High Level Classification Totally Based on Complex Networks

Murillo G. Carneiro, Liang Zhao
Institute of Mathematics and Computer Science, ICMC
University of São Paulo, USP
São Carlos, Brasil 13566-590
Email: {carneiro, zhao}@icmc.usp.br

***Abstract*—Differently from traditional machine learning techniques applied to data classification, high level classification considers not only the physical features of the data (distance, similarity or distribution), but also the pattern formation of the data. In this latter case, a set of complex network measures are employed because of their abilities to capture spatial, functional and topological relations. Although high level techniques offer powerful features, good classification performance is usually obtained by combining them with some low level algorithms, which, in turn, reduces the efficiency of the overall technique. A priori, the reason is that low level and high level techniques provide different visions of classification. In this way, one cannot simply substitute another. This paper presents a data classification technique in which low level and high level classifications are embedded in a unique scheme, i.e., the proposed technique does not need a separated low level technique. The novelty is the use of a simple and recently proposed complex network measure, named component efficiency. Thus, our algorithm computes the efficiency of information exchanging among vertices in each component and the resulting values are used to drive the classification of the new instances i.e., a new instance will be classified into one of the components (class), in which their local features are in conformity with the insertion of the new instance. The experiments performed with artificial and real-world data sets show our approach totally based on complex networks is promising and it provides better results than some traditional classification techniques.**

Keywords*—*high level classification, complex networks, data classification, component efficiency measure, machine learning

I. Introduction

Complex networks have become one of the major research themes in complex systems and have been employed in a wide range of problems in many distinct fields of research (see [1]–[5] for instance). They are graphs with non trivial connections that gather concepts from statistics, complex systems and graph theory (see [6]–[8]. Their ability to detect spatial, functional and topological relations is also one of their most important characteristics.

Although complex networks have already been used in a large number of areas there are still plenty of tasks that could be potentially helped by them, such as data classification. In data classification, the techniques construct computer programs able to learn from labeled data sets and, subsequently, predict unlabeled instances [9]–[14] . Once there is a big number of applications for this task, many data classification techniques have been employed [15], such as the well-known decision trees [16], [17], Naive-Bayes [18], Artificial Neural Networks [19] and Support Vector Machines [20].

A drawback with these traditional machine learning techniques is that they consider exclusively the physical features of the data (eg. distance, similarity or distribution). This limited way to classify is known as low level classification. On the other hand, data classification algorithms that consider not only physical attributes, but also the pattern formation are referred to as high level classification [21]–[23]. These kind of classification exploits complex network properties to obtain a high level view on the data. As a motivation to investigate high level algorithms, Fig. 1 shows a data set with a clear pattern formation and everyone traditional algorithm is not able to classify the magenta diamond according to this pattern because they performs the classification only based on the data physical attributes.

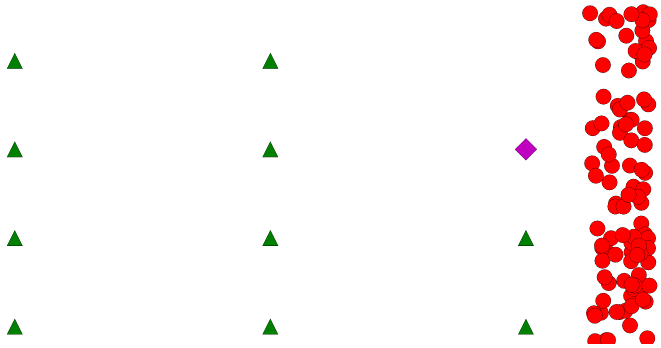

Fig. 1: Data set composed by data items of two classes (green $\triangle$ and red $\circ$). Green data items present a clear pattern. Magenta ($\Diamond$) data item needs to be classified

Normally, data classification using high level approaches can be divided into two phases: network construction and prediction. In the first, the training data are represented as a network in which each instance is a vertex and the edges (or links) denote the similarity relations between vertices. The techniques mostly used for this stage in the literature are K-NN and ϵ-radius [24]. In the second phase, the aim is to explore the network constructed in the first stage to classify new input instances unseen in the training.

