Pattern Recognition using Complex Networks

Thiago Christiano Silva[1] and Liang Zhao
Department of Computer Sciences, Institute of Mathematics and Computer Science (ICMC)
University of São Paulo (USP)
Av. Trabalhador São-carlense, 400, 13560-970, São Carlos, SP, Brazil
e-mail: {thiagoch, zhao}@icmc.usp.br.

***Abstract*—Complex networks have emerged as a unified representation of complex systems in various branches of science. In this work, we analyze the problem of clustering and recognizing handwritten patterns using two different perspectives with the aid of complex networks. In the first approach, we use a rigorously described competitive learning system composed of several particles navigating in a network with the purpose of conquering as many vertices as they are able to. The particle's walking rule is composed of a stochastic combination of random and preferential movements. At the end of the particle competition process, each particle dominates a pattern cluster. Computer simulations reveal that the proposed technique presents high precision of pattern detection, as well as low computational complexity. In the second approach, we propose a hybrid high level classification technique for pattern recognition. In this scheme, the hybrid classifier comprises a mixture of low and high level classifiers, in which the former derives its decisions using physical features of the data items and the latter predicts the class labels using the pattern formations of the data by exploiting the topological properties of the network. Again, we show that the hybrid classifier can really improve the pattern recognition rate, where it is capable of identifying variations and distortions of handwritten digit images.**

***Keywords*-Stochastic competitive learning; high level classification; data clustering; pattern recognition.**

I. Introduction

Over the last decade, there has been an increasing interest in network research, with the focus shifting away from the analysis of small graphs to the consideration of large-scale graphs, called *complex networks*. Such networks have emerged as a unified representation of complex systems in various branches of science. In general, they are used to model systems which have a nontrivial topology and are composed of a large amount of vertices [1], [2]. Machine learning and pattern recognition techniques using complex networks have triggered increased attention. This is because networks are ubiquitous in nature and everyday life. Examples include the Internet, the World Wide Web, biological neural networks, social networks (between individuals and among companies and organizations), food webs, metabolic networks and distribution (such as the bloodstream, postal delivery routes and distribution of electric energy), etc. The main motivation of graph theory research is the ability to describe the topological structure of the original system. In the machine learning domain, it has been shown that the topological structure is quite useful to detect clusters of arbitrary forms in data clustering [3].

In this paper, we tackle the problem of clustering and recognizing handwritten patterns using two different network-based perspectives. In the first, the patterns are grouped by means of a competitive learning system, which is composed of several particles navigating in a network with the purpose of conquering as many vertices as they are able to. In this case, it is expected that the similar patterns to be unveiled by analyzing the subgroup of vertices that each particle is dominating. The salient advantage of the model is that it is robust and has low computational complexity. Also, results indicate that the competitive system is able to obtain good pattern detection rates. In the second, the patterns are recognized via a novel hybrid high level classifier. In this sense, the hybrid classifier is composed of a mixture of low and high level classifiers, in which the former derives its decisions using physical features or class topologies of the data items (for instance: SVM, decision tree) and the latter predicts using the pattern formations of the data. The high level order of learning does not use spatial relation among data items, but instead it exploits the topological properties of the network, which is built using the data items, in search of hidden pattern formation. Simulation results show that the proposed hybrid classifier is able to outperform traditional classification techniques when applied to handwritten digits recognition.

The remainder of the paper is organized as follows: Sections II and III discuss the particle competition and the hybrid high level classifier, respectively; Section IV draws the final remarks of the paper; and Section V lists the publications of this research work.

II. Particle Competition Algorithm

In this section, we review the particle competition algorithm [4], [5]. The readers are invited to read the aforementioned references for more details.

A. Overview of the Technique

In the proposed competitive learning model, a set of particles $\mathcal{K} = \{1, ..., K\}$ is put into the vertices of the network $G = \langle \mathcal{V}, \mathcal{E} \rangle$ in a random manner, where $\mathcal{V} = \{1, \ldots, V\}$ is the set of vertices and $\mathcal{E} = \{1, \ldots, E\}$, the set of edges. Each particle can be conceptualized as a flag carrier with its main objective being to conquer new vertices, while defending its current dominated vertices. When a particle visits an arbitrary

[1]This paper is a brief overview of the most relevant research topics derived by the first author in his ongoing Doctoral Thesis.

vertex, it strengthens its own domination level on that vertex and, simultaneously, weakens the domination levels of all other rival particles on the same vertex. It is expected that this model, in a broad horizon of time, will end up uncovering the communities in the network in such a way that each particle dominates a community.

