Journal of Applied Nonlinear Dynamics

https://lhscientificpublishing.com/Journals/JAND-Default.aspx

Advantages of Edge-Centric Collective Dynamics in Machine Learning Tasks

Filipe Alves Neto Verri[1†], Paulo Roberto Urio[2], and Liang Zhao[3]

[1]Institute of Mathematical and Computer Sciences, University of São Paulo, São Carlos, Brazil
School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe AZ, USA
[2]Institute of Mathematical and Computer Sciences, University of São Paulo, São Carlos, Brazil
[3]Ribeirão Preto School of Philosophy, Science and Literature, University of São Paulo, Ribeirão Preto, Brazil

Submission Info

Communicated by Referees
Received DAY MON YEAR
Accepted DAY MON YEAR
Available online DAY MON YEAR

Keywords

Complex networks
Nonlinear dynamics
Collective dynamics
Particle competition
Machine learning

Abstract

We study how effectively edge-centric dynamics solve machine learning tasks. The Edge Domination System is an algorithm to reveal patterns and obtain information of the underlying complex network. The algorithm consists in the simulation of a collective dynamical system based on particle competition for the dominance of edges. In this paper, we propose a vertex-centric version of this model and assess the differences between the edge-centric model. The edge-centric system offers better features in machine learning task, such as greater exploration behavior and faster convergence.

1 Introduction

We study emergent collective behavior in dynamical systems and how it can benefit an algorithm in data learning tasks. The study of systems with the emergence of collective dynamics has led to algorithms based on the behavior of birds, bees, ants, bacterial colony, and many other forms of intelligence. In these biological situations, the driving factor of interactions ultimately aims at the individuals' survival, which manifests in the behavior of cooperation, competition, or a compound of them. Such behaviors often inspire machinery for solutions in optimization and learning problems, and are increasingly found in the literature [1–3].

We introduced in [4] the Edge Domination System (EDS), an algorithm that deterministically obtains information of the underlying complex network by exploring patterns revealed by the collective dynamics of particle competition. The algorithm models particles randomly walking a complex network, represented by a graph, in which teams of particles dominate the edges they visited more frequently than their rivals. Afterward, the algorithm promotes an *unfolding*: it breaks the network down in subnetworks

†Corresponding author.
Email address: filipeneto@usp.br

ISSN 2164-6457, eISSN 2164-6473/$-see front materials
DOI:10.5890/JAND.2016.01.001

induced by the edges and their dominant label. The unfolded networks allow the algorithm to finally draw conclusions regarding the labels of the vertices, which is the result for the learning task.

Unlike similar approaches with particle competition [5], the edges are the resource the particles compete for instead of the vertices. Since this approach is rather new, the advantages of edge competition is still unclear and need to be explored. We presented an initial assessment in [6], and in this paper, we will further explore the differences between the two resources of competition. We will model a comparable algorithm of particles competing for vertices, and we will compare it with EDS, whose particles compete for edges.

We conclude that the edge-centric dynamics is preferred over the vertex-centric, since it acquires more information and converges faster at the same computational cost.

The remaining of this paper is organized as follows. Section 2 reviews the original EDS and introduces an artificial network model to represent learning problems. The effectiveness of a simplified version of EDS and its vertex-based counterpart are compared in section 3. The convergence of the classification results in both systems is studied in section 4. Finally, section 5 concludes the paper.

2 Methods

In this section, we review the original Edge Domination System and propose a artificial network model. We use this network model to represent the learning scenarios in which we compare EDS and its vertex-based counterpart. Specifically, it generates binary classification tasks with controllable difficulty.

2.1 Review of Edge Domination System

The Edge Domination System [4] models a process in a complex network in which teams of particles compete to dominate the largest number of edges. The interaction network is represented by a graph $G = (\mathscr{V}, \mathscr{E})$ that is simple, unweighted, and undirected. The set of vertices $\mathscr{V} = \mathscr{L} \cup \mathscr{U}$ splits vertices in l labeled vertices, $\mathscr{L} = \{v_1, \ldots, v_l\}$, and u unlabeled vertices, $\mathscr{U} = \{v_{l+1}, \ldots, v_{l+u}\}$, with a total of $|\mathscr{V}| = l + u$ vertices, where we suppose few labeled vertices, $l \ll u$. The problem contains $C > 1$ possible labels, and each v_i from the labeled set $\mathscr{L}$ has the label $y_i \in \{1, \ldots, C\}$. The set of edges $\mathscr{E} \subseteq \mathscr{V} \times \mathscr{V}$ contains an edge (i, j) if v_i and v_j are connected. The network is also represented by the adjacency matrix $A = (a_{ij})$ where $a_{ij} = a_{ji} = 1$ if the edge (i, j) exists.

One team of particles exists for each possible label, totalizing C teams of particles and of vertices. All the particles randomly walk from vertex to vertex, one vertex at a time. When any particle passes through an edge, the edge counts the visit for the particle's team. An edge registers two values for a team: the number of visits solely from the current time, and the cumulative number of visits since the beginning. At any moment, the team that dominates an edge is determined by the team with the highest number of visits. The system models no cooperation between teams, and any two particles or vertices are rival if they belong to different teams. Particles are always bonded to the same team, but the size of the teams vary because particles can be removed from the system or new particles can be generated by the labeled vertices. These rules are described below as three actions, which combine to model the evolution rule of the system.

Walking. A particle is always in a vertex and chooses a connected vertex to move at each time, with equal probability between the neighbors of the current vertex.

Absorption. The transitory *team domination* is the number of visits in an edge by team that occurred at the last time. In the next movement, the transitory team domination determines the success rate of particles moving to their next vertices, or to fail and being removed from the system—the higher the presence of a team, the easier particles from the same team pass through the edge. Additionally, all the labeled vertices thwart every rival particle trying to reach them, automatically removing those

particles from the system—this action characterizes such vertices as *sinks*.

Generation. Besides being sinks, the labeled vertices are also *sources* for new particles. A source vertex generates particles for its team according to the neighborhood size and the size of the population. This rule maintains the size of the population close to its initial size.

Every edge has transitory and cumulative values of team domination, which assist an algorithm solving machine learning problems. An approach considers only the team that dominates each edge, grouping the edges by the team that dominates. The algorithm unfolds the interaction network in subnetworks by group: the *unfolding* of team c at time t is the subnetwork containing all the edges dominated by team c at time t. Over the evolution of the competition process, the network's unfoldings change less and less until they converge to stable subnetworks that are afterward analyzed to the machine learning task.

In [4], we proposed a system that describes the expected evolution of the particles over time. This system models deterministic evolution rules, and it is denoted as

$$X(t) = \begin{bmatrix} \boldsymbol{n}^c(t) = \left[n_i^c(t)\right]_{i=1,\ldots,|\mathscr{V}|} \\ N^c(t) = \left(n_{ij}^c(t)\right)_{i,j=1,\ldots,|\mathscr{V}|} \\ \Delta^c(t) = \left(\delta_{ij}^c(t)\right)_{i,j=1,\ldots,|\mathscr{V}|} \end{bmatrix}_{c=1,\ldots,C}, \tag{1}$$

where $\boldsymbol{n}^c(t)$ is a row vector whose elements $n_i^c(t)$ describe the population of particles from team c in each vertex v_i at time t. Stored in sparse matrices, the elements $n_{ij}^c(t) \in N^c(t)$ and $\delta_{ij}^c(t) \in \Delta^c(t)$ are the *transitory directed domination* and the *cumulative domination*, in that order. At time t, $n_{ij}^c(t)$ is the portion of particles from team c that moved from v_i to v_j, while $\delta_{ij}^c(t)$ is the accumulated portion of visits since the beginning.