The high level classification was originally proposed in [21], where a general framework that combines low and high level algorithms is presented. However, the approach have many parameters to set, especially for the network construction (eg. K-NN and ϵ-radius techniques). A non parametric way to construct the network in the high level classification is the motivation of the work proposed in [22] and refined in [23] that proposes also a high level algorithm without parameters, a new complex network measure to high level classification, named component efficiency, and a high level technique that works on components instead the whole classes.

A commun characteristic of all these works related to high level prediction is that despite the powerful characteristics of the complex networks, the high level approach always is combined with some low level algorithm. A priori, the reason for this is that low level and high level techniques provide different visions of classification. In this way, one cannot simply substitute another.

On the other hand, our investigation shows an approach totally based on complex networks, in which low level and high level classifications are embedded in a unique scheme, i.e., the proposed technique does not need a separated low level technique. Thus, differently from the previous works about high level classification, this paper presents a new approach that uses exclusively network measures to perform a high level classification. In fact, complex networks provide both high level and low level features. Therefore, our technique not is combined with everyone low level algorithm. This is an advantage especially because the proposed algorithm considers as the physical attributes as the pattern formation of the data sets. In addition, the computational cost of our algorithm is reduced because neither additional techniques nor parameters are necessary.

In more details, the great novelty in our approach is the use of a simple and recently proposed complex network measure (component efficiency). Our algorithm computes how efficiently the vertices in each component exchange information among themselves and uses these analyses to drive the classification of the new instances. Consequently, the classification process is guided by local features of the components and also establishes a heuristic that excludes components that are not in conformity with the insertion of the new instance. Initial experiments on artificial and real-world data sets were performed to evaluate the proposed technique. Simulations have shown that our algorithm is able to detect physical and topological relations on the components, is very robust to provide low level and high level features, and can find better results than some traditional machine learning techniques.

The remainder of the paper is organized as follows. The proposed technique and the contributions of this investigation are detailed in Sect. II. Empirical evaluation and discussions about the proposed algorithm on artificial and real data sets are showed in Sect. III. Finally, Sect. IV concludes the paper.

II. Model Description

Considering the machine learning field, graph-based techniques have been widely employed in the context of unsupervised [4], [24] and semi-supervised learning [25]–[27], especially in problems related to clustering, transductive learning and dimensionality reduction. However, the investigation of complex network-based techniques for supervised learning is recent [21]–[23], [28], [29] and the results obtained show complex networks are very effective when applied to supervised learning problems, such as data classification and features selection.

This paper offers a new view about the high level data classification. Previous studies have combined high level and low level classifications, as a framework, to obtain good classification performance. The motivation of them is that both low level and high level techniques provide different visions of classification. In this way, we present a data classification technique in which low level and high level classifications are embedded in a unique scheme, i.e., the proposed technique does not need a separated low level technique. The advantages of this approach are the reduction in the computational cost, the absence of parameters and mainly the exploitation of complex network measures in the whole classification process.

Basically, the proposed technique has three major steps: (i) construction of the network in the training phase, (ii) filtering of components to be considered for each new instance and (iii) the application of complex network measures to perform the high level classification. Initially, the network is generated from the k-associated optimal graph proposed in [28]. In the following step, the high level algorithm uses a complex network measures, named component efficiency measure to verify what the components where each new instance can be inserted. In the third step, the high level algorithm is applied to classify new instances by checking the variation of complex network measures in each component [filtered by step (ii)] before and after the insertion of the new instance.

In this section, the high level technique totally based on complex networks is mathematically defined. For the sake of clarity, it was divided into two subsections: Subsect. II-A, which describes how work the high level algorithm, and Subsect. II-B, which provides the major novelty of this paper: the use of a complex network measure, named component efficiency [23], to drive the classification process.