A particle in this model can be in two states: *active* or *exhausted*. Whenever the particle is active, it navigates in the network according to a combined behavior of random and preferential walking. The random walking term is responsible for the adventuring behavior of the particle, i.e., it randomly visits vertices without taking into account their domination levels. The preferential walking term is responsible for the defensive behavior of the particle, i.e., it prefers to reinforce its owned territory rather than visiting a vertex that is not being dominated by that particle. So as to make this process suitable, each particle carries an energy term with it. This energy increases when the particle is visiting an already dominated vertex by itself, and decreases whenever it visits a vertex that is being owned by a rival particle. If this energy drops under a minimum allowed value, the particle becomes exhausted and is teleported back to a safe ground, which is one of the vertices dominated by the present particle. At the next step, the exhausted particle will be possibly recharged by visiting the vertices dominated by itself. With this confinement mechanism, we expect to restrain the acting region of each particle and, thus, reduce long range and redundant visits in the network.

B. Proposed Model

The proposed dynamical system is nonlinear and stochastic. The internal state of the system is given by $X(t) = [p(t)\ N(t)\ E(t)\ S(t)]^T$, where:

- $p(t)$ is a K-dimensional vector, where the k-entry denotes the position of particle k at the instant t.
- $N(t)$ is a $N \times K$ matrix, where the (i,k)-entry indicates the number of visits that vertex i has received from particle k up to time t.
- $E(t)$ is a K-dimensional vector, where the k-entry provides the energy of particle k at the instant t.
- $S(t)$ is a K-dimensional vector, where the k-entry supplies the state of particle k at time t: active or exhausted.

The update rule of the proposed dynamical system is expressed by:

$$\phi : \begin{cases} p^{(k)}(t+1) & = j, \quad j \sim \mathbb{P}^{(k)}_{\text{transition}}(t) \\ N_i^{(j)}(t+1) & = N_i^{(j)}(t) + \delta\left(p^{(j)}(t+1) = i\right) \\ E^{(j)}(t+1) & = \begin{cases} \min(\omega_{\max}, E^{(j)}(t) + \Delta), \text{if owner}(j,t) \\ \max(\omega_{\min}, E^{(j)}(t) - \Delta), \text{if } \sim \text{owner}(j,t) \end{cases} \\ S^{(j)}(t+1) & = \delta\left(E^{(j)}(t+1) = \omega_{\min}\right) \end{cases} \tag{1}$$

where $\delta(.)$ is the indicator function that yields 1 if the argument is logically true and 0, otherwise. In the following, we present the meaning of the four expressions shown in the competitive dynamical system ϕ:

- *Particle's Transition Rule (1st Expression)*: This rule provides a stochastic process for moving each particle to a new vertex j, where j is determined according to the time-varying transition matrix:

$$\mathbb{P}^{(k)}_{\text{transition}}(t) \triangleq (1 - S^{(k)}(t)) \left[\lambda \mathbb{P}^{(k)}_{\text{pref}}(t) + (1-\lambda)\mathbb{P}^{(k)}_{\text{rand}}\right] + S^{(k)}(t)\mathbb{P}^{(k)}_{\text{rean}}(t). \tag{2}$$

  Note that the two terms are mutually exclusive and are triggered depending on the value of the state of particle k, $S^{(k)}(t)$. When it is active ($S^{(k)}(t) = 0$), only the first term is enabled. In this case, the particle performs a mixture of random-preferential walk, whose distribution matrices are given by:

$$\mathbb{P}^{(k)}_{\text{rand}}(i,j) \triangleq \frac{a_{i,j}}{\sum_{u=1}^{V} a_{i,u}}, \tag{3}$$

$$\mathbb{P}^{(k)}_{\text{pref}}(i,j,t) \triangleq \frac{a_{i,j}\bar{N}^{(k)}_j(t)}{\sum_{u=1}^{V} a_{i,u}\bar{N}^{(k)}_u(t)}, \tag{4}$$