The system X is a nonlinear Markovian dynamical system with the deterministic evolution function

$$\phi\colon \begin{cases} \boldsymbol{n}^c(t+1) = \boldsymbol{n}^c(t) \times P^c(X(t)) + \boldsymbol{g}^c(X(t)) \\ N^c(t+1) = \operatorname{diag} \boldsymbol{n}^c(t) \times P^c(X(t)) \\ \Delta^c(t+1) = \Delta^c(t) + N^c(t+1) \end{cases}, \tag{2}$$

where $\operatorname{diag} \boldsymbol{v}$ is a square matrix with the elements of vector $\boldsymbol{v}$ on the main diagonal and $\times$ stands for the vector–matrix product.

The function $P^c(X(t))$ of the system X at time t is a square matrix whose elements are

$$p_{ij}^c(X(t)) = \begin{cases} 0 & \text{if } v_j \in \mathscr{L} \text{ and } y_j \neq c, \\ \dfrac{a_{ij}}{\deg v_i}\left(1 - \lambda\, \sigma_{ij}^c(X(t))\right) & \text{otherwise,} \end{cases} \tag{3}$$

where

$$\sigma_{ij}^c(X(t)) = \begin{cases} 1 - \dfrac{n_{ij}^c(t) + n_{ji}^c(t)}{\sum_{q=1}^{C} n_{ij}^q(t) + n_{ji}^q(t)} & \text{if } \sum_{q=1}^{C} n_{ij}^q(t) + n_{ji}^q(t) > 0, \\ 1 - \dfrac{1}{C} & \text{otherwise.} \end{cases} \tag{4}$$

Given that we know the initial state $X(0)$ of the system, the function $\boldsymbol{g}^c(X(t))$ of the system X at time t returns a row vector where the i-th element is

$$g_i^c(X(t)) = \rho_i^c \max\{0,\ \mathbf{1} \cdot \boldsymbol{n}^c(0) - \mathbf{1} \cdot \boldsymbol{n}^c(t)\}, \tag{5}$$

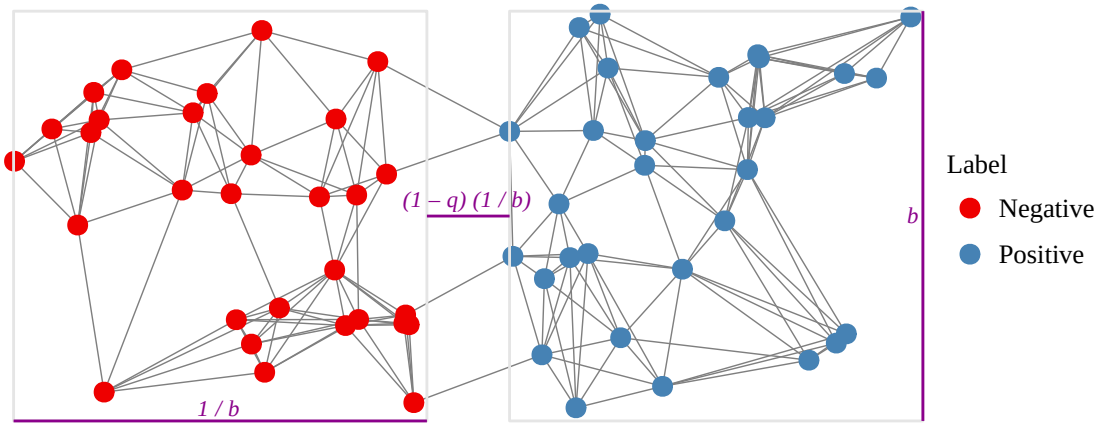

Fig. 1 Example of network $G(60,6,0.8,1)$. Vertices are colored according with their label and positioned according with their associated point in the space.

where $\mathbf{1}$ is a row vector whose elements are 1, '$\cdot$' is the inner product between vectors, and

$$\rho_i^c = \begin{cases} \dfrac{\deg v_i}{\sum_{v_j \in \mathscr{G}^c} \deg v_j} & \text{if } v_i \in \mathscr{G}^c, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Using the cumulative domination, we group edges by team domination. For each team c, the subset of edges $\mathscr{E}^c(t) \subseteq \mathscr{E}$ contains the edges dominated by the team c,

$$\mathscr{E}^c(t) = \left\{ (i,j) \middle| \arg\max_q \left(\delta_{ij}^q(t) + \delta_{ji}^q(t) \right) = c \right\}. \tag{7}$$

We define the subnetwork

$$G^c(t) = (\mathscr{V}, \mathscr{E}^c(t)) \tag{8}$$

as the *unfolding* of network G obtained with the edges of team c at time t.

The initial state of the system X is determined by an arbitrary *population size* $n_i^c(0)$ of initial active particles and

$$\begin{cases} n_{ij}^c(0) = 0, \\ \delta_{ij}^c(0) = 0. \end{cases} \tag{9}$$

2.2 Artificial network model

To study the EDS, we introduce a random network with labeled vertices. The network model represents a binary classification task with controllable overlap between instances with different labels.

The undirected random network $G(n,k,q,b)$ has n vertices of which half of them are positive-labeled. The other half has negative label. Each vertex v_i is associated with a point x_i in a bi-dimensional space. Let $y_i \in \{+,-\}$ be the label of vertex v_i, the points x_i such that $y_i = -$ is uniformly distributed in the interval $\left[0, b^{-1}\right] \times [0,b]$, and the points x_j such that $y_j = +$ is in the interval $\left[(2-q)b^{-1}, (3-q)b^{-1}\right] \times [0,b]$. The two vertices v_i and v_j are connected if x_i is one of the k-neighborhood of x_j according to the Euclidean metric, or vice-versa.

The parameter $q \in [0,2]$ controls the overlapping area between the surfaces from which the points are sampled. If $q < 1$, the surfaces do not overlap and the inter-label region of the resulting network is less dense than the intra-label. If $q = 1$, the density of the resulting graph is uniform along the graph. If $q > 1$, the surfaces overlap and there are high density of edges in the overlapping region. In the extreme

$q = 2$, points with different labels are sampled from the same surface. The parameter $b > 0$ controls the size of the border, increasing the number of vertices that may have at least one connection with vertices with other label.

The networks $G(n,k,q,b)$ are, thus, spatial networks with controllable border and overlap between vertices with different labels. The *border size* is b and the *overlapping area* is $\max(0,q-1)$.

3 Effectiveness in learning problems

We assess the differences between the edge-based competition system and its vertex-based counterpart. To simplify our studies, we slightly modify the original EDS under certain assumptions that should not impair the performance in machine learning problems. We compare both models in the synthetic networks that provide controllable learning scenarios, see section 2.2.

3.1 Simplified systems

To simplify the EDS, we assume two properties: the system's unfoldings always converges, and the initial population is balanced between vertices with different labels.

Assumption of unfolding convergence The original model considers the domination accumulated over time to unfold the network. After few iterations, we observed the current domination becomes proportional to the accumulated value, leading us to simplify the model. We dropped the accumulated domination strategy to only observe the current domination.

