario, applying (9.7) to this configuration yields (9.60).

(b) The maximum value happens if three conditions are satisfied: (i) particle k initially spawns at vertex i; (ii) particle k visits i in the quickest possible manner; and (iii) all of the other particles $u \in \mathscr{K} \setminus \{k\}$ never visit i. In this scenario, vertex i receives $\mathbf{N}_{i_{\max}}^{(k)}(t) + (K-1)$ visits, where the second term in the summation is due to the initialization of $\mathbf{N}(0)$, as the second expression in (9.24) reveals. This information, together with (9.7), implies (9.61). ■

Remark 9.4. If the network does not have self-loops, then (9.60) reduces to:

$$\bar{\mathbf{N}}_{i_{\min}}^{(k)}(t) = \frac{1}{1 + (K-1)\mathbf{N}_{i_{\max}}^{(k)}(t)}. \tag{9.62}$$

The following Lemma provides all reachable elements that an arbitrary entry of $\bar{\mathbf{N}}(t)$ can have.

Lemma 9.4. *Denote* $^{\text{num}}/_{\text{den}}$ *as an arbitrary irreducible fraction. Consider that the set* $\mathscr{I}_t$ *retains all the reachable values of* $\bar{\mathbf{N}}_i^{(k)}(t)$, $\forall (i,k) \in \mathscr{V} \times \mathscr{K}$, *for a fixed t. Then, the elements of* $\mathscr{I}_t$ *are composed of all elements satisfying the following constraints:*

(a) The minimum element is given by the expression in (9.60).
(b) The maximum element is given by the expression in (9.61).
(c) All the irreducible fractions within the interval delimited by (a) and (b) such that:

I. num, den $\in \mathbb{N}^*$;
II. num $\leq \mathbf{N}_{i_{\max}}^{(k)}(t)$;
III. den $\leq \sum_{u\in\mathscr{K}} \mathbf{N}_{i_{\max}}^{(u)}(t)$.

Proof. (a) and (b) Straightforward from Lemma 9.3; (c) Firstly, we need to remember that $\mathbf{N}_i^{(k)}(t)$ may only take integer values. According to (9.7), $\bar{\mathbf{N}}_i^{(k)}(t) = {}^{\mathrm{num}}/_{\mathrm{den}}$ is a ratio of integer numbers. As a consequence, num and den must be integers and clause I is demonstrated. In view of (9.7), num only registers visits performed by a single particle. Therefore, the upper bound of it is established by Lemma 9.1, i.e., $\mathbf{N}_{i_{\max}}^{(k)}(t)$. Hence, clause II is proved. Looking at the same expression, observe that den registers the number of visits performed by all particles. Again, using Lemma 9.1 proves clause III. ■

Another interesting feature of the set $\mathscr{I}_t$ is elucidated in the following Lemma.

Lemma 9.5. *Given $t \leq \infty$, the set $\mathscr{I}_t$ indicated in Lemma 9.4 is always finite.*

Proof. In order to demonstrate this lemma, it is enough to verify that each set appearing in the *caput* of Lemma 9.4 is finite.

(a) are (b) are scalars, hence, they are finite sets. (c) Clause I indicates a lower bound for the numerator and the denominator. Clauses II and III reveal upper bounds for the numerator and denominator, respectively. Also from clause I, it can be inferred that the interval delimited by the lower and upper bounds is discrete. Therefore, the number of irreducible fractions that can be made from these two limits is finite.

As all the sets are finite for any t, since $\mathscr{I}_t$ is the union of all these subsets, it follows that $\mathscr{I}_t$ is also finite for any t. ■

Lemma 9.4 supplies the reachable values of an arbitrary entry of $\bar{\mathbf{N}}(t)$ by means of the definition of the set $\mathscr{I}_t$. Next, we simply extend this notion to the space spawned by the matrices $\bar{\mathbf{N}}(t)$ with dimensions $V \times K$, in such a way that each entry of it must be an element of $\mathscr{I}_t$ as follows:

$$\mathscr{M}_t \triangleq \{\bar{\mathbf{N}} : \bar{\mathbf{N}}_i^{(k)}(t) \in \mathscr{I}_t, \forall (i,k) \in \mathscr{V} \times \mathscr{K}\}. \tag{9.63}$$

In light of all these previous consideration, we can now provide a compact way of determining the distribution of $\bar{\mathbf{N}}(t)$. Following the aforementioned strategy, $P(\bar{\mathbf{N}}(t))$ can be calculated by summing over all multiples of $u\mathbf{N}(t)$, $u \in \{1, \ldots, t\}$ such that $f(u\mathbf{N}(t)) = \bar{\mathbf{N}}(t)$, where f is the normalization function defined in (9.7). On account of this, we have:

$$P\big(\bar{\mathbf{N}}(t) = \mathbf{U} : \mathbf{U} \in \mathscr{M}_t\big) = \sum_{u=1}^{t} P\big(f(u\mathbf{N}(t)) = \mathbf{U}\big), \tag{9.64}$$

in which the upper limit provided in summation of (9.64) is taken using a conservative approach. Indeed, the probability for events such that $\mathbf{N}_i^{(k)}(t) > \mathbf{N}_{i_{\max}}^{(k)}(t)$ is zero. By virtue of that, we can stop summing whenever any entry of matrix $u\mathbf{N}(t)$ exceeds this value. We have omitted this observation from (9.64) for the sake of clarity.

As $t \to \infty$, $P(\bar{\mathbf{N}}(t))$ provides enough information for grouping the vertices. In this case, they are grouped accordingly to the particle that is imposing the highest domination level. Since the domination level is a continuous random variable, the output of this model is fuzzy.

9.3.2 Linking the Particle Competition Model and the Classical Multiple Independent Random Walks System

Multiple random walks are modeled as dynamical systems and have been extensively studied by the literature [8]. In these systems, particles cannot communicate with each other. In effect, the model of multiple random walks can be understood as a system with multiple single random walks stacked up. The particle competition model that we have explored in this chapter, however, permits communication between different particles. The communication is modeled via the domination level matrix, which encodes the fraction of visits each vertex has received from particles in the network. This happens because the fraction of visits is computed by a normalization procedure that effectively entangles the walking dynamic of all particles with one another.[1]

The interaction or communication between particles, nonetheless, can be turned off when $\lambda = 0$ and $\Delta = 0$. This is equivalent to saying that the particle competitive model investigated in this chapter is a generalization of the classical dynamical system of multiple independent random walkers. Whenever $\lambda > 0$, the competitive mechanism is enabled and the combination of random-preferential interacting walks occurs. In this case, the reanimation feature is presented depending on the choice of Δ.

We now prove the assertion that when $\lambda = 0$ and $\Delta = 0$ holds, the particle competition model produces the same dynamics of multiple independent random walks.