  where $\bar{N}(t)$ denotes the domination matrix of the system at time t, which is given by the row normalization of the matrix $N(t)$.
  Now, when $S^{(k)}(t) = 1$, the particle becomes exhausted and the second term indicated by the matrix $\mathbb{P}^{(k)}_{\text{rean}}(t)$ is enabled. Such matrix is given by:

$$\mathbb{P}^{(k)}_{\text{rean}}(i,j,t) \triangleq \frac{\delta\left(\arg\max_{m \in \mathcal{K}} \left(\bar{N}^{(m)}_j(t)\right) = k\right)}{\sum_{u=1}^{V} \delta\left(\arg\max_{m \in \mathcal{K}} \left(\bar{N}^{(m)}_u(t)\right) = k\right)}. \tag{5}$$

- *Update Rule of the Number of Visits (2nd Expression)*: The update rule states that whenever a particle visits a vertex, the corresponding entry in $N(t)$ must be incremented.
- *Update Rule of the Particle's Energy (3rd Expression)*: In the model, each entry of $E(t)$ is limited by the interval $[\omega_{\min}, \omega_{\max}]$. $\Delta > 0$ symbolizes the increment or decrement of energy that each particle receives at time t. The increment and decrement of energy are given by the first and second expressions, respectively.
- *Update Rule of the Particle's State (4th Expression)*: This expression derives the state of the particle by monitoring whether its current energy is at the minimum threshold.

In the next subsections, we present some extensions of the model that have been made by the authors. Due to the page limit, we will not go deeper into them.

C. Mathematical Analysis of the Model

In [5], we have analyzed the competitive system in a mathematical form. Specifically, we have derived closed probabilistic formula that supplies the distribution of $\bar{N}(t)$ over time. With this tool, one can analytically understand the dynamics of the

model and predict the outcome of the model without using numerical models. A validation has been performed and the derived expressions do approximate the real behavior of the model.

D. Estimating the Number of Communities or Groups in a Data Set

In [4], we have devised how to estimate the number of communities or groups in a data set. Specifically, we use the average maximum domination level $\langle R(t) \rangle \in [0, 1]$ measure, which is given by the following expression:

$$\langle R(t) \rangle = \frac{1}{V} \sum_{u=1}^{V} \max_{m \in \mathcal{K}} \left(\bar{N}_u^{(m)}(t) \right). \tag{6}$$

This measure has been shown to work well both in artificial and real-world data sets. It will be utilized in our simulations in the computer simulation section.

E. Discovering Overlapped Patterns

In [6], we have derived a method for detecting overlapping vertices (patterns) only using information generated by the competition process itself. In contrast to the majority of the related techniques, which occasionally possesses a dedicated process to infer overlapping characteristics of the items, our technique artlessly supplies this information.

F. Computer Simulations

In all simulations, we will make use of the USPS, MNIST, and Letter Recognition data sets. While the first two have 10 classes representing the digits "0" to "9", the third one has 26 classes indicating the English alphabet. More details of these data sets as well as the configuration of the particle competition technique are available in [4], [7].

1) Estimating the number of groups: Here, we use the heuristic which has been developed in [4], [7]. Figures 1a, 1b, and 1c 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.

2) Handwritten Data Clustering: Here, we report the cluster detection accuracy reached by our algorithm in detail, along with the data clustering accuracy reached by LCGMM, GMM, K-Means, NCut and variations. These techniques are described in [7]. For the calculation of the cluster detection accuracy, we set that the ideal result is that each cluster represents a "digit" (in the USPS and MNIST data set) or a "letter" (in the Letter Recognition data set).

Table I reports the data clustering accuracy reached by our method and the aforementioned competing algorithms. As we can verify by looking at the Average Rank column, our algorithm has reached one of the best positions, showing the effectiveness of the proposed technique. In order to examine the results in a statistical manner, we utilize the Friedman Test. This test is used to check whether the measured average ranks are significantly distinct from the mean rank. Applying it with a 10% significance level, the null-hypothesis is rejected, meaning that the algorithms are statistically different. Nonetheless, one can see that our algorithm has obtained the best average rank in relation to the other algorithms for these three data sets.