A. High Level Technique

The high level classification of a new instance y for a given class j, is given by:

$$C_y^{(j)} = \frac{\max\limits_{C_y^\alpha \in j} H_y^{(\alpha)}}{\sum_j \max\limits_{C_y^\alpha \in j} H_y^{(\alpha)}}, \qquad (1)$$

where $\alpha \in j$ denotes all components that belong to class j and $C_y^{(j)} \in [0,1]$ receives the highest value of $H_y^{(\alpha)}$, such that $\alpha \in j$, divided by a normalization term. The idea is very simple: from the filtered components of y, the high level technique examines these components in a way that the algorithm associates the new instance y with the more compatible component for each class and computes the classification probabilities for them.

In (1), $H_y^{(\alpha)}$ means the compatibility of each component α with the new instance y and is given by:

$$H_y^{(\alpha)} = \frac{\sum_{u=1}^{Z} \delta_y(u)[1 - f_y^{(\alpha)}(u)]}{\sum_{g \in L} \sum_{u=1}^{Z} \delta_y(u)[1 - f_y^{(g)}(u)]}, \qquad (2)$$

in which u is related to the network measures employed in the high level algorithm, $\delta_y(u) \in [0,1], \forall u \in \{1,\ldots,Z\}$ indicates the influence of each network measure in the classification process and $f_y^{(\alpha)}(u)$ provides an answer whether the test instance y presents the same patterns of the component α or not, considering the u-th network measure. The denominator term in (2) is only for normalization.

An automatic way for the weight assignment among the employed network measures, i.e., the δ term in Eq. (2) is presented in [23] and determined by:

$$\delta_y(u) = \frac{1 - (\max\limits_{\alpha} \Delta G_y^{(\alpha)}(u) - \min\limits_{\alpha} \Delta G_y^{(\alpha)}(u))}{\sum_{u=1}^{K} 1 - (\max\limits_{\alpha} \Delta G_y^{(\alpha)}(u) - \min\limits_{\alpha} \Delta G_y^{(\alpha)}(u))}, \tag{3}$$

where $\Delta G_y^{(\alpha)}(u) \in [0,1]$ represents the variation that occurs in a complex network measure whenever a new instance $y \in Y$ is inserted. This definition is important because it does not allow only one network measure to dominate the classification decision. Note that term δ is valid only if $\sum_{u=1}^{K} \delta_y(u) = 1$.

About $f_y^{(j)}(u)$, it is given by:

$$f_y^{(\alpha)}(u) = \frac{\Delta G_y^{(\alpha)}(u)\ p^{(\alpha)}}{\sum\limits_{\alpha} \Delta G_y^{(\alpha)}(u)\ p^{(\alpha)}}, \tag{4}$$

in which $\Delta G_y^{(\alpha)}(u) \in [0,1]$ represents the variation that occurs in a complex network measure whenever a new instance $y \in Y$ is inserted and $p^{(\alpha)} \in [0,1]$ is the proportion of instances that belong to component α.

Complex network measures are used to provide a high level analysis of the data [30]. When a new instance y needs to be classified, the technique computes the impact by inserting this new vertex for each filtered component in an isolated way. Basically, the variation in the results in network measures indicates which is the component that y belongs. In other words, if there is a little variation in the pattern formation of that component when connecting y to it, a high level prediction returns a high value indicating that y is according with this pattern. In the opposite, if there is a great variation when linking y to a component, it returns a small value denoting that y is not according with this pattern.

In this work, three network measures are employed to check the pattern formation of the input data: assortativity, clustering coefficient and average degree [31].