In fact, if the matrices $N^c(t)$ converge for all c,

$$\lim_{t\to\infty} \arg\max_c \left(\delta_{ij}^c(t) + \delta_{ji}^c(t)\right) = \lim_{t\to\infty} \arg\max_c \left(n_{ij}^c(t) + n_{ji}^c(t)\right).$$

We can obtain the same limiting unfoldings without the accumulated domination. The convergence of the unfoldings are analytically and experimentally studied in section 4. (We have verified that the unfoldings of all simulations are convergent in this paper.)

Assumption of no preferred label In Verri, Urio, and Zhao [4], the semi-supervised learning method based on the EDS uses an initial population of particles distributed by the vertex degree, without considering the particles' team. Conversely, we adopt a less restrictive assumption of the initial condition. We assume there is no preferred label, that is, given $n_0 > 0$,

$$\sum_i n_i^c(0) = n_0, \tag{10}$$

for all labels c.

With this assumption, and relying in the scale invariant property of the system, we simplify the particle generation term, Eq. (5). The scale invariant property means that multiplying the initial population of particles in every vertex by a positive constant scales the population at any time by the same factor. Because we need to know which team dominates and not the size of each team, we may scale any initial population to be a discrete distribution of particles and maintain the same resulting unfoldings. To do so, we multiply the initial population of each vertex by n_0^{-1}. The summation over a distribution is always 1, so is the total population of the teams at the beginning.

Under such assumptions, we prove the size of every team never exceeds 1. The population limit enables us to drop the max function in Eq. (5).

Theorem 1. *Population limit. The total population* $\sum_i n_i^c(t)$ *of every label* c *is less than or equal to* 1

if the initial population is $\sum_i n_i^c(0) = 1$.

Proof. We know that

$$n_i^c(1) = \sum_j n_j^c(0) p_{ji}^c(X(0)) \leq \sum_j n_j^c(0) \frac{a_{ji}}{\deg v_j} \left(1 - \lambda \frac{C-1}{C}\right)$$

and thus

$$\sum_i n_i^c(1) \leq \left(1 - \lambda \frac{C-1}{C}\right) \sum_j n_j^c(0) \sum_i \frac{a_{ji}}{\deg v_j} \leq \sum_j n_j^c(0) = 1.$$

Now, we assume $\sum_i n_i^c(t) \leq 1$ for all c and $t > 1$, and prove the inequality for $t+1$.

$$\begin{aligned} n_i^c(t+1) = \sum_j n_j^c(t) p_{ji}^c(X(t)) + g_i^c(X(t)) \leq \\ \sum_j n_j^c(t) \frac{a_{ji}}{\deg v_j} \left(1 - \lambda \sigma_{ji}^c(X(t))\right) + \rho_i^c \max\{0,\, \mathbf{1} \cdot \boldsymbol{n}^c(0) - \mathbf{1} \cdot \boldsymbol{n}^c(t)\} \leq \\ \sum_j n_j^c(t) \frac{a_{ji}}{\deg v_j} \left(1 - \lambda \sigma_{ji}^c(X(t))\right) + \rho_i^c \max\{0,\, 1 - \mathbf{1} \cdot \boldsymbol{n}^c(t)\} \leq \\ \sum_j n_j^c(t) \frac{a_{ji}}{\deg v_j} \left(1 - \lambda \sigma_{ji}^c(X(t))\right) + \rho_i^c \left(1 - \sum_k n_k^c(t)\right) \end{aligned} \tag{11}$$

and thus

$$\begin{aligned} \sum_i n_i^c(t+1) \leq \sum_i \sum_j n_j^c(t) \frac{a_{ji}}{\deg v_j} \left(1 - \lambda \sigma_{ji}^c(X(t))\right) + \left(1 - \sum_k n_k^c(t)\right) \sum_i \rho_i^c \leq \\ \sum_j n_j^c(t) \sum_i \frac{a_{ji}}{\deg v_j} \left(1 - \lambda \sigma_{ji}^c(X(t))\right) + 1 - \sum_k n_k^c(t) \leq \sum_j n_j^c(t) + 1 - \sum_k n_k^c(t) \leq 1. \end{aligned} \tag{12}$$

By mathematical induction, theorem 1 is proved for all time t and team c. □

3.1.1 Simplified Edge Domination System

From the assumptions, the simplified edge-based system is

$$X(t) = \begin{bmatrix} \boldsymbol{n}^c(t) = \left[n_i^c(t)\right]_{i=1,\ldots,|\mathcal{V}|} \\ N^c(t) = \left(n_{ij}^c(t)\right)_{i,j=1,\ldots,|\mathcal{V}|} \end{bmatrix}_{c=1,\ldots,C}, \tag{13}$$

with evolution

$$\phi: \begin{cases} \boldsymbol{n}^c(t+1) = \boldsymbol{n}^c(t) \times P^c(X(t)) + \boldsymbol{g}^c(X(t)) \\ N^c(t+1) = \operatorname{diag} \boldsymbol{n}^c(t) \times P^c(X(t)), \end{cases} \tag{14}$$

such that $P^c(X(t))$ is Eq. (3) and $\boldsymbol{g}^c(X(t))$ is a row vector where the i-th element is

$$g_i^c(X(t)) = \rho_i^c \left(1 - \mathbf{1} \cdot \boldsymbol{n}^c(t)\right). \tag{15}$$

The initial state is given by an arbitrary *discrete distribution* $\boldsymbol{n}^c(0)$, and the zero matrix $N^c(0)$. The unfoldings are now based on the current domination instead of the cumulative. Thus, the unfoldings

are $G^c(t) = (\mathscr{V}, \mathscr{E}^c(t))$ such that

$$\mathscr{E}^c(t) = \left\{ (i,j) \middle| \arg\max_q \left(n_{ij}^q(t) + n_{ji}^q(t) \right) = c \right\}. \tag{16}$$

3.1.2 Simplified Vertex Domination System

From system in Eq. (13), it is straightforward to shift the competition to the vertices. We eliminate the states that track the current domination in the edges and rewrite $\boldsymbol{\sigma}$, Eq. (4), in terms of the current domination in the vertices. The Vertex Domination System (VDS) is

$$X(t) = \left[\boldsymbol{n}^c(t) = \left[n_i^c(t) \right]_{i=1,\dots,|\mathscr{V}|} \right]_{c=1,\dots,C}, \tag{17}$$

with evolution function

$$\boldsymbol{n}^c(t+1) = \boldsymbol{n}^c(t) \times P^c(X(t)) + \boldsymbol{g}^c(X(t)), \tag{18}$$

where $P^c(X(t))$ is Eq. (3) with

$$\sigma_{ij}^c(X(t)) = \begin{cases} 1 - \dfrac{n_j^c(t)}{\sum_{q=1}^C n_j^q(t)} & \text{if } \sum_{q=1}^C n_j^q > 0, \\ 1 - \dfrac{1}{C} & \text{otherwise,} \end{cases} \tag{19}$$

and $\boldsymbol{g}^c(X(t))$ is a row vector where the i-th element is Eq. (15).

The initial state of the system X is given by an arbitrary *discrete distribution* $\boldsymbol{n}^c(0)$. The concept of unfoldings is not defined in this system.

3.2 Experimental results

To compare the systems Edge Domination System (EDS) and Vertex Domination System (VDS), we formulate a simple example where the systems behave quite differently. The toy example motivates us to explore the reasons of the differences and the impact of them in machine learning tasks. We further compare the effectiveness of each system in artificial data.