Proposition 9.2. *If $\lambda = 0$ and $\Delta = 0$, then system ϕ reduces to the case of multiple independent random walks.*

Proof. First, note that, when $\lambda = 0$, the influence of the transition matrix that encodes the preferential movement, $\mathbf{P}_{\text{pref}}(t)$, is taken away. Indeed, when $\lambda = 0$, the coupling between $\mathbf{N}(t)$ and $p(t)$ ceases to exist, because the calculation step of $P_{\text{pref}}(t)$ (responsible for the coupling) can be skipped. Moreover, if $\Delta = 0$, then the particles can never get exhausted. In view of these characteristics, the dynamical system ϕ can be easily described by a traditional Markovian process given by:

$$p(t+1) = p(t)\mathbf{P}_{\text{transition}}, \tag{9.65}$$

[1] Recall the evaluation of each entry of the domination matrix in (9.7).

in which $\mathbf{P}_{\text{transition}} = \mathbf{P}_{\text{rand}} \otimes \mathbf{P}_{\text{rand}} \otimes \ldots \otimes \mathbf{P}_{\text{rand}}$ and $p(t)$ is an enumerated state encompassing all the particles, as described before. Here, the independence among the particles is demonstrated by showing that the generated $\mathbf{N}(t)$ by system ϕ is exactly the same as the one produced by the potential matrix of the Markov chains theory as introduced in Definition 2.68. In other words, $\mathbf{N}(t)$ can be implicitly calculated from the stochastic process $\{p(t) : t \geq 0\}$.

We now find a closed expression for $\mathbf{N}(t)$ in terms of $\mathbf{N}(0)$. This can be easily done if we iterate the matrix equation $\mathbf{N}(t+1) = \mathbf{N}(t) + Q$, where Q is as given in (9.40). In doing so, we get:

$$\mathbf{N}(t) = \begin{bmatrix} 1 & \cdots & 1 \\ 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} + \sum_{i=0}^{t} \begin{bmatrix} \mathbb{1}_{[p^{(1)}(i)=1]} & \cdots & \mathbb{1}_{[p^{(K)}(i)=1]} \\ \mathbb{1}_{[p^{(1)}(i)=2]} & \cdots & \mathbb{1}_{[p^{(K)}(i)=2]} \\ \vdots & \ddots & \vdots \\ \mathbb{1}_{[p^{(1)}(i)=V]} & \cdots & \mathbb{1}_{[p^{(K)}(i)=V]} \end{bmatrix}. \tag{9.66}$$

Since this process is stochastic, it is worth determining the expectation of the number of visits $\mathbf{N}(t)$ given the particle's initial location $p(0)$. Noting that $\mathbb{E}[\mathbb{1}_{[A]}] = P(A)$, we have:

$$\mathbb{E}[\mathbf{N}(t) \mid p(0)] = \begin{bmatrix} 1 & \cdots & 1 \\ 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} + \sum_{i=0}^{t} \begin{bmatrix} \mathbf{P}^{i}(p_1(0), 1) & \cdots & \mathbf{P}^{i}(p_K(0), 1) \\ \mathbf{P}^{i}(p_1(0), 2) & \cdots & \mathbf{P}^{i}(p_K(0), 2) \\ \vdots & \ddots & \vdots \\ \mathbf{P}^{i}(p_1(0), V) & \cdots & \mathbf{P}^{i}(p_K(0), V) \end{bmatrix}, \tag{9.67}$$

in which $\mathbf{P}^{i}(p_j(0), 1)$ denotes the $(p_j(0), 1)$-entry of $\mathbf{P}_{\text{transition}}$ to the i-th power. But, from the Markov chains theory, we have that the so-called truncated potential matrix [8] is given by:

$$R_t(v, k) \triangleq \sum_{i=0}^{t} \mathbf{P}^{i}_{\text{transition}}(v, k). \tag{9.68}$$

By virtue of (9.68), each entry of the matrix equation in (9.67) can be rewritten as:

$$\mathbb{E}[\mathbf{N}_i^{(j)}(t) \mid p(0)] = 1 + R_t(p_j(0), i). \tag{9.69}$$

From (9.69), we can infer that each particle does perform an independent random walk according to a Markov Chain. Thus, we are able to conclude that, for $\lambda = 0$ and $\Delta = 0$, all the states of system ϕ follow a traditional Markov Chain process, except for a constant, as demonstrated in (9.69). ■

Proposition 9.2 states that system ϕ reduces to the case of multiple random walks when $\lambda = 0$ and $\Delta = 0$, i.e., we could think that there is a blind competition among the participants. Alternatively, when $0 < \lambda \leq 1$, some orientation is given to the participants, in the sense of defending their territory and not only keep adventuring through the network with no strategy at all. In either case, the reanimation procedure is enabled depending on the choice of Δ.

9.3.3 A Numerical Example

For the sake of clarity, we provide an example showing how to use the theoretical results supplied in the previous section. We limit the demonstration for a single iteration, which is the transition from $t = 0$ to $t = 1$. The simple example is composed of a trivial 3-vertex regular network, identical to the one in Fig. 9.10a. For the referred example, suppose there are $K = 2$ particles into the network, i.e., $\mathscr{K} = \{1, 2\}$. Let particle 1 be spawned at vertex 1 and particle 2 at vertex 2, i.e., we have certainty about the initial locations of the particles at $t = 0$:

$$P(\mathbf{X}(0)) = P\left(\mathbf{N}(0) = \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 1 & 1 \end{bmatrix}, p(0) = [1 \;\; 2], E(0), S(0)\right) = 1, \tag{9.70}$$

i.e., there is 100 % chance (certainty) that particles 1 and 2 are generated at vertices 1 and 2, respectively. Observe that $\mathbf{N}(0)$, $E(0)$, and S(0) are chosen such as to satisfy (9.24), (9.25), and (9.26), respectively. Otherwise, the probability should be 0, in view of (9.45). It is worth emphasizing that the competitive model accepts uncertainty about the initial location of the particles, in a way that we could specify different probabilities to each particle to spawn at different locations. This characteristic is not present in [32], in which it must be fixed a certain position for each particle.