1) Assortativity: The assortativity measure quantifies the tendency of connections between vertices [31] in a complex network. This measure analyzes whether a link occurs preferentially between vertices with similar degree or not. The assortativity with regards to each component α of the data set is given by:

$$r^{(\alpha)} = \frac{L^{-1} \sum\limits_{u \in U_\alpha} i_u k_u - [L^{-1} \sum\limits_{u \in U_\alpha} \frac{1}{2}(i_u + k_u)]^2}{L^{-1} \sum\limits_{u \in U_\alpha} \frac{1}{2}(i_u^2 + k_u^2) - [L^{-1} \sum\limits_{u \in U_\alpha} \frac{1}{2}(i_u + k_u)]^2} \tag{5}$$

where $r^{(\alpha)} \in [-1,1]$, $U_\alpha = \{u : i_u \in \alpha \wedge k_u \in \alpha\}$ encompasses all the edges within component α, u represents an edge, and i_u, k_u indicate the vertices at each end of edge u.

Therefore, the membership value of a test instance y with respect to the component α is given by:

$$\Delta G_y^{(\alpha)}(1) = \frac{\mid r'^{(\alpha)} - r^{(\alpha)} \mid}{\sum\limits_{u \in U} \mid r'^{(u)} - r^{(u)} \mid}, \tag{6}$$

2) Clustering Coefficient: Clustering coefficient is a measure that quantifies the degree at which local nodes in a network tend to cluster together [6]. The clustering coefficient with regards to each component α of the data set is given by:

$$CC_i^{(\alpha)} = \frac{\mid e_{us} \mid}{k_i(k_i - 1)}, \tag{7}$$

$$CC^{(\alpha)} = \frac{1}{V_\alpha} \sum_{i=1}^{V_\alpha} CC_i^{(\alpha)}, \tag{8}$$

in which $CC_i^{(\alpha)} \in [0,1]$ and V_α denotes the number of vertices in component α. Thus, the membership value of a test instance y with respect to the component α is given by:

$$\Delta G_y^{(\alpha)}(2) = \frac{\mid CC'^{(\alpha)} - CC^{(\alpha)} \mid}{\sum\limits_{u \in U} \mid CC'^{(u)} - CC^{(u)} \mid}. \tag{9}$$

3) Average Degree: The average degree is a very simple measure. It quantifies, statistically, the average degree of the vertices in a component. The average degree with regards to each component α is given by:

$$\langle k^{(\alpha)} \rangle = \frac{1}{V_\alpha} \sum_{i=1}^{V_\alpha} k_i^{(\alpha)}, \tag{10}$$

in which $k^{(\alpha)} \in [0,1]$ and V_α denotes the number of vertices in component α. Regarding the membership value of a test instance $y \in Y$ with respect to component α, it is given by:

$$\Delta G_i^{(\alpha)}(3) = \frac{\mid \langle k'^{(\alpha)} \rangle - \langle k^{(\alpha)} \rangle \mid}{\sum\limits_{u \in \Gamma} \mid \langle k'^{(u)} \rangle - \langle k^{(u)} \rangle \mid} \tag{11}$$

B. The Unified Approach

The great novelty in this paper is the use of a simple and recently proposed complex network measure, named component efficiency, to drive the classification process. In this way, our algorithm computes the efficiency of exchanging information between the vertices of each component and the resulting values of this network measure are used to drive the classification of the new instances i.e., a new instance will be classified in one of those components (class), in which their component efficiency measures are in conformity with the insertion of the new instance.

1) Component Efficiency Measure: Motivated by the concept of efficiency of a network [32], the component efficiency measure was proposed in [23] and it quantifies the average efficiency of the component in sending information between its vertices, i.e., it measures how efficiently each component exchanges information internally.

Initially, suppose a vertex i in a component α. The average efficiency of i is given by:

$$E_i^{(\alpha)} = \frac{1}{V_i} \sum_{t \in \Lambda_i} q_{it}, \tag{12}$$

where V_i denotes the number of links from i, Λ_i represents the vertex that receives links from i and q_{it} is related to the local geodesic distance (similarity) between i and t.