3.2.1 Toy example

Consider the network whose vertices are points in a two-dimensional plane depicted in Fig. 2(a). The vertices of this network can be either positive (blue, light, shaping a vertical pattern) or negative (red, dark, shaping a horizontal pattern). The central vertex (grey, light) has no label. Four vertices (square-shaped) are labeled. We connect each vertex to its nearest neighbor. Using this network as the input for EDS and VDS, one shall expect the fraction of positive and negative particles to be the same in the central vertex for both systems. Figure 3(a) demonstrates this behavior.

Now, consider adding a new vertex to this dataset as shown in Fig. 2(b). Notice that the modification barely changes the network's topology. However, in this network both systems behave differently, see Fig. 3(b). In the edge-centric system, the proportion of different kinds of particles in the central vertex alternates about the same. In the vertex-centric system, on the other hand, the central vertex is dominated by positive-labeled particles. This example indicates the two models are not equivalent, and it motivates us to investigate the advantages of the edge-centric model against the vertex-centric one.

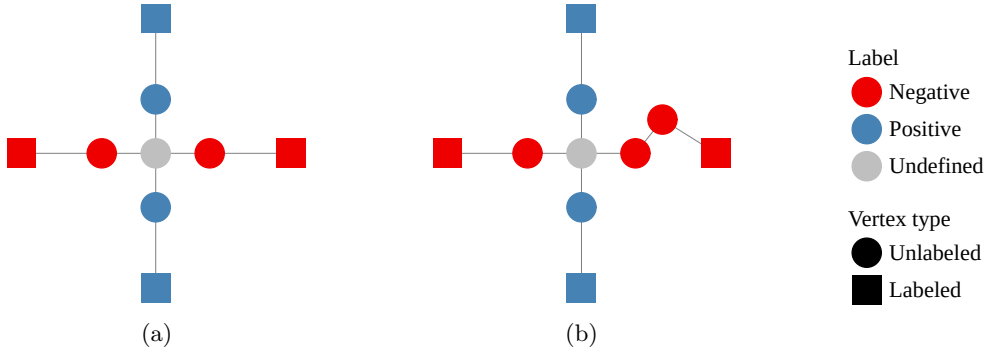

Fig. 2 Toy networks with (a) 9 vertices and (b) 10 vertices.

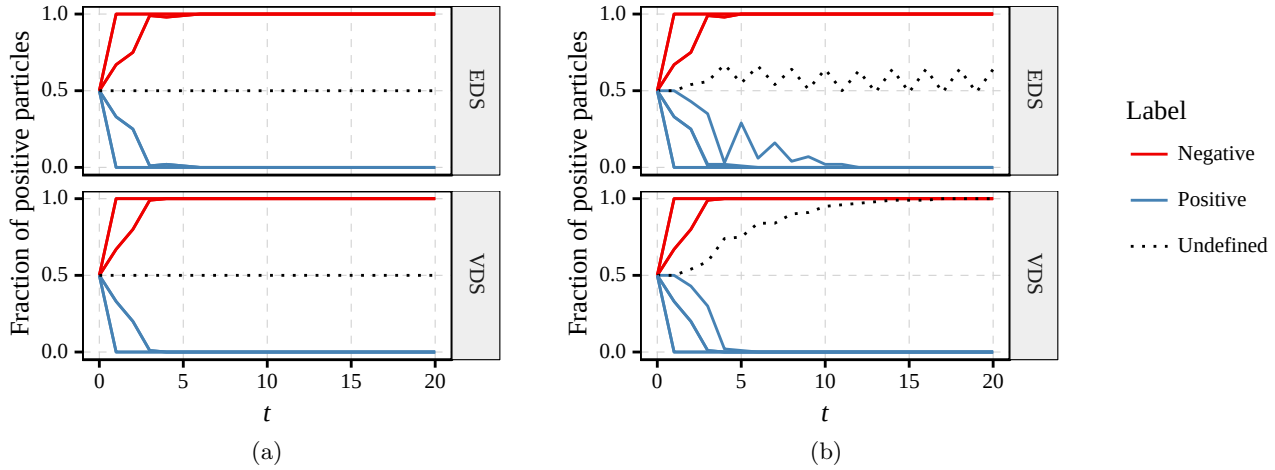

Fig. 3 Fraction of positive particles in each vertex over time with (a) 9-vertices and (b) 10-vertices networks as input.

3.2.2 Classification error

We compare the classification error of the systems for increasing levels of difficulty. The error is obtained from 30 independently sampled networks $G(250,k,q,b)$ for each $k \in \{6,8,\ldots,12\}$, $q \in \{0.8,0.9,\ldots,1.4\}$, and $b \in \{1,2,3\}$. We fixed $\lambda = 1$ and the fraction of initially labeled vertices in 10%.

We start both systems with population size

$$n_i^c(0) = \frac{\deg v_i}{2\,|\mathscr{E}|}$$

independently of label c. To classify an unlabeled vertex v_i, we adopt the original strategy in Verri, Urio, and Zhao [4] for the EDS at time $\tau = 100$, $y_i = \arg\max_c |\mathscr{E}(\mathscr{N}_i^c(\tau))|$, where $|\mathscr{E}(\mathscr{N}_i^c(\tau))|$ stands for the number of edges in the neighborhood of v_i in $G^c(\tau)$. In VDS, we just consider the dominating label of the vertex at time $\tau = 100$, that is, $y_i = \arg\max_c n_i^c(\tau)$.

The classification results varying the network G is shown in Fig. 4. Consistently, as q and b increases, the problem becomes harder, and the systems obtain higher error rates. Besides both systems having similar results, EDS slightly outperforms VDS when the overlapping area is large, $b = 3$ and $q > 1$. VDS, on the other hand, performs better when there is no overlap, $q < 1$. We also note that the EDS benefit most from the increasing number of edges.

The lower error rate of EDS in problems with large overlap supports the hypothesis that this system