TABLE I
RESULTS FOR THE DATA CLUSTERING ACCURACY. THE MEAN VALUE OF THIRTY INDEPENDENT RUNS IS PROVIDED.

	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**
Proposed Technique	80.46	74.53	91.37	**2.00**

In order to further verify the robustness of the proposed technique, we inspect the samples that compose the same cluster. Specifically, Fig. 2 shows some samples of the cluster representing the pattern "2" of the MNIST data set. These samples are captured using the following strategy: we 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 the these number patterns each of which in a single representative cluster, showing the robustness of the proposed model.

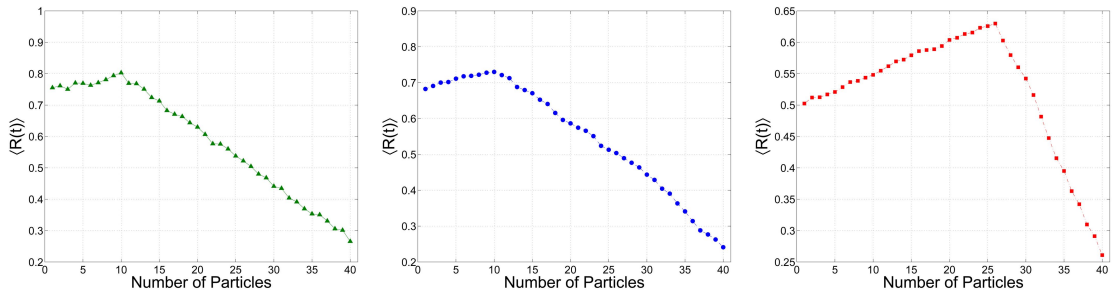
III. HYBRID HIGH LEVEL CLASSIFIER

In this section, we review the hybrid high level classifier [8]. The readers can get more details in [8].

A. Overview of the Technique

The proposed technique works in a fully supervised network-based environment. With this in mind, the induction is performed in two steps:

- *Training Phase*: the data items are mapped into a graph such that each class holds a unique component (subgraph). For example, Fig. 3a shows a schematic of how the network looks like for a three-class problem when the training phase has been completed. In this case, each class holds a representative component. In the figure, the surrounding circles denote these components: $\mathcal{G}_{C_1}$, $\mathcal{G}_{C_2}$, and $\mathcal{G}_{C_3}$.
- *Classification Phase*: the test instances are presented to the classifier one by one. Since we do not know the label of the test instance, it is temporally inserted into the network in a way that it is connected to its most similar vertices. Once the data item is inserted, each



(a) (b) (c)

Fig. 1. Determination of the optimal number of particles K (the optimal number of clusters) in real-world data sets. 20 independent runs are performed and the average value is reported.

(a) (b) (c) (d) (e) (f) (g)

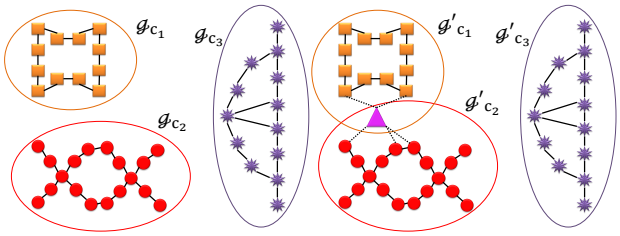
Fig. 2. A broad set of samples of that were classified as being member of the cluster representing the pattern "2". The transitions from the sample (a) to (g) were captured from the maximum geodesic distance between two vertices in the cluster representing pattern 2. In this case, the diameter of such cluster is 17. We have only provided 7 representative samples above.

class analyzes, in isolation, its impact on the respective class component using the complex topological features of it. In the high level model, each class retains an isolated graph component. Each of these components calculate the changes that occur in its pattern formation with the insertion of this test instance. If slight or no changes occur, then it is said that the test instance is in compliance with that class pattern. As a result, the high level classifier yields a great membership value for that test instance on that class. Conversely, if these changes dramatically modify the class pattern, then the high level classifier produces a small membership value on that class. These changes are quantified via network measures, each of which numerically translating the organization of the component from a local to global fashion. For the sake of clarity, Fig. 3b exhibits a schematic of how the classification process is performed. Note that, once the test instance gets classified, it is either discarded or incorporated to the training set with the corresponding predicted label. In any case, each class is still represented by a single graph component.