The efficiency of a component α is the average of the local efficiency of the nodes that belong to α. So, we have:

$$E^{(\alpha)} = \frac{1}{V_\alpha} \sum_{i=1}^{V_\alpha} E_i^{(\alpha)}, \tag{13}$$

where V_α is the number of vertices in component α.

2) Filtering Components: Suppose a new instance y will be classified. Differently from other high level approaches, the proposed technique employs the component efficiency measure to determine the components which their efficiency are into to the minimal local efficiency of y. This information is important especially because it considers the local features of the components and establishes a heuristic that excludes components that are not in conformity with the insertion of y into them.

In a more formal definition, let us consider a component α and a set L related to the components in which the variations of complex network measures will be computed. For each new instance y, L_y is given by:

$$L_y \Leftarrow L_y \cup \{\alpha \mid \min e_y^{(\alpha)} \leq E^{(\alpha)}\}, \tag{14}$$

where $e_y^{(\alpha)}$ denotes the local efficiency of y to each vertex that belongs to component α and $E^{(\alpha)}$ is the component efficiency of α.

The next phase is the insertion of y in each $\alpha \in L_y$. According to our technique, y makes connections with each vertex $i \in \alpha$ following the equation given by:

$$\alpha_y \Leftarrow \alpha \cup \{i \mid e_y^{(i)} \leq E^{(\alpha)}\}, \tag{15}$$

where α_y includes component α and the connections between y and their vertices, $e_y^{(i)}$ is the local efficiency in exchanging information between y and i, and $e_y^{(i)} \leq E^{(\alpha)}$ is the condition to be satisfied to assure a link between y and i.

Let us now show the implications from (14) and (15):

- once the component efficiency measure works on the geodesic distance between the vertices, there are some features of the low level algorithm on it;
- by considering the minimal local efficiency of y in a component α, the proposed technique carefully selects a good L_y set, avoiding the exclusion of those components which y is in conformation in the high level classification process;
- the local efficiency of the vertices extends the concept of the constructed network by considering not only the nearest neighbors of y, but also features related to the component efficiency in the exchange of information among the vertices of the component. Intuitively, this is an implicit combination between high level and low level features.

3) Algorithm: A general view about the unified approach proposed in this paper is given by the Alg. 1. The technique receives as input the data set (X), which is divided in training data set (X_{Train}) and test data set (X_{Test}). In the first line, the algorithm uses the k-associated optimal graph to obtain a network from X_{Train}. In this context, G and Λ denote, respectively, the constructed network, where $G = \{V, E\}$ describes the sets of vertices and edges and Λ comprehends all components in G. In the second line, the complex network measures (assortativity, clustering coefficient and average degree in this paper) are computed on each component $\alpha \in \Lambda$, so M describes the results of applying these network measures to the components. There is a loop in the line three from which all test instances ($y \in X_{Test}$) are considered. In the fourth line, the technique employs the component efficiency measure to provide low level and high level information. Consequently, by definitions (14) and (15), G_y denotes each component α_y which is in conformity with the insertion of y, such that $\alpha_y \in G_y$. In the fifth line, the technique computes the complex network measures variations for these components. Thus, ΔG_y denotes the set of network measures variations on each component α_y, such that $\Delta G_y^{\alpha_y}(.) \in \Delta G_y$. Finally, the sixth line provides the classification probabilities of y in each class j. Obviously, the technique classifies y for the class with the highest probability.

Algorithm 1 Unified High Level Approach

```
Require: data set $X$: $X_{Train}$, $X_{Test}$
1: $G$, $\Lambda \Leftarrow$ Construct Network (X$_{Train}$)
2: $M \Leftarrow$ Complex Network Measures ($\Lambda$)
3: for each $y$ in $X_{Test}$ do
4:     $G_y \Leftarrow$ Filter Component Efficiency ($\Lambda$,$y$)
5:     $\Delta G_y \Leftarrow$ Complex Network Measures Variations ($G_y$)
6:     $C_y^j \Leftarrow$ High Level Classification ($\Delta G_y$)
7: end for
```

Note that if $L_y = \emptyset$ in (14) (very unusual situations), the algorithm employs the k_α value associated with each component $\alpha \in \Lambda$ and verifies what the vertices in α are one of the k_α-nearest neighbors of y. If there is at least one vertex that satisfies this condition in component α, then the complex network measures are applied to this component, otherwise α is unconsidered in the classification phase.