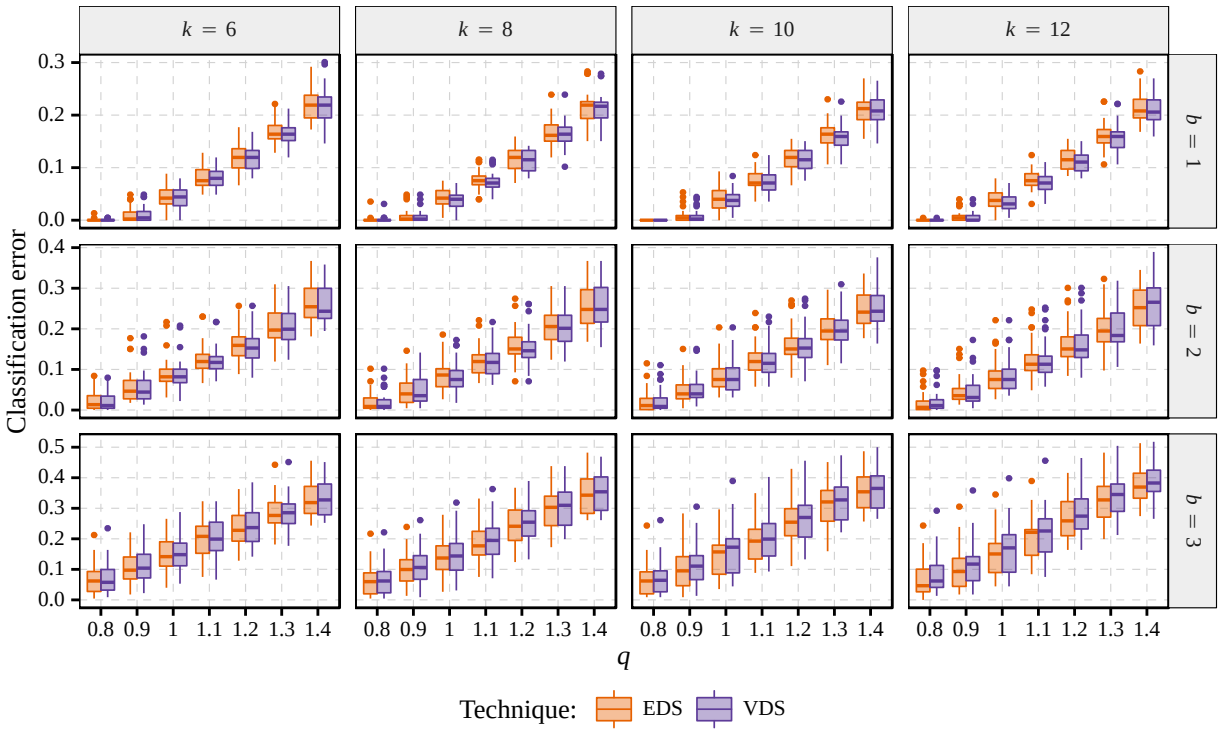

Fig. 4 Classification error of both systems in different networks $G(250,k,q,b)$ for 10% of initially labeled vertices.

can obtain more information of the border than the VDS. In the next subsection, we address this issue.

3.2.3 Overlap prediction error

In [4], EDS shows the ability of recognize data items of different labels sharing the same attribute space. We question whether VDS has this capacity. We use the fraction of positive-labeled particles in each vertex, similarly to how we have done in section 3.2.1, to predict the fraction of positive-labeled neighbors of each vertex.

Let $\mathscr{P}_i$ be the set of positive neighbors of the vertex v_i, we use the root-mean-square error

$$\mathrm{RMSE}\left(\left[\frac{n_i^+(\tau=100)}{n_i^+(\tau=100)+n_i^-(\tau=100)}\right]_i, \left[\frac{|\mathscr{P}_i|}{\deg v_i}\right]_i\right),\ i \in \mathscr{V}\setminus\mathscr{L}$$

to assess the effectiveness of the overlap prediction. The error is obtained from the same networks $G(250,k,q,b)$ and initially labeled vertices as the experiment in section 3.2.2.

Fig. 5 summarizes the results. Unlike the results of the classification error, EDS consistently outperforms VDS in all cases. Such result demonstrates that EDS achieve more information than VDS, but the classification mechanism proposed in [4] fails in taking advantage of the extra information.

We also show that a vertex tends to be totally dominated by either positive or negative particles in VDS. To study this property, we define the entropy $h_i(t)$ of vertex v_i at time t as

$$h_i(t) = -\sum_{c\in\{+,-\}} \frac{n_i^c(t)}{n_i^+(t)+n_i^-(t)} \log_2\left(\frac{n_i^c(t)}{n_i^+(t)+n_i^-(t)}\right). \tag{20}$$

On one extreme, entropy $h_i(t)=0$ means that the vertex v_i has the same population of positive and negative particles at time t. On the other extreme, entropy $h_i(t)=1$ indicates that only one kind of particles is present in vertex v_i at time t.

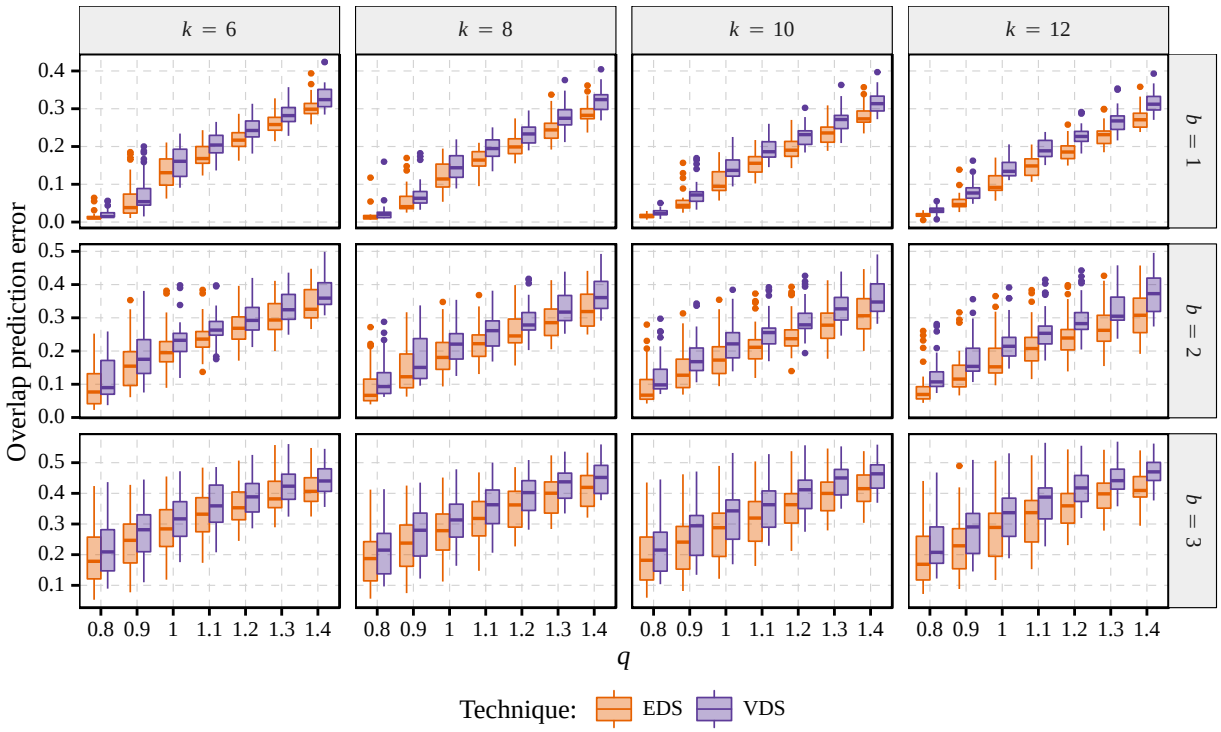

Fig. 5 Root-mean-square error of the overlap prediction for both systems in different networks $G(250,k,q,b)$ for 10% of initially labeled vertices.