B. Hybrid High Level Classification

The hybrid classifier M consists of a convex combination of two terms: (i) a low level classifier and (ii) a high level classifier, which is responsible for classifying a test instance according to its pattern formation with the data. Mathematically, the membership of the test instance $x_i \in \mathbf{X}_{\text{test}}$ with respect to the class $j \in \mathcal{L}$, here written as $M_i^{(j)}$, is given by:

$$M_i^{(j)} = (1-\lambda)T_i^{(j)} + \lambda C_i^{(j)}, \tag{7}$$



(a) (b)

Fig. 3. (a) Schematic of the network in the training phase. (b) Schematic of how the classification inference is done.

where $T_i^{(j)} \in [0,1]$ and $C_i^{(j)} \in [0,1]$ denote the memberships of the test instance x_i on class j produced by a low level and a high level classifier, respectively, and $\lambda \in [0,1]$ is the *compliance term*, which plays the role of counterbalancing the classification decision supplied by both classifiers.

The inference of pattern formation, which is used by the classifier C, is processed using the generated network. The motivation behind using networks is that it can describe topological structures among the data items. With that in mind, the pattern formation is extracted using network measures developed in the complex network literature. Suppose that K measures are selected to comprise the high level classifier C. Mathematically, the membership of the test instance $x_i \in \mathbf{X}_{\text{test}}$ with respect to the class $j \in \mathcal{L}$ yielded by the high level classifier, here written as $C_i^{(j)}$, is given by:

$$C_i^{(j)} = \frac{\sum_{u=1}^{K} \alpha(u) \left[1 - f_i^{(j)}(u)\right]}{\sum_{g \in L} \sum_{u=1}^{K} \alpha(u) \left[1 - f_i^{(g)}(u)\right]}, \tag{8}$$

where $\alpha(u) \in [0,1], \forall u \in \{1,\ldots,K\}$, $\sum_{u=1}^{K} \alpha(u) = 1$, are user-controllable coefficients that indicate the influence of each network measure in the classification process and $f_i^{(j)}(u)$ is a function that depends on the uth network measure applied to the ith data item with regard to the class j. This function is responsible for providing an answer whether the test instance x_i presents the same patterns of the class j or not. The denominator in (8) has been introduced solely for normalization matters.

With respect to $f_i^{(j)}(u)$, it possesses a general closed form given by:

$$f_i^{(j)}(u) = \Delta G_i^{(j)}(u) p^{(j)}, \tag{9}$$

where $\Delta G_i^{(j)}(u) \in [0,1]$ is the variation of the uth network measure that occurs on the component representing class j if x_i joins it and $p^{(j)} \in [0,1]$ is the proportion of data items pertaining to the class j. Remembering that each class has a component representing itself, the strategy to check the pattern compliance of a test instance is to examine whether its insertion causes a great variation of the network measures representing the class component.

The high level term of the hybrid classifier is composed by the assortativity, clustering coefficient, and average degree. The reason why these three measures have been chosen is as follows: the degree measure figures out strict local scalar information of each vertex in the network; the clustering coefficient of each vertex captures local structures by means of counting triangles formed by the current vertex and any of its two neighbors; and the assortativity coefficient considers not only the current vertex and its neighbors, but also the second level of neighbors (neighbor of neighbor), the third level of neighbors, and so on. We can perceive that the three measures characterize the network's topological properties in a local to global fashion. In this way, the combination of these measures is expected to capture the pattern formation of the underlying network in a systematic manner.