III. Experiments and Discussion

This section presents a set of empirical evaluations to analyze the performance of our proposal. It is divided into two subsections: Subsect. III-A discusses the results of artificial data sets, emphasizing some important characteristics of the new high level approach that make its classification more robust than those of another low level algorithms; and Subsect. III-B provides simulations on real-world data sets, showing the proposed technique is applicable to practical situations, especially because it can find competitive results with traditional machine learning techniques and it does not have parameters. Note that the Euclidian distance is used as the similarity measure in all experiments. In addition, our high level technique is referred to in this section as HL.

A. *Artificial Data Sets*

Computer simulations are performed to evaluate the proposed technique on artificial data sets, which are characterized by showing strong patterns. They provide particular situations in which traditional classifiers have trouble to correctly classify the data items in the test set. Moreover, this subsection emphasize the great features of our totally based on complex networks algorithm, which also serves as a tool for better motivating the usage of the new approach.

1) Binary-Class Data Set I: The first experiment is performed with the data set showed in Fig. 1. One can see the green ($\triangle$) class data items exhibit a strong pattern. Since the traditional machine learning techniques are not able to consider the pattern formation of classes, they cannot classify the test instance y (magenta $\Diamond$ vertex) correctly. Moreover, these techniques consider only the physical distance among the data items, which contributes to classify y as belonging to red class ($\circ$). On the other hand, previous high level algorithms perform the classification of y by considering or an ϵ-radius variable [21] or by computing the network measures variations for all graph components [23]. The drawbacks of these algorithms are, respectively, the necessity of parameter selection for the classification stage (few variations in the parameters generate very different results) and a smaller force in terms of classification probabilities of y for each class j, especially because the network measures are computed on all components and for each is associated one probability P_j, such that $\sum_j P_j = 1$.

However, differently from traditional techniques, HL is able to detect the clear pattern in Fig. 1 by following the steps in Alg. 1. Moreover, differently from other high level algorithms, HL does not use any parameter and it produces a stronger representation in terms of classification probabilities of instance y in the components, especially because it provides a heuristic that reduces the number of components to which the network measures are applied.

2) Binary-Class Data Set II: A very interesting data set is showed in Fig. 2. There are data items of the two classes: red ($\circ$) and green ($\triangle$), and a new instance y (magenta $\Diamond$) to be classified. Intuitively, red data items exhibit a straight line pattern. However, traditional techniques classify y in the green class because they perform data classification considering only the physical attributes of the data. Another important characteristic of this data set is that y is very close to the green data items and the red data items are distant among them. Thus, a high value of ϵ will be necessary to allow the high level classification to detect this pattern. Also, the combination of low level and high level algorithms requires a great portion of the high level classification to perform a correct prediction. Again, HL is able to correctly detect this pattern without the use of parameters because its component efficiency measure offers more than low level features: it also provides high level information about the structure and topology of each component.

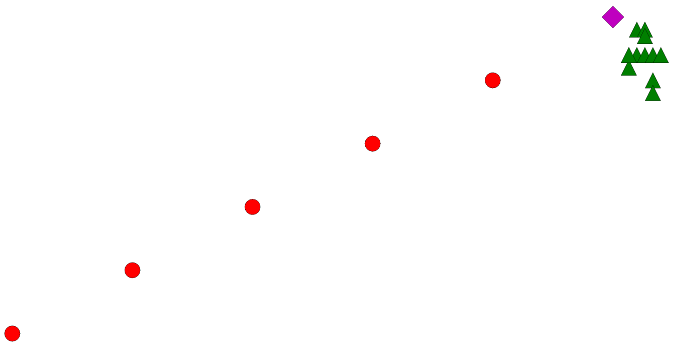

Fig. 2: Data set composed by data items of two classes: red ($\circ$) and green ($\triangle$). Magenta ($\Diamond$) data item needs to be classified. Red data items form a straight line.