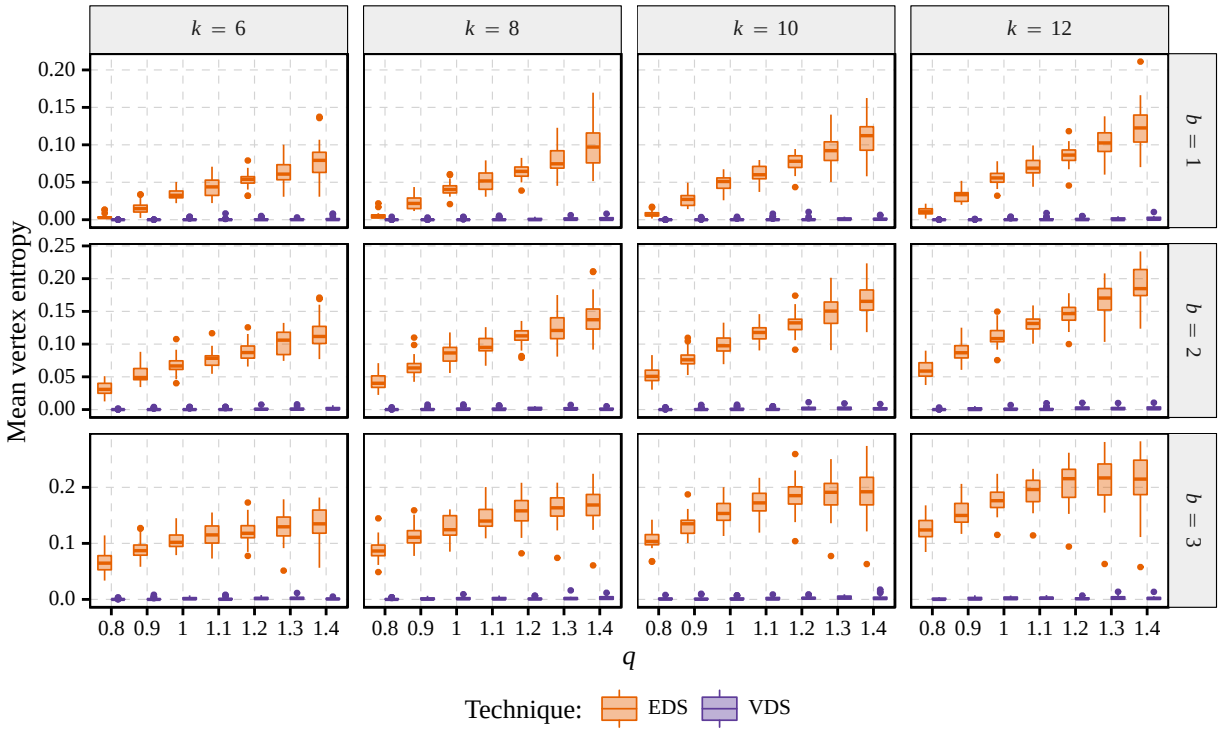

Fig. 6 Mean vertex entropy in both systems at time $\tau = 100$ for different networks $G(250,k,q,b)$ with 10% of initially labeled vertices.

The measurements of the average entropy at time $\tau = 100$ in the same networks as the previous experiment are shown in Fig. 6. We verify that the entropy of the vertices in the VDS are extremely low. However, in EDS, the entropy increases as the overlapping area becomes larger. We imply that this increasing corresponds to the detection of the border. High entropy also suggests that particles propagate farther, reaching larger regions dominated by rival particles. As a result, a greater exploration behavior emerges, and particles are more likely to learn from the network's topology.

4 Convergence of the classification results

In the previous experiments, we assess the effectiveness of each system in learning problems. Classification accuracy, however, is not the only important feature in machine learning. A high time complexity order may restrict the application of a learning algorithm.

Both VDS and EDS have linear time complexity regarding the number of vertices in the input network. However, one may be interested in the convergent state of the systems. In such case, the complexity order might be much higher if the convergence time is not sublinear regarding the input. In this section, we address the classification result convergence in both system with maximum competition ($\lambda = 1$).

4.1 Further simplification

Both systems are further simplified fixing the competition parameter in $\lambda = 1$. Considering two labels, $+$ and $-$, we rewrite both system to contain a variable explicitly related to the classification output. We restrict the initial conditions of both systems to eliminate the conditional terms in Eq. (3) and Eq. (4). Two requirements over the network are also necessary: every vertex should be connected with at least one non-labeled vertex, and two labeled vertices may be connected only if they share the same label. Mathematically, it is

$$\begin{cases} \sum\limits_{j\,|\,v_j \in \mathscr{V}\setminus\mathscr{L}} a_{ij} > 0, & \text{for all } i \in \{1,\ldots,|\mathscr{V}|\}, \text{ and} \\ a_{ij} = 0 & \text{if } v_i, v_j \in \mathscr{L} \text{ and } y_i \neq y_j. \end{cases} \tag{21}$$

4.1.1 Further simplified Edge Domination System

In EDS, the classification result depends on the unfoldings. These are convergent if

$$\lim_{t\to\infty} \arg\max_q \left(n_{ij}^q(t) + n_{ji}^q(t) \right) = c$$

for $c = +$ or $c = -$. To further simplify EDS, we substitute the nonzero entries of the current domination matrix $N^c(t)$ for a set of variables $s_{ij}(t)$ that represents the probability of a positive-labeled particle surviving at time t given it has decided to move from vertex v_i to vertex v_j. So, let

$$s_{ij}(t) = \frac{n_{ij}^+(t) + n_{ji}^+(t)}{n_{ij}^+(t) + n_{ji}^+(t) + n_{ij}^-(t) + n_{ji}^-(t)} \tag{22}$$

and imposing the restrictions

$$\begin{cases} n_i^c(0) = 0 & \text{if } v_i \in \mathscr{L} \text{ and } y_i \neq c, \\ n_i^c(0) > 0 & \text{otherwise,} \end{cases} \tag{23}$$

$$\sum_i n_i^c(0) = 1, \text{ and}$$

$$n_{ij}^c(0) = \begin{cases} 0 & \text{if } v_i \in \mathscr{L} \text{ and } y_i \neq c, \\ 0 & \text{if } v_j \in \mathscr{L} \text{ and } y_j \neq c, \\ 1 & \text{otherwise,} \end{cases} \tag{24}$$

we rewrite the evolution function of EDS as

$$\phi : \begin{cases} n_i^+(t+1) = \rho_i^+ \left(1 - \sum_j n_j^+(t)\right) + \sum_j n_j^+(t) \frac{a_{ji}}{\deg v_j} s_{ji}(t) \\ n_i^-(t+1) = \rho_i^- \left(1 - \sum_j n_j^-(t)\right) + \sum_j n_j^-(t) \frac{a_{ji}}{\deg v_j} [1 - s_{ji}(t)] \\ s_{ij}(t+1) = \dfrac{\left(n_i^+(t)\frac{a_{ij}}{\deg v_i} + n_j^+(t)\frac{a_{ji}}{\deg v_j}\right) s_{ij}(t)}{\left(n_i^+(t)\frac{a_{ij}}{\deg v_i} + n_j^+(t)\frac{a_{ji}}{\deg v_j}\right) s_{ij}(t) + \left(n_i^-(t)\frac{a_{ij}}{\deg v_i} + n_j^-(t)\frac{a_{ji}}{\deg v_j}\right) [1 - s_{ij}(t)]} \end{cases}, \tag{25}$$

with initial conditions

$$s_{ij}(0) = \begin{cases} 0 & \text{if } v_i \in \mathscr{L} \text{ and } y_i = -, \\ 0 & \text{if } v_j \in \mathscr{L} \text{ and } y_j = -, \\ 1 & \text{if } v_i \in \mathscr{L} \text{ and } y_i = +, \\ 1 & \text{if } v_j \in \mathscr{L} \text{ and } y_j = +, \\ 0.5 & \text{otherwise.} \end{cases} \tag{26}$$

The unfoldings $G^c(t) = (\mathscr{V}, \mathscr{E}^c(t))$ are also rewritten in terms of s_{ij}, that is,

$$\mathscr{E}^+(t) = \left\{(i,j) \middle| s_{ij}(t) + s_{ji}(t) > 1\right\}, \text{ and } \mathscr{E}^-(t) = \left\{(i,j) \middle| s_{ij}(t) + s_{ji}(t) < 1\right\}. \tag{27}$$

To find out the equations for $s_{ij}(t+1)$, $n_{ij}^+(t+1)$, and $n_{ij}^-(t+1)$, we first simplify Eq. (3) under the previous assumptions.