C. Computer Simulations

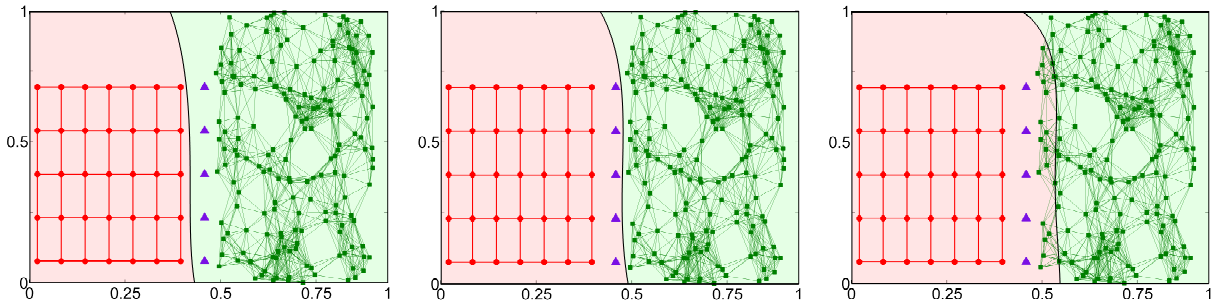
1) Simple Example Showing How the Technique Works: This example serves as the gist of how the proposed classifier draws its decisions. Consider the structured toy data set illustrated in Fig. 4, in which there are 2 classes: a red or "circle" (35 vertices) and a green or "square" (120 vertices). The fuzzy SVM with RBF kernel ($C = 100$ and $\gamma = 2^{-2}$) is adopted for the low level classifier. By inspection of the figure, the red or "circle" class shows a strong clear pattern: a grid or lattice, whereas the green or "square" class does not indicate any clear patterns. The goal is to classify the triangle-shaped data items (test set) one by one only using the information of the training set. Figures 4a, 4b, and 4c exhibit the decision boundaries of the two classes when $\lambda = 0$, $\lambda = 0.5$, and $\lambda = 0.8$, respectively. When $\lambda = 0$, only the SVM prediction is used by the proposed technique. In this case, one can see that the five data items are not correctly classified. Notice that the decision boundaries are pushed near the red or "circle" class by virtue of the large amount of green or "square" items in the vicinity. Now, when $\lambda = 0.5$, the SVM and the high level classifier predictions are used in the same intensity. In this case, the decision borders are dragged toward the green or "square" class, because of the strong pattern that the red or "square" class exhibits. When $\lambda = 0.8$, the decision derived by the high level classifier is so strong that is capable of pushing the decision boundaries inside the high density area of the green or "square" class. This happens on account of the strong pattern that the red or "square" class shows. In the two former cases, the proposed technique can successfully classify the triangle-shaped data items.

2) Handwritten Digits Recognition: Figure 5 shows the accuracy rate reached by the three discussed techniques against several values of λ when the MNIST data set is used. Our main goal here is to reveal that a proper mixture of traditional and high level classifiers is able to increase the overall accuracy rate of the predicting model. Investigating this figure, one can see that the perceptron neural network achieved 88% of classification rate when only a traditional classifier ($\lambda = 0$) is employed. A little increase of the compliance term, in the case, $\lambda = 0.25$, is responsible for a boost in the accuracy rate. In particular, it achieves almost 91% of correctly labeled test instances. With regard to the k-nearest neighbor algorithm, for a pure traditional classifier, we have obtained 95% of accuracy rate, against 96% when $\lambda = 0.2$. In the ϵ-radius classifier, we have obtained 98.49% of correctly classified data items for $\lambda = 0$, against 99.06% when $\lambda = 0.2$. Finally, this experiment indicates that the optimal compliance term might be intrinsic to the data set, since, for three completely distinct low level classifiers, the maximum accuracy rate is achieved in the surroundings of $\lambda = 0.225$. We intend to investigate this conjecture as future work.

It should be mentioned that the classification rate improvement by using high level technique is significant because the applied traditional classification techniques have already been optimized by using the best parameter values.

IV. Final Remarks

This paper presents two main results obtained in the research of the ongoing doctorate thesis. In the first one, it is provided a rigorous definition of a new model for competitive learning in complex networks, 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. A mathematical analysis and several extensions have been discussed. In the second one, a novel combination of low and high level classifiers (hybrid) has been introduced to perform supervised data classification using high order of learning. The low level term classifies test instances according to their physical features or class topologies, while the second term measures the compliance to the pattern formation of the training data, by means of exploiting the complex topological properties of



(a) (b) (c)

Fig. 4. Decision boundaries when (a) $\lambda = 0$; (b) $\lambda = 0.5$; and (c) $\lambda = 0.8$