3) Multi-Class Data Set: Fig. 3a shows a multi-class data set with five classes. Each class is denoted by a symbol and a color: $\triangle$ (red), $\times$ (green), $\circ$ (blue), $\triangledown$ (yellow) and $\square$ (black). Four classes (red, green, yellow and black) are obtained from Gaussian distributions and the blue class exhibits a strong pattern of circunference. In the figure, there is a new instance to be classified (represented by the symbol $\Diamond$ and the magenta color). At the first moment, this is a very easy classification problem, especially because the classes are linearly separable. Obviously, traditional techniques can predict the label correctly.

However, considering a more applicable scenario, we know data sets not present so representative information as in Fig. 3a. Once the number of instances can influence the representativeness of the information inherent to the data sets, when there is some absence of this information, the pattern formation is not so easy to predict from traditional machine learning techniques, especially because they work only on physical attributes of the data. As a way to show this in a practical situation, some labeled data items are removed from the data set showed in Fig. 3a to obtain the data set exhibited in Fig. 3b.

Despite one can observe the pattern formation of blue class (circunference) also remains strong in Fig. 3b, now traditional algorithms are not able to classify the new instance correctly. On the other hand, our high level technique detects this formation pattern correctly. In a general context, this experiment emphasizes our technique is less sensitive to the absence of information in data sets because it exploits relations among the vertices from distinct visions of the data items (through of complex network measures). Intuitively, this makes the technique more robust to identify formation pattern of the data. In a specific context, our algorithm did not divide the data items in classes, but in graph components, i.e., our technique work on a more operational level, in which it is able to capture important information about the formation pattern of the data sets.

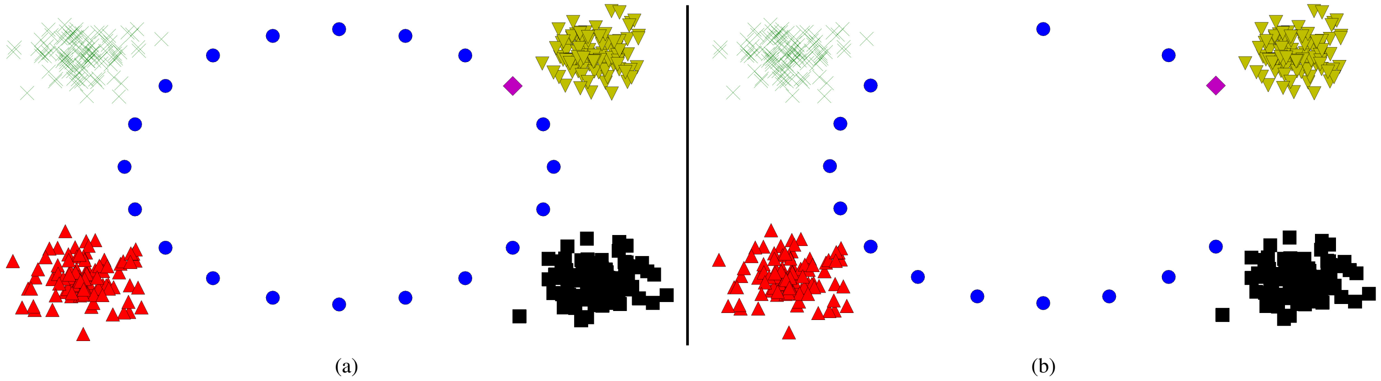

Fig. 3: Data sets composed by data items of five classes (△ - red, × - green, ○ - blue, ▽ - yellow and □ - black). Blue data items present a clear pattern of a circunference. Magenta diamond needs to be classified in the two data sets: (a) a data set with big representativity of the instances; (b) a data set with some absence of information, but also with clear formation pattern.