Lemma 1. *Assuming $\lambda = 1$, the network requirements in Eq.* (21) *and the initial conditions in Eqs.* (23) *and* (24)*,*

$$p_{ij}^c(X(t)) = \frac{a_{ij}}{\deg v_i} \frac{n_{ij}^c(t) + n_{ji}^c(t)}{n_{ij}^+(t) + n_{ji}^+(t) + n_{ij}^-(t) + n_{ji}^-(t)},$$

for all time t and labels $c \in \{+,-\}$ in EDS.

Proof. Let $\mathsf{sink}_c v_i$ be a logical function that yields true if v_i is a sink for particles of team c. We observe the evolution of $n_i^c(t)$, $n_{ji}^c(t)$, and $n_{ij}^c(t)$ in vertices v_i such that $\mathsf{sink}_c v_i$ holds true. From Eqs. (23) and (24), $n_i^c(0) = n_{ij}^c(0) = 0$, for all j. For any $t \geq 0$,

$$n_i^c(t+1) = \sum_j n_j^c(t) p_{ji}^c(t) + \rho_i^c \left(1 - \sum_j n_j^c(t)\right) = 0,$$

$$n_{ji}^c(t+1) = n_j^c(t) p_{ji}^c(t) = 0,$$

since $p_{ji}^c(t) = \rho_i^c = 0$ if v_i is a sink for particle of team c, see Eq. (3). Once $n_i^c(t) = 0$ for any time t,

$$n_{ij}^c(t+1) = n_i^c(t) p_{ij}^c(t) = 0,$$

as well.

For the other vertices v_i and v_j, such that $\mathsf{sink}_c v_i$ and $\mathsf{sink}_c v_j$ holds false, we show that $n_i^c(t), n_{ji}^c(t) > 0$ by mathematical induction. Eqs. (23) and (24) guarantee these inequalities for $t = 0$. Assuming they hold true for t, we extend to $t+1$:

$$n_i^c(t+1) = \sum_j n_j^c(t) p_{ji}^c(t) + \rho_i^c \left(1 - \sum_j n_j^c(t)\right) \geq \sum_j n_j^c(t) p_{ji}^c(t) = \sum_j n_j^c(t) \frac{a_{ji}}{\deg v_j} \left(1 - \sigma_{ji}^c(X(t))\right) = \sum_{j \mid \neg \mathsf{sink}_c v_j} n_j^c(t) \frac{a_{ji}}{\deg v_j} \left(1 - \sigma_{ji}^c(X(t))\right) > 0,$$

$$n_{ji}^c(t+1) = n_j^c(t) p_{ji}^c(t) = n_j^c(t) \frac{a_{ji}}{\deg v_j} \left(1 - \sigma_{ji}^c(X(t))\right) > 0,$$

since

$$\sigma_{ji}^c(X(t)) = 1 - \frac{n_{ij}^c(t) + n_{ji}^c(t)}{n_{ij}^+(t) + n_{ji}^+(t) + n_{ij}^-(t) + n_{ji}^-(t)} < 1,$$

for all i, j such that $\neg \mathsf{sink}_c v_i$ and $\neg \mathsf{sink}_c v_j$, and every vertex has a non-labeled neighbor.

From these results, $n_{ij}^+(t) + n_{ji}^+(t) + n_{ij}^-(t) + n_{ji}^-(t) > 0$, as a vertex cannot be a source for both positive and negative particles. Thus, we drop the conditional in Eq. (4),

$$p_{ij}^c(X(t)) = \begin{cases} 0 & \text{if } \mathsf{sink}_c v_j, \\ \dfrac{a_{ij}}{\deg v_i} \dfrac{n_{ij}^c(t) + n_{ji}^c(t)}{n_{ij}^+(t) + n_{ji}^+(t) + n_{ij}^-(t) + n_{ji}^-(t)} & \text{otherwise.} \end{cases}$$

However, $n_{ij}^c(t) + n_{ji}^c(t) = 0$ if $\mathsf{sink}_c v_j$, eliminating the need of the conditional. □

From lemma 1, it is straightforward derive the simplified system. The restriction of two rival sinks not being connected guarantees the coherence of the initial state.

4.1.2 Further simplified Vertex Domination System

Analogously, we introduce to VDS a set of variables $s_i(t)$ that is the probability of a positive-labeled particle surviving at time t given it has decided to move to vertex v_i independently of its origin. Thus, let

$$s_i(t) = \frac{n_i^+(t)}{n_i^+(t) + n_i^-(t)}$$

and considering the restriction in Eq. (23), the VDS is rewritten as

$$\begin{cases} n_i^+(t+1) = \rho_i^+ \left(1 - \sum_j n_j^+(t)\right) + s_i(t) \sum_j n_j^+(t) \dfrac{a_{ji}}{\deg v_j} \\ n_i^-(t+1) = \rho_i^- \left(1 - \sum_j n_j^-(t)\right) + [1 - s_i(t)] \sum_j n_j^-(t) \dfrac{a_{ji}}{\deg v_j} \\ s_i(t+1) = \dfrac{s_i(t) \sum_j n_j^+(t) \frac{a_{ji}}{\deg v_j}}{s_i(t) \sum_j n_j^+(t) \frac{a_{ji}}{\deg v_j} + [1 - s_i(t)] \sum_j n_j^-(t) \frac{a_{ji}}{\deg v_j}} \end{cases}, \tag{28}$$

with initial conditions

$$s_i(0) = \begin{cases} 0 & \text{if } v_i \in \mathscr{L} \text{ and } y_i = -, \\ 1 & \text{if } v_i \in \mathscr{L} \text{ and } y_i = +, \\ 0.5 & \text{otherwise.} \end{cases} \tag{29}$$

The classification mechanism adopted in this paper for VDS becomes

$$y_i = \begin{cases} + & \text{if } s_i(t) > 0.5, \\ - & \text{otherwise.} \end{cases} \tag{30}$$

The proof is analogous to the EDS simplification.

4.2 Analytical results

The introduced variables s_{ij} and s_i are directed related to the classification results of the EDS and VDS, respectively. Thus, their convergence imply the convergence of the classification results.

In EDS, we show that once an edge is completely dominated by particles of the same team, its domination level will not change anymore. Let $s_{ij}(t+1) = s_{ij}(t) = s_{ij}$ at some time t and some i, j. Thus,

$$s_{ij} = \frac{\left(n_i^+(t)\frac{a_{ij}}{\deg v_i} + n_j^+(t)\frac{a_{ji}}{\deg v_j}\right) s_{ij}}{\left(n_i^+(t)\frac{a_{ij}}{\deg v_i} + n_j^+(t)\frac{a_{ji}}{\deg v_j}\right) s_{ij} + \left(n_i^-(t)\frac{a_{ij}}{\deg v_i} + n_j^-(t)\frac{a_{ji}}{\deg v_j}\right)[1 - s_{ij}]}$$

implies that $s_{ij} = 0$ and $s_{ij} = 1$ are fixed points.

The same is true for the vertex dominance in VDS. Let $s_i(t+1) = s_i(t) = s_i$ for some t and for all i. Thus,

$$s_i = \frac{s_i \sum_j n_j^+(t)\frac{a_{ji}}{\deg v_j}}{s_i \sum_j n_j^+(t)\frac{a_{ji}}{\deg v_j} + [1 - s_i]\sum_j n_j^-(t)\frac{a_{ji}}{\deg v_j}}$$

implies that $s_i = 0$ and $s_i = 1$ are fixed points.