From Fig. 9.10a we can deduce the adjacency matrix $\mathbf{A}$ of the network and, therefore, determine the transition matrix associated to the random movement term for a single particle. Recall that the random matrix is the same for all of the particles. Then, applying (9.2) on $\mathbf{A}$, we get:

$$\mathbf{P}_{\text{rand}} = \begin{bmatrix} 0 & 0.50 & 0.50 \\ 0.50 & 0 & 0.50 \\ 0.50 & 0.50 & 0 \end{bmatrix}. \tag{9.71}$$

Given $\mathbf{N}(0)$, we can readily establish $\bar{\mathbf{N}}(0)$ with the aid of (9.7):

$$\bar{\mathbf{N}}(0) = \begin{bmatrix} 0.67 & 0.33 \\ 0.33 & 0.67 \\ 0.50 & 0.50 \end{bmatrix}. \tag{9.72}$$

Using (9.8) we are able to calculate the matrices associated to the preferential movement policy for each particle in the network as:

$$\mathbf{P}^{(1)}_{\text{pref}}(0) = \begin{bmatrix} 0 & 0.40 & 0.60 \\ 0.57 & 0 & 0.43 \\ 0.67 & 0.33 & 0 \end{bmatrix}, \tag{9.73}$$

$$\mathbf{P}^{(2)}_{\text{pref}}(0) = \begin{bmatrix} 0 & 0.57 & 0.43 \\ 0.40 & 0 & 0.60 \\ 0.33 & 0.67 & 0 \end{bmatrix}. \tag{9.74}$$

In order to ease the calculations, let us assume that $\lambda = 1$, so that (9.20) reduces to $\mathbf{P}_{\text{transition}}(0) = \mathbf{P}^{(1)}_{\text{pref}}(0) \otimes \mathbf{P}^{(2)}_{\text{pref}}(0)$ at time $t = 0$,[2] which is a matrix with dimensions 9×9 that is given by:

$$\mathbf{P}_{\text{transition}}(0) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0.228 & 0.172 & 0 & 0.342 & 0.258 \\ 0 & 0 & 0 & 0.160 & 0 & 0.240 & 0.240 & 0 & 0.360 \\ 0 & 0 & 0 & 0.132 & 0.268 & 0 & 0.198 & 0.402 & 0 \\ 0 & 0.325 & 0.245 & 0 & 0 & 0 & 0 & 0.245 & 0.185 \\ 0.228 & 0 & 0.342 & 0 & 0 & 0 & 0.172 & 0 & 0.258 \\ 0.188 & 0.382 & 0 & 0 & 0 & 0 & 0.142 & 0.288 & 0 \\ 0 & 0.382 & 0.288 & 0 & 0.188 & 0.142 & 0 & 0 & 0 \\ 0.268 & 0 & 0.402 & 0.132 & 0 & 0.198 & 0 & 0 & 0 \\ 0.221 & 0.449 & 0 & 0.109 & 0.221 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{9.75}$$

Since in the initial condition depicted in (9.70) particles 1 and 2 start out at vertices 1 and 2, respectively, the enumerated scalar state for the matters of calculating $p(t+1)$ is $(1,2) \rightarrow 2$. Hence, we turn our attention to the second row of $\mathbf{P}_{\text{transition}}(0)$, which completely characterizes the transition probabilities for the next state of the dynamical system. A quick analysis of the second row in (9.75) shows that, out of the 9 possible "next states" of the system, only 4 are plausible. (The remaining states have probability 0 to be reached.) In this way:

$$P\left(\mathbf{N}(1) = \begin{bmatrix} 2 & 2 \\ 2 & 2 \\ 1 & 1 \end{bmatrix},\quad p(1) = \begin{bmatrix} 2 & 1 \end{bmatrix}, E(1), S(1) \mid \mathbf{X}(0)\right) = 0.160, \tag{9.76}$$

$$P\left(\mathbf{N}(1) = \begin{bmatrix} 2 & 1 \\ 2 & 2 \\ 1 & 2 \end{bmatrix},\quad p(1) = \begin{bmatrix} 2 & 3 \end{bmatrix}, E(1), S(1) \mid \mathbf{X}(0)\right) = 0.240, \tag{9.77}$$

[2] Recall that all particles are active at the initial state in view of (9.26).

$$P\left(\mathbf{N}(1) = \begin{bmatrix} 2 & 2 \\ 1 & 2 \\ 2 & 1 \end{bmatrix}, \quad p(1) = \begin{bmatrix} 3 & 1 \end{bmatrix}, E(1), S(1) \mid \mathbf{X}(0)\right) = 0.240, \tag{9.78}$$

$$P\left(\mathbf{N}(1) = \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 2 & 2 \end{bmatrix}, \quad p(1) = \begin{bmatrix} 3 & 3 \end{bmatrix}, E(1), S(1) \mid \mathbf{X}(0)\right) = 0.360, \tag{9.79}$$

in which $\mathbf{X}(0)$ is as given in (9.70). Equations (9.76)–(9.79) match our intuition if we take a careful look at Fig. 9.10a: self-looping is not allowed, so the state space that is probabilistically possible of happening can only be these previous 4 states. In other terms, starting from vertex 1, there are only two different choices that the particle can make: either visit vertex 2 or 3. The same reasoning can be applied when we start at vertex 2. Since it is a joint distribution, we multiply these factors, which totalizes 4 different states. Furthermore, as we have fixed $\lambda = 1$, it is expected that the transition probabilities will be heavily dependent on the domination levels imposed on the neighboring vertices. In this case, strongly dominated vertices constitute repulsive forces that act against rival particles. In this regard, the preferential or defensive behavior of these particles prevents particles from visiting these type of vertices. This is exactly symbolized in (9.79), which denotes the transition probability $(1, 2) \rightarrow (3, 3)$ and also possesses the highest transition probability, in account of the neutrality of vertex 3, as opposed to the remaining two vertices.

Remark 9.5. Alternatively, we could have used the collection of two matrices 3×3, as given in (9.73) and (9.74) with no loss of generality. Here, we clarify this concept by calculating a single entry of $\mathbf{P}_{\text{transition}}(0)$ using this methodology. Consider we are to calculate the probability according to (9.76), i.e., particle 1 performs a transition from vertex 1 to vertex 2 and particle 2 executes a transition from vertex 2 to vertex 1. For the former case, according to the particle 1's transition matrix (see (9.73)) we have $\mathbf{P}^{(1)}_{\text{pref}}(0)(1, 2) = 0.40$. Likewise, for the last case (see (9.74)), we have $\mathbf{P}^{(2)}_{\text{pref}}(0)(2, 1) = 0.40$. Remembering that $p(0) = [1\ 2]$ in a scalar form corresponds to the second state of $\mathbf{P}_{\text{transition}}$ and $p(1) = [2\ 1]$ corresponds to the fourth state, then $\mathbf{P}_{\text{transition}}(0)(2, 4) = \mathbf{P}^{(1)}_{\text{pref}}(0)(1, 2) \times \mathbf{P}^{(2)}_{\text{pref}}(0)(2, 1) = 0.40 \times 0.40 = 0.16$, which is equal to the corresponding entry of the matrix in (9.75).