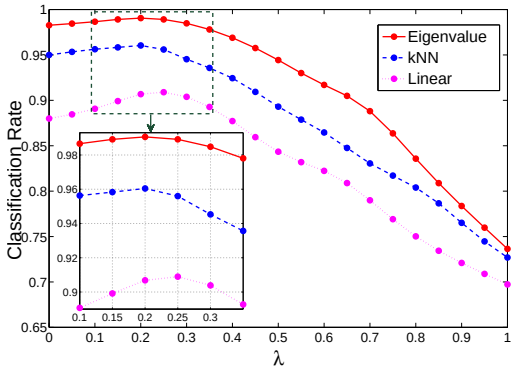


Fig. 5. An analysis of the impact of the compliance term λ on three different traditional low level techniques applied to the MNIST database.

the network. In both cases, several computer experiments have been undertaken and satisfactory results have been obtained.

V. Publications during the Doctorate Period

During the doctorate period, several research results have been obtained and a series of papers have been published. The main results have been published in IEEE Transactions on Neural Networks and Learning Systems [4], [5], [8], which is one of the leading scientific journals with impact factor 2.6 currently. In the doctorate period, up to now (is still active), we have generated the following publications:

- Seven publications in international journals (see [4], [5], [7]–[11]);
- Eight publications in international and national conferences (see [6], [12]–[18]).
- Various manuscripts are at submission stage.

Acknowledgment

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

References

[1] M. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.

[2] A. Barrat, M. Barthélemy, and A. Vespignani, *Dynamical Processes on Complex Networks*. Cambridge University Press, 2008.

[3] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, pp. 75–174, 2010.

[4] T. C. Silva and L. Zhao, "Stochastic competitive learning in complex networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 3, pp. 385–398, 2012.

[5] ——, "Network-based stochastic semisupervised learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 3, pp. 451–466, 2012.

[6] T. C. Silva, T. H. Cupertino, and L. Zhao, "Detecting overlapping structures via network-based competitive learning," in *WCCI 2012 IEEE World Congress on Computational Intelligence, Proceedings of the International Joint Conference on Neural Networks (IJCNN 2012)*, 2012, pp. 3036–3043.

[7] T. C. Silva, L. Zhao, and T. H. Cupertino, "Handwritten data clustering using agents competition in networks," *Journal of Mathematical Imaging and Vision*, DOI: 10.1007/s10851-012-0353-z, 2012.

[8] T. C. Silva and L. Zhao, "Network-based high level data classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 6, pp. 954–970, 2012.

[9] ——, "Semi-supervised learning guided by the modularity measure in complex networks," *Neurocomputing*, vol. 78, no. 1, pp. 30 – 37, 2012.

[10] T. C. Silva and D. R. Amancio, "Word sense disambiguation via high order of learning in complex networks," *Europhysics Letters*, vol. 98, p. 58001, 2012.

[11] T. H. Cupertino, T. C. Silva, and L. Zhao, "Classification of multiple observation sets via network modularity," *Neural Computing and Applications*, 2012 (to appear).

[12] T. C. Silva and L. Zhao, "Pixel clustering by using complex network community detection technique," in *Proceedings of 7th International Conference on Intelligent Systems Design and Applications*, 2007, pp. 925–932.

[13] ——, "Network-based learning through particle competition for data clustering," in *2011 International Joint Conference on Neural Networks*, 2011, pp. 45–52.

[14] ——, "Semi-supervised learning in complex networks," in *XXXI Congresso da Sociedade Brasileira de Computação, ENIA*, 2011, pp. 406–417.

[15] T. C. Silva, T. H. Cupertino, and L. Zhao, "Stochastic competitive learning applied to handwritten digit and letter clustering," in *Proceedings of the XXIV Sibgrapi Conference on Graphics, Patterns and Images*, 2011, pp. 313–320.

[16] ——, "High level classification for pattern recognition," in *Proceedings of the XXIV Sibgrapi Conference on Graphics, Patterns and Images*, 2011, pp. 344–351.

[17] ——, "Preventing error propagation in semi-supervised learning," in *Proceedings of The Ninth International Symposium on Neural Networks (ISNN 2012)*, vol. 7367, 2012, pp. 565–572.

[18] B. Araújo, F. A. Rodrigues, T. C. Silva, and L. Zhao, "Identifying abnormal nodes in protein-protein interaction networks," in *XI Brazilian Symposium on Artificial Neural Network (SBRN)*, 2010, pp. 97–102.