4.3 Experimental results

Regarding the convergence speed, we experimentally compare the time the variables s_{ij} and s_i take to converge. The convergence time is obtained from 30 independently sampled networks $G(n,k,q,b)$ for every $n \in \{50, 100, \ldots, 500\}$, $k \in \{6, 7, \ldots, 10\}$, $q \in \{0.8, 1.2\}$, and $b \in \{1, 2, 3\}$. We set the fraction of initially labeled vertices in 10%.

We start both systems with population size

$$n_i^c(0) = \begin{cases} 0 & \text{if } v_i \in \mathscr{L} \text{ and } y_i \neq c, \\ \left|\mathscr{V} \setminus \{v_j | v_j \in \mathscr{L} \text{ and } y_j \neq c\}\right|^{-1} & \text{otherwise,} \end{cases}$$

to comply with the simplification requirements.

Fig. 7 depicts the convergence time by the number of edges in the network. The convergence time is quite different depending on the overlap of the networks.

For $q = 0.8$, only a few vertices have neighbors with different label; which results in less competition between particles of different teams. In this scenario, VDS benefits from the increasing number of edges and the variables s_i of VDS take fewer iterations to converge than the variables s_{ij} of EDS. It may happen because particles generated in the source reach all vertices of the same team earlier with more edges; which decreases the networks diameter.

For $q = 1.2$, however, the overlapping area is substantial. Since there is more competition, both

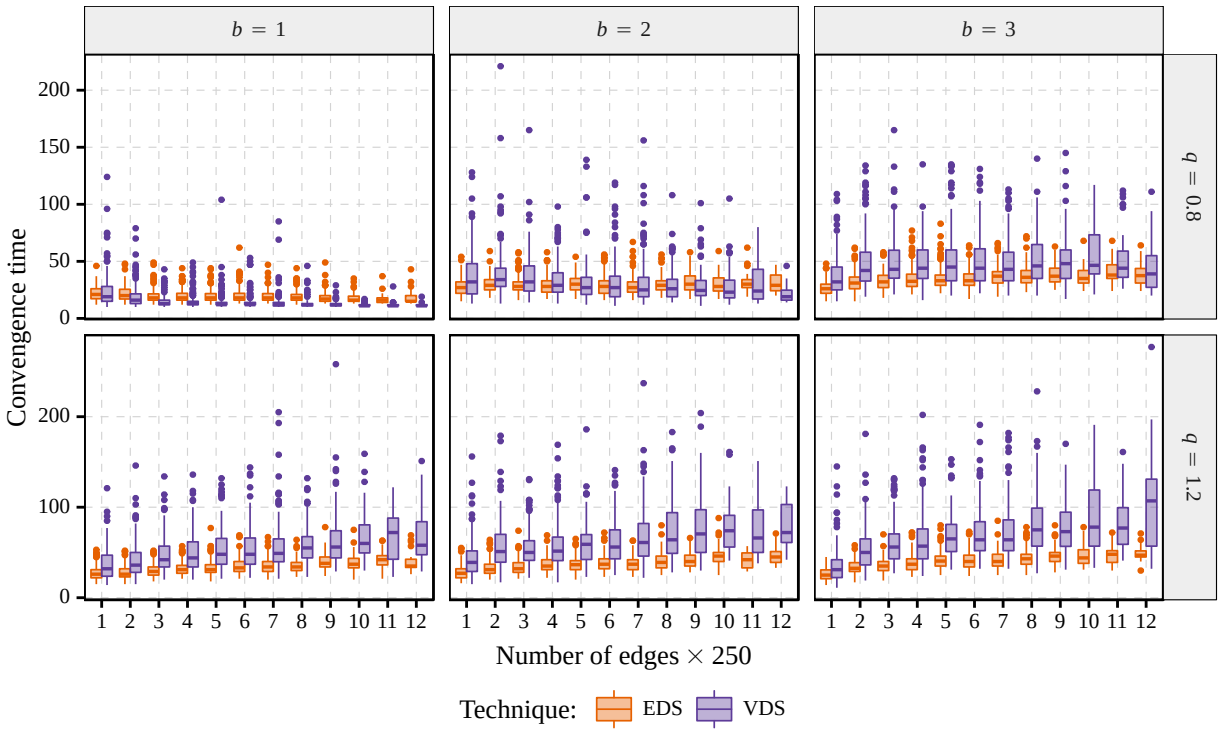

Fig. 7 Convergence time of variables s_{ij} in EDS and s_i in VDS for increasing number of edges in networks $G(n,k,q,b)$ with 10% of initially labeled vertices.

systems need more time to converge as the number of edges increases. In this case, the classification results of EDS converge faster than those of VDS.

Besides the differences between the systems, the convergence time in both of them is clearly sublinear in the number of edges of the network. Machine learning models can employ the convergent results of these systems with subquadratic time complexity order. EDS, however, offers much lower deviance in the time of convergence.

5 Conclusion

In this paper, we study the advantages of an edge-centric collective dynamics in machine learning tasks. The previously proposed Edge Domination System (EDS) [4] is a particle competition system that can be used to solve data classification tasks. We present a vertex-based version of the system, Vertex Domination System (VDS), and study their differences.

We show that the original system can be simplified without impairing its performance in learning problems. In summary, the models offer the same performance in classification and the same time complexity order. However, we conclude that EDS is preferred over VDS, as

- EDS acquires more information about the overlapping area between vertices with different labels, see section 3.2.3;
- particles in EDS tend to propagate farther producing a greater exploration behavior, see section 3.2.3;
- average convergence time of the classification results in EDS are lower than in VDS, see section 4.3;
- standard deviation of the convergence time of the classification results in VDS seems to increase

with the number of edges, see section 4.3.

In future works, we will propose a new classification mechanism for EDS that takes more advantage over the unfoldings. We expect that this classification mechanism will increase the classification accuracy of EDS. We also intend to estimate the expected time of convergence analytically.

Acknowledgments

This research was supported by the São Paulo State Research Foundation (FAPESP), the Coordination for the Improvement of Higher Education Personnel (CAPES), and the Brazilian National Research Council (CNPq).

References

[1] Gueleri, R.A., Cupertino, T.H., de Carvalho, A.C.P.L.F., and Zhao, L. (2014), A flocking-like technique to perform semi-supervised learning, *International Joint Conference on Neural Networks (IJCNN)*, IEEE, pp. 1579–1586, DOI: 10.1109/IJCNN.2014.6889434.

[2] Zhang, Z., Long, K., Wang, J., and Dressler, F. (2014), On swarm intelligence inspired self-organized networking: Its bionic mechanisms, designing principles and optimization approaches, *IEEE Commun. Surveys Tuts.*, **16**(1), pp. 513–537.

[3] Breve, F., Quiles, M.G., and Zhao, L. (2015), Interactive image segmentation using particle competition and cooperation, *International Joint Conference in Neural Networks, Proceedings*, IEEE, pp. 1–8.

[4] Verri, F.A.N., Urio, P.R., and Zhao, L. (2016), Network unfolding map by edge dynamics modeling, arXiv:1603.01182.

[5] Silva, T.C. and Zhao, L. (2016), Machine Learning in Complex Networks, 1st ed., Springer, p. 331, ISBN: 978-3-319-17290-3, DOI: 10.1007/978-3-319-17290-3.

[6] Urio, P.R., Verri, F.A.N., and Zhao, L. (2016), Features of Edge-Centric Collective Dynamics in Machine Learning Tasks, *International Conference on Nonlinear Science and Complexity, Proceedings.*