Before doing the calculation of the marginal distribution $P(\mathbf{N}(1))$, we are required to fix an upper limit for an arbitrary entry of the matrix $\mathbf{N}(1)$. This is readily evaluated from (9.51), which results in $\mathbf{N}^{(k)}_{i_{\max}}(1) = \mathbf{N}^{(k)}_{i_{\max}}(1) = 2$, $\forall (i, k) \in \mathscr{V} \times \mathscr{K}$, implying that we are only needed to take all numerical combinations for the matrix $\mathbf{N}(0)$ such that each entry may only take the values $\{1, 2\}$, since larger values yield probability 0 according to Lemma 9.1. Moreover, we need to iterate through every feasible value of every entry of $E(0)$ and $E(1)$. In order to do so, we fix $\Delta = 0.25$, $\omega_{\min} = 0$, and $\omega_{\max} = 1$. With that, we are able to make use of Lemma 9.2, which yields $E(t) \in \{0, 0.25, 0.5, 0.75, 1\}$. The limits of the remaining system variables $S(0)$ and $S(1)$ are straightforward. In the present conditions, we have enough information to calculate the marginal distribution $P(\mathbf{N}(1))$, according to (9.57):

$$P\left(\mathbf{N}(1) = \begin{bmatrix} 2 & 2 \\ 2 & 2 \\ 1 & 1 \end{bmatrix}\right) = 1 \times 0.160 = 0.160, \tag{9.80}$$

$$P\left(\mathbf{N}(1) = \begin{bmatrix} 2 & 1 \\ 2 & 2 \\ 1 & 2 \end{bmatrix}\right) = 1 \times 0.240 = 0.240, \tag{9.81}$$

$$P\left(\mathbf{N}(1) = \begin{bmatrix} 2 & 2 \\ 1 & 2 \\ 2 & 1 \end{bmatrix}\right) = 1 \times 0.240 = 0.240, \tag{9.82}$$

$$P\left(\mathbf{N}(1) = \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 2 & 2 \end{bmatrix}\right) = 1 \times 0.360 = 0.360. \tag{9.83}$$

As the last goal, our task is to determine the distribution $P(\bar{\mathbf{N}}(1))$. According to the specified steps in the previous section, we need to find all irreducible fractions that lie within the interval [0, 1] with the constraints derived in the previous section. This means that we only have to consider entries of matrix $\bar{\mathbf{N}}(t)$ that contain elements of $\mathscr{I}_1$; the remainder $\bar{\mathbf{N}}(t)$ are infeasible and, thus, occur with probability 0. In view of the constraints previously enumerated, $\mathscr{I}_1 = \{1/4, 1/3, 1/2, 2/3, 3/4\}$. Observing that we have the complete distribution of $\mathbf{N}(1)$, it is an easy task to apply (9.64), as follows:

$$P\left(\bar{\mathbf{N}}(1) = \begin{bmatrix} 1/2 & 1/2 \\ 1/3 & 2/3 \\ 1/2 & 1/2 \end{bmatrix}\right) = 0.160, \tag{9.84}$$

$$P\left(\bar{\mathbf{N}}(1) = \begin{bmatrix} 2/3 & 1/3 \\ 1/2 & 1/2 \\ 1/3 & 2/3 \end{bmatrix}\right) = 0.240, \tag{9.85}$$

$$P\left(\bar{\mathbf{N}}(1) = \begin{bmatrix} 1/2 & 1/2 \\ 1/3 & 2/3 \\ 2/3 & 1/3 \end{bmatrix}\right) = 0.240, \tag{9.86}$$

$$P\left(\bar{\mathbf{N}}(1) = \begin{bmatrix} 2/3 & 1/3 \\ 1/3 & 2/3 \\ 1/2 & 1/2 \end{bmatrix}\right) = 0.360. \tag{9.87}$$

It is noteworthy to reinforce that the mapping between the probabilities of $\mathbf{N}(t)$ and $\bar{\mathbf{N}}(t)$ is not bijective: in this special simple case that we are studying, we did not have distinct $\mathbf{N}(t)$ that could generate the same $\bar{\mathbf{N}}(t)$, but as t increases, this is likely to happen quite frequently. This process is repeated until a sufficiently large t or until the system converges to a quasi-stationary state of $\bar{\mathbf{N}}(t)$, as discussed in the empirical section.

9.4 Numerical Analysis of the Detection of Overlapping Vertices and Communities

In this section, some simulation results are presented with the purpose of assessing the effectiveness of the particle competition technique on detecting overlapping vertices and communities. Note that the index that estimates the overlapping nature of each network vertex is computed using (9.28). The obtained results are also compared to classical overlap vertex measures [11, 12].

9.4.1 Zachary's Karate Club Network

First, the particle competition technique is applied to detect fuzzy community structure in the Zachary's "karate club" network [42]. 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 34 members of a club at an American University in the 1970s. The members are represented by vertices and an edge exists between two members if they know each other. Shortly after the formation of the network, the club dismembered in two as the consequence of an internal dispute, making it an interesting problem for detecting communities. Figure 9.11 shows the outcome of the community detection task. The red (dark gray) and blue (gray) colors denote the communities detected by the algorithm. Only vertex 3 (the yellow or light gray vertex) is incorrectly grouped as a member of the red (dark gray) community. In the literature, vertices 3 (e.g., see [13]) and 10 (e.g., see [29]) are often misclassified by many community detection algorithms. This happens because the number of edges that they share between the two communities is the same, i.e., they are inherently overlapping, making their clustering a hard problem. We apply the overlapping index of the particle competition model and report the results in Fig. 9.12. We see

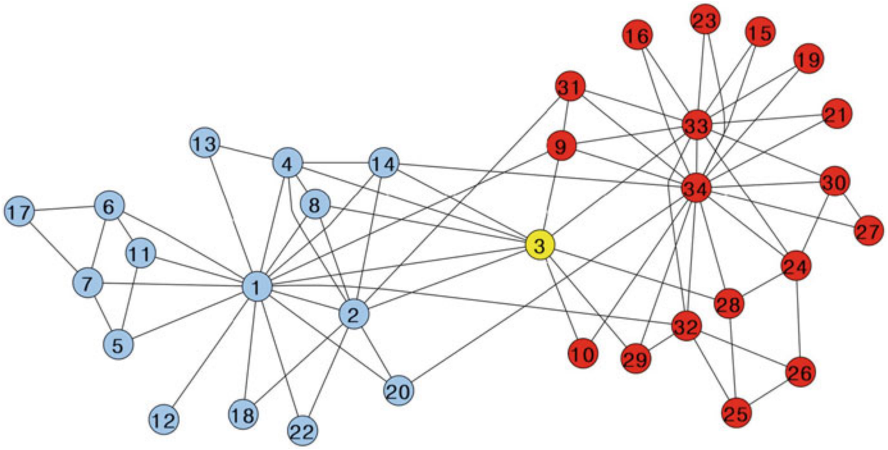

Fig. 9.11 Community detection result in the Zachary's karate club network when we apply the particle competition method. The *red* (*dark gray*) and *blue* (*gray*) *colors* denote the detected communities. Only the *yellow* or *light gray* vertex (vertex 3 in the original database) is incorrectly grouped. Reproduced from [36] with permission from Elsevier