B. *Real-World Data Sets*

We also have conducted computer simulations on real-world data sets. The objective is to analyze the features of our technique in real data sets and evaluate its performance in comparison to two traditional and widely employed techniques: Decision Tree (DT) and Support Vector Machine (SVM). These traditional techniques are available in the python machine learning module named Scikit Learn [33]. Grid search algorithm was employed to select parameters for those techniques. Following, we give more details about the process of tuning the parameters of the traditional algorithms. About the decision tree algorithm, Scikit Learn provides an optimized version of the CART algorithm. Two parameters are configured in these experiments: the minimum density over the set {0,0.1,0.25,0.5,0.8,1}, which controls a trade-off in an optimization heuristic, and the minimum number of samples required to be at a leaf node, here denoted as m, which is optimized over the set $m \in$ {0,1,2,3,4,5,10,15,20,30,50}. In Support Vector Machine (SVM) simulations, we reduce the search-space for the optimization process by fixing a single well-known kernel, namely Radial Basis Function (RBF) kernel. The stopping criterion for the optimization method is defined as the Karush-Kuhn-Tucker violation to be less than 10^{-3}. For each data set the model selection is performed by considering the kernel parameter $\gamma \in \{2^4, 2^3, \ldots, 2^{-10}\ \}$ and the cost parameter $C\ \in \{2^{12}, 2^{11}, \ldots, 2^{-2}\}$. Our high level approach does not use parameters, so, it not need of a step to tune parameters.

Four real-world data sets have been employed in the experiments: *Ecoli*, *Iris*, *Optical Recognition of Handwritten Digits* and *SPECTFHeart*. These data sets are available in UCI [34] and KEEL [35] repository. A brief introduction about them is given as follows:

- *Ecoli* data set: this data set address the problem of classifying proteins into their various cellular localization sites based on their amino acid sequences. Proteins can be classified in eight classes and the number of attributes, which were calculated from amino acid sequences, is 7[1]. related to distinct sequence names for each instance and. Also, *Ecoli* has 336 instances.
- *Iris* data set: this is a well known data set in the machine learning literature. It provides labeled data items from three different classes of iris plant. There are 4 attributes that are related with the lenght and width of the plant sepal and petal. *Iris* contains 150 instances.
- *Optical Recognition of Handwritten Digits* or *Opt. Digits* data set: this is a data set obtained from the contribution of fourty three people. Preprocessing programs made available by NIST was used to extract normalized bitmaps of handwritten digits from a preprinted form. 32x32 bitmaps are divided into non-overlapping blocks of 4x4 and the number of on pixels are counted in each block. This generates an input matrix of 8x8 where each element is an integer in the range 0..16. *Opt. Digits* has 64 attributes and 5624 instances.
- *SPECTFHeart* data set: the dataset describes diagnosing of cardiac Single Proton Emission Computed Tomography (SPECT) images. Each of the patients is classified into two categories: normal and abnormal. The database of 267 SPECT image sets (patients) was processed to extract features that summarize the original SPECT images. *SPECTFHeart* contains 44 attributes.

Tab. I provides the meta-data of the data sets employed and the results yielded by each algorithm averaged over thirty runs using the stratified 10-fold cross-validation process [36]. In order to analyze statistically the results, we adopted a statistical test that compares multiple classifier over multiple data sets [37]. Firstly, Friedman test is calculated to check whether the performance of the classifiers are significantly different. Using a significance level of 5%, the null hypothesis is rejected. This means that the algorithms under study are not equivalent.

[1] Originally, this data set contains 8 attributes. The “sequence name” attribute was removed because it does not make sense for the classification task

TABLE I: Comparative results provided by HL, DT and SVM on four real-world data sets. "Acc" and "Std." denote, respectively, the average of accuracy and the