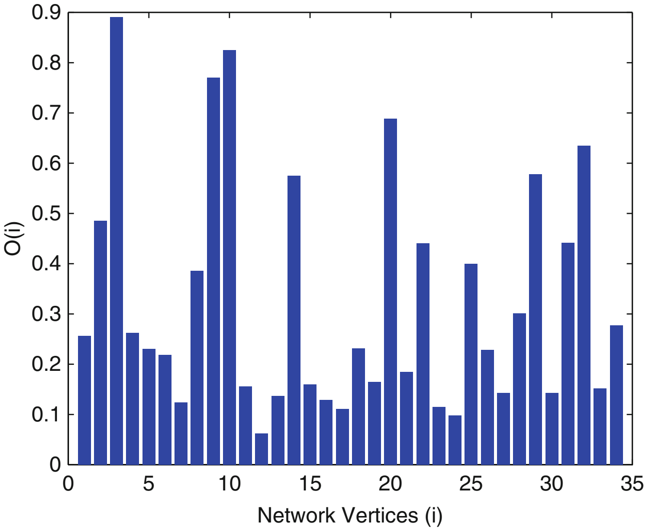

Fig. 9.12 Result of the calculation of the overlapping index for all vertices in the Zachary's "karate club" network. Reproduced from [36] with permission from Elsevier

that vertices 3 and 10 show the highest overlapping indices, confirming the previous analysis. Moreover, vertices 9, 14, 20, 29, and 32 also present a significant level of overlapping characteristics, since these are placed in the borders of each community.

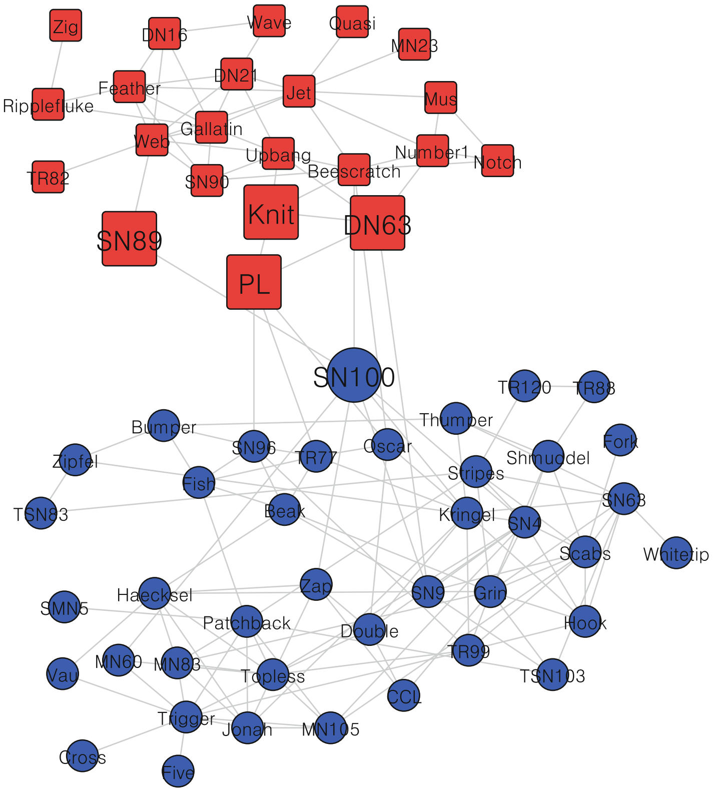

Fig. 9.13 Dolphin Social Network observed by Lusseau. $K = 2$ and $\lambda = 0.6$. The five vertices with the highest overlapping structure are displayed with larger sizes. Reproduced from [36] with permission from Elsevier

9.4.2 *Dolphin Social Network*

The Dolphin Social Network [26] is composed of 62 bottlenose dolphins living in Doubtful Sound, New Zealand. In this case, we consider that the dolphins represent the vertices, whereas edges between dolphin pairs are established by observation when there is a statistically significant frequent association between them. Figure 9.13 indicates the community detection outcome of the particle competition technique. The five most overlapping vertices are displayed 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 split into two communities seems to match the known division of the dolphin community, except for the dolphin "PL," which is a member of the blue (gray) community. Based on a 2-year research period, Lusseau reported that the bottlenose dolphins segregated into two communities, apparently by virtue of the disappearance of the dolphins located at the boundaries of each of 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 particle competition algorithm captures as the vertices with the most overlapping nature, as we can verify in Fig. 9.13 from the larger vertices, i.e., “DN63,” “Knit,” “PL,” “SN89,” and “SN100.”

9.4.3 *Les misérables* Novel Network

Les Misérables is an interaction network between major characters comprising the Victor Hugo’s sprawling novel of crime and redemption in post-restoration France. Using the list of 77 character appearances by scene, compiled by Knuth [18], the network was constructed in a way that vertices represent characters and an edge between two vertices represents co-appearance of the corresponding characters in one or more scenes [30]. In this case, the quantity $\langle R(t) \rangle$ is maximized when $K = 6$. Figure 9.14 shows the outcome of the particle competition technique, along with the 10 most overlapping vertices portrayed in larger sizes. The communities clearly reflect the subplot structure of the book. As one can expect, the protagonist Jean Valjean and his nemesis, the police officer Javert, are captured as being the 2 most overlapping vertices of the network, since they are central to the Hugo’s play and form the hubs of communities composed of their respective adherents.

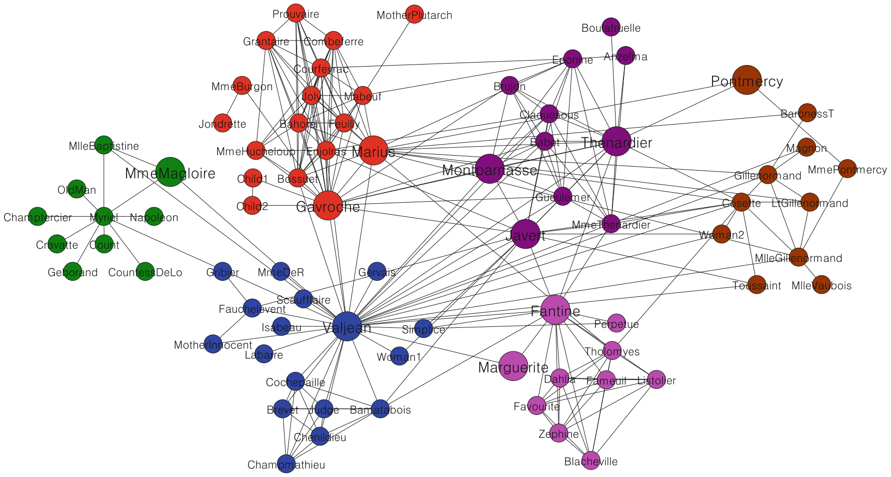

Fig. 9.14 Hugo’s sprawling novel of crime and redemption in post-restoration France entitled Les Misérables. $K = 6$ and $\lambda = 0.6$. The 10 vertices with the highest overlapping structure are depicted with larger sizes. Reproduced from [36] with permission from Elsevier

9.5 Application: Handwritten Digits and Letters Clustering

In order to give a concrete vision of the particle competition model, we here review the application in data clustering for the particle competitive method presented in [36]. We employ three real-world data sets that are composed of handwritten digits and letters, which are the USPS, the MNIST, and the Letter Recognition data sets.

Section 9.5.1 briefly discusses on the employed data sets. Section 9.5.2 applies the index for estimating the most likely number of groups in these three data sets. Finally, Sect. 9.5.3 provides the data clustering results.

9.5.1 *Brief Information of the Handwritten Digits and Letters Data Sets*

The data sets in which the particle competition model is tested against are given in the following:

- *USPS data set*: Comprised of 9298 images of handwritten digits. The digits 0 to 9 have 1553, 1269, 929, 824, 852, 716, 834, 792, 708, and 821 samples respectively. The US Postal Service (USPS) digits data were gathered at the Center of Excellence in Document Analysis and Recognition (CEDAR) at SUNY Buffalo, as part of a project sponsored by the US Postal Service. For more details about this data set, refer to [15]. Each image has dimensions of 16 × 16 pixels, with 256 grey levels per pixel. We employ the weighted eigenvalue similarity measure as defined in Sect. 8.6.3. Instead of using 16 eigenvalues, we only work with the four greatest ones. In this case, we use the following β function: $\beta(x) = 16\exp(\frac{x}{3})$.
- *MNIST data set*: Originally composed of images with dimensions 28 × 28. We only use the public set composed of 10,000 vertices. Moreover, we make use of the dissimilarity measure based on the first 4 eigenvalues of each image out of 28 eigenvalues. The same β function employed in the USPS data set is used here. More information is given in Sect. 8.6.2.
- *Letter Recognition data set*: Composed of characteristic vectors with 16 entries. There are 20,000 vertices.

Since none of these data sets are in a network form, the methodology is divided into two general steps: the network formation and the data clustering tasks. In the first, we use the k-nearest neighbor network formation technique with $k = 3$ after we apply a preprocessing step. In this preprocessing, we standardize the data such as to have zero mean and unitary standard deviation. As for the distance measure, we either use the weighted eigenvalue dissimilarity (for the first two data sets above) or the Euclidean distance (for the last one). The reason behind not using the weighted eigenvalue on the third data set is because the samples are not provided as images, but as image descriptors. Since the latter is formed by merely scalars, we cannot apply the dissimilarity measure. In the second step, the data clustering algorithm based on particle competition is applied. As we are dealing with unsupervised learning, we do not use any external information, such as labels or exogenous knowledge. Instead, we limit ourselves to discovering explicit or implicit relationships among the data by the mechanism of particle competition.

9.5.2 Determining the Optimal Number of Particles and Clusters

Figure 9.15a–c show the determination of the optimal K for the USPS, MNIST, and Letter Recognition data sets, respectively. One can verify that $\langle R(t)\rangle$ is maximized exactly when the number of particles is equal to the number of clusters in the network, confirming the effectiveness of such heuristic.

9.5.3 Handwritten Data Clustering

Here, we show "digit" and "letter" cluster detection results using the particle competition algorithm. Table 9.1 supplies details about the algorithms chosen for comparison matters. The genetic algorithm implemented in the Global Optimization Toolbox of MATLAB is used to optimize the parameters of the particle competition algorithm. Specifically, the λ parameter is optimized over the range $0.2 \leq \lambda \leq 0.8$ and its optimal values for the USPS, MNIST, and Letter Recognition data sets are 0.58, 0.60, 0.60, respectively. The number of particles to be inserted is determined according to the previous analysis, i.e., we choose the number of particles that maximizes the quantity $\langle R(t)\rangle$ measure, which are 10, 10, and 26 for the USPS, MNIST, and Letter Recognition data sets, respectively.

Table 9.2 presents the data clustering rate reached by the particle competition method and the aforementioned competing algorithms. Some of these results are readily extracted from [33] and [23]. Within this table, we have provided the Average Rank of each algorithm, which is calculated as follows: (1) for each data set we rank the algorithms according to their average performance (average data clustering accuracy), i.e., the best algorithm is ranked as 1, the second best one is ranked as 2, and so on; (2) for each algorithm, the Average Rank is given by the average value of its rank achieved in all the data sets. As we can verify by looking at the Average Rank column, the stochastic competition algorithm has reached one of the best positions, showing the effectiveness of the particle competition scheme.

In order to further verify the robustness of the particle competition method, we inspect the samples that compose a same cluster. Specifically, Figs. 9.16 and 9.17 show some samples of the clusters representing the pattern "2" and "5", respectively, of the MNIST data set. These samples are captured using the following strategy: we

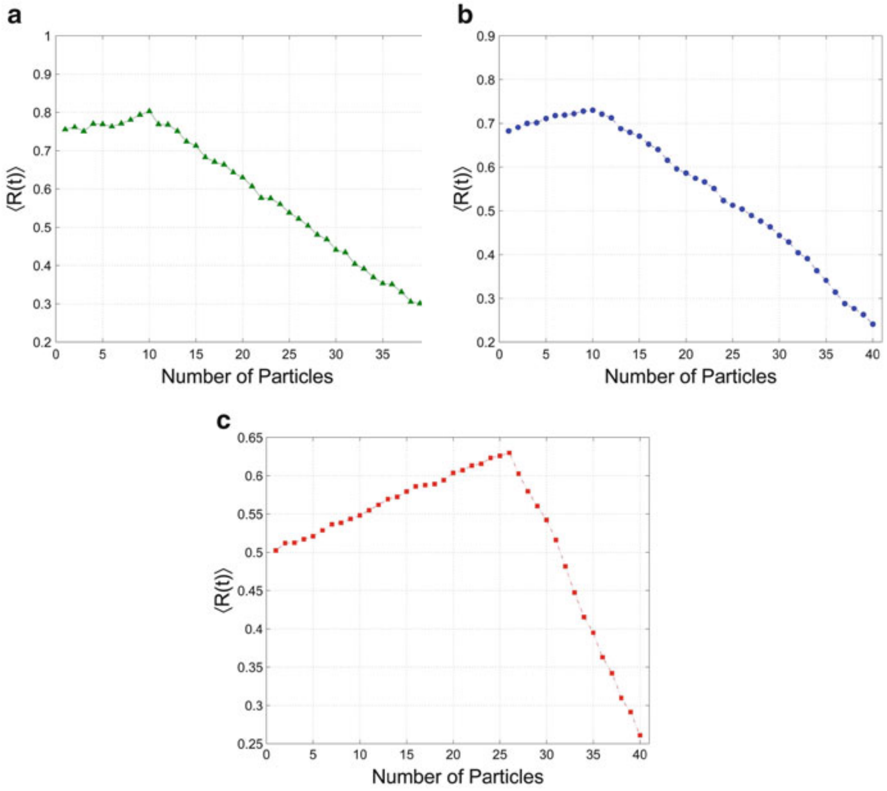

Fig. 9.15 Determination of the optimal number of particles *K* (the optimal number of clusters) in real-world data sets. The number of classes that each data set originally possesses are: (**a**) the USPS data set has 10 clusters (each cluster corresponding to a number from "0" to "9"); (**b**) the MNIST data set has 10 clusters (each cluster corresponding to a number from "0" to "9"); and (**c**) the Letter Recognition data set has 26 clusters (each cluster corresponding to a letter from the English alphabet ("A" to "Z")). 20 independent runs are performed and the average value is reported. Reproduced from [37] with permission from Springer

Table 9.1 Description of the competing state-of-the-art data clustering techniques

Technique	Reference
Gaussian mixture Model (GMM)	[5]
K-Means	[27]
Locally consistent Gaussian mixture Model (LCGMM)	[23]
Spectral clustering algorithm with normalized cut (Ncut)	[34]
Ncut embedding all (NcutEmb$^{\text{All}}$)	[33]
Ncut embedding maximum (NcutEmb$^{\text{Max}}$)	[33]

Reproduced from [37] with permission from Springer

Table 9.2 Data clustering accuracy reached by the particle competition technique and the competing methods listed in Table 9.1

	USPS	MNIST	Letter recognition	Avg. rank
LCGMM	73.83	73.60	93.03	2.33
GMM	67.30	66.60	91.24	5.33
K-Means	69.80	53.10	87.94	6.33
NCut	69.34	68.80	88.72	5.67
NCutEmb$^{\text{All}}$	72.72	75.10	90.07	3.67
NCutEmb$^{\text{Max}}$	72.97	75.63	90.59	2.67
Particle competition technique	80.46	74.53	91.37	2.00

For the stochastic methods, such as the particle competition method, thirty independent runs were performed and the corresponding mean is provided. Reproduced from [37] with permission from Springer

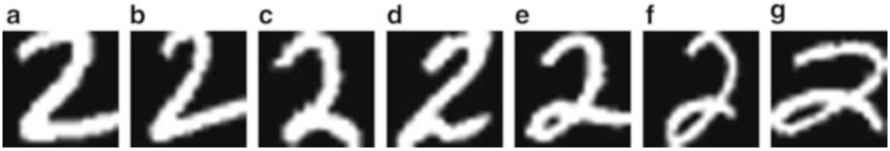

Fig. 9.16 A broad set of samples that were classified as being member of the cluster representing the pattern “2”. Note that samples that are adjacent are similar with regard to the weighted eigenvalue dissimilarity function. The transitions from the sample (**a**)–(**g**) were captured from the maximum geodesic distance between two vertices in the cluster representing pattern 2. In this case, the diameter of such cluster is 17. We have only provided 7 representative samples above. Reproduced from [37] with permission from Springer

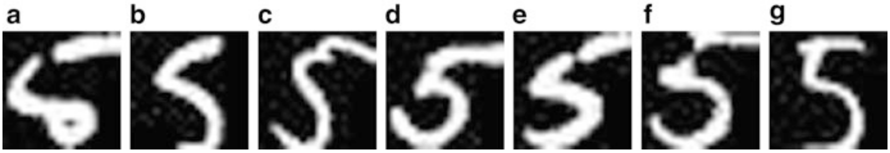

Fig. 9.17 A broad set of samples that were classified as being member of the cluster representing the pattern “5”. Likewise the previous figure, adjacent samples are more similar to each other than distance samples. Reproduced from [37] with permission from Springer

compute the vertices that compose the maximum geodesic distance of the cluster representing each pattern (cluster diameter). Now, we select a representative subset of vertices composing the cluster diameter trajectory for illustrative purposes. In these figures, samples that are adjacent are more similar than those distant from one to another. On the basis of this analysis, we conclude that the graph representation has successfully captured several variations of these number patterns each of which in a single representative cluster, showing the robustness of the model.

9.6 Chapter Remarks

In this chapter, a rigorous definition of a competitive learning scheme in complex networks has been studied, whose foundations are biologically inspired by the competition process taking place in many nature and social systems. In this model, several particles navigate in the network to explore their territory and, at the same time, attempt to defend their territory from rival particles. If a particle frequently visits a specific vertex, it occurs that the domination level of the visiting particle on that vertex is strengthened. Concurrently to that, the domination levels of all of the other particles on the same vertex are weakened. In the long-run dynamic, each particle is expected to be confined within a community of the network.

The particle competition model is nonlinear and stochastic. Owing to the mathematical formality that the model is built upon, theoretical and empirical analyses have been conducted to better understand the underlying properties of the competitive model. A convergence analysis has shown that the dynamical system presents structural stability rather than asymptotic stability. This is a welcomed characteristic, since it better describes the uncertainty that revolves around real-world problems, which have noise and uncontrolled variables. In addition, due to this analysis, we have found that the model is a generalization of the process of single independent random walkers in a network. Specifically, we have shown that the model's behavior acts as multiple interacting walkers in a network. The interaction is molded in a competitive way, by using a probabilistic convex combination of random and preferential walks. Such generalization is realized by calibrating the values of the parameter λ and Δ of the system. If $\lambda = 0$, the model reduces to multiple non-interacting random walks; but, when $\lambda > 0$, the interaction among particles is turned on.

Furthermore, measures for detecting overlapping structures and for estimating the number of actual clusters or communities in a network have been discussed, whose calculations are embedded into the model's own algorithm. This permits their calculation to be performed in an efficient way.

Simulations have been carried out with the purpose of quantifying the robustness of the particle competition scheme on artificial and real-world data sets for the tasks of data clustering and community detection. Computer simulations have revealed that the model works well for community detection and for data clustering tasks. Finally, an application on handwritten digits and letters recognition has been provided and high clustering accuracies have been obtained. Moreover, we have analyzed the composition of the clusters formed in the MNIST data set and have verified that, within a specific cluster, several variations of the same pattern can be encountered, confirming the robustness of the model.

References

1. Allinson, N., Yin, H., Allinson, L., Slack, J.: Advances in Self-Organising Maps. Springer, New York (2001)
2. Amorim, D.G., Delgado, M.F., Ameneiro, S.B.: Polytope ARTMAP: pattern classification without vigilance based on general geometry categories. IEEE Trans. Neural Netw. **18**(5), 1306–1325 (2007)
3. Athinarayanan, R., Sayeh, M.R., Wood, D.A.: Adaptive competitive self-organizing associative memory. IEEE Trans. Syst. Man Cybern. Part A **32**(4), 461–471 (2002)
4. Bacciu, D., Starita, A.: Competitive repetition suppression (CoRe) clustering: a biologically inspired learning model with application to robust clustering. IEEE Trans. Neural Netw. **19**(11), 1922–1940 (2008)
5. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer, New York (2007)
6. Carpenter, G.A., Grossberg, S.: Self-organization of stable category recognition codes for analog input patterns. Appl. Opt. **26**(23), 4919–4930 (1987)
7. Chen, M., Ghorbani, A.A., Bhavsar, V.C.: Incremental communication for adaptive resonance theory networks. IEEE Trans. Neural Netw. **16**(1), 132–144 (2005)
8. Çinlar, E.: Introduction to Stochastic Processes. Prentice-Hall, Englewood Cliffs (1975)
9. Deboeck, G.J., Kohonen, T.K.: Visual Explorations in Finance: With Self-Organizing Maps. Springer, New York (2010)
10. do Rêgo, R.L.M.E., Araújo, A.F.R., Neto, F.B.L.: Growing self-reconstruction maps. IEEE Trans. Neural Netw. **21**(2), 211–223 (2010)
11. Fortunato, S.: Community detection in graphs. Phys. Rep. **486**, 75–174 (2010)
12. Fu, X., Wang, L.: Data dimensionality reduction with application to simplifying rbf network structure and improving classification performance. IEEE Trans. Syst. Man Cybern., Part B: Cybern. **33**(3), 399–409 (2003)
13. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proc. Natl. Acad. Sci. USA **99**(12), 7821–7826 (2002)
14. Grossberg, S.: Competitive learning: from interactive activation to adaptive resonance. Cogn. Sci. **11**, 23–63 (1987)
15. Hull, J.J.: A database for handwritten text recognition research. IEEE Trans. Pattern Anal. Mach. Intell. **16**, 550–554 (1994)
16. Jain, L.C., Lazzerini, B., Ugur, H.: Innovations in ART Neural Networks (Studies in Fuzziness and Soft Computing). Physica, Heidelberg (2010)
17. Kaylani, A., Georgiopoulos, M., Mollaghasemi, M., Anagnostopoulos, G.C., Sentelle, C., Zhong, M.: An adaptive multiobjective approach to evolving ART architectures. IEEE Trans. Neural Netw. **21**(4), 529–550 (2010)
18. Knuth, D.E.: The Stanford GraphBase: A Platform for Combinatorial Computing. ACM, New York (1993)
19. Kohonen, T.: The self-organizing map. Proc. IEEE **78**(9), 1464–1480 (1990)
20. Kosko, B.: Stochastic competitive learning. IEEE Trans. Neural Netw. **2**(5), 522–529 (1991)
21. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. Phys. Rev. E **78**(4), 046,110(1–5) (2008)
22. Liu, D., Pang, Z., Lloyd, S.R.: A neural network method for detection of obstructive sleep apnea and narcolepsy based on pupil size and EEG. IEEE Trans. Neural Netw. **19**(2), 308–318 (2008)
23. Liu, J., Cai, D., He, X.: Gaussian mixture model with local consistency. In: AAAI'10, vol. 1, pp. 512–517 (2010)
24. López-Rubio, E., de Lazcano-Lobato, J.M.O., López-Rodríguez, D.: Probabilistic PCA self-organizing maps. IEEE Trans. Neural Netw. **20**(9), 1474–1489 (2009)
25. Lu, Z., Ip, H.H.S.: Generalized competitive learning of gaussian mixture models. IEEE Trans. Syst. Man Cybern., Part B: Cybern. **39**(4), 901–909 (2009)

26. Lusseau, D.: The emergent properties of a dolphin social network. Proc. R. Soc. B Biol. Sci. **270**(Suppl 2), S186–S188 (2003)
27. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297. University of California Press, Berkeley (1967)
28. Meyer-Bäse, A., Thümmler, V.: Local and global stability analysis of an unsupervised competitive neural network. IEEE Trans. Neural Netw. **19**(2), 346–351 (2008)
29. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. Phys. Rev. E **69**(6), 066,133 (2004)
30. Newman, M.E.J.: Modularity and community structure in networks. Proc. Natl. Acad. Sci. **103**(23), 8577–8582 (2006)
31. Príncipe, J.C., Miikkulainen, R.: Advances in Self-Organizing Maps - 7th International Workshop, WSOM 2009. Lecture Notes in Computer Science, vol. 5629. Springer, New York (2009)
32. Quiles, M.G., Zhao, L., Alonso, R.L., Romero, R.A.F.: Particle competition for complex network community detection. Chaos **18**(3), 033,107 (2008)
33. Ratle, F., Weston, J., Miller, M.L.: Large-scale clustering through functional embedding. In: Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases - Part II, European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), pp. 266–281. Springer, New York (2008)
34. Shi, J., Malik, J.: Normalized Cut and Image Segmentation. Tech. rep., University of California at Berkeley, Berkeley (1997)
35. Silva, T.C., Zhao, L.: Stochastic competitive learning in complex networks. IEEE Trans. Neural Netw. Learn. Syst. **23**(3), 385–398 (2012)
36. Silva, T.C., Zhao, L.: Uncovering overlapping cluster structures via stochastic competitive learning. Inf. Sci. **247**, 40–61 (2013)
37. Silva, T.C., Zhao, L., Cupertino, T.H.: Handwritten data clustering using agents competition in networks. J. Math. Imaging Vision **45**(3), 264–276 (2013)
38. Sugar, C.A., James, G.M.: Finding the number of clusters in a data set: an information theoretic approach. J. Am. Stat. Assoc. **98**, 750–763 (2003)
39. Tan, A.H., Lu, N., Xiao, D.: Integrating temporal difference methods and self-organizing neural networks for reinforcement learning with delayed evaluative feedback. IEEE Trans. Neural Netw. **19**(2), 230–244 (2008)
40. Wang, Y., Li, C., Zuo, Y.: A selection model for optimal fuzzy clustering algorithm and number of clusters based on competitive comprehensive fuzzy evaluation. IEEE Trans. Fuzzy Syst. **17**(3), 568–577 (2009)
41. Xu, R., II, D.W.: Survey of clustering algorithms. IEEE Trans. Neural Netw. **16**(3), 645–678 (2005)
42. Zachary, W.W.: An information flow model for conflict and fission in small groups. J. Anthropol. Res. **33**, 452–473 (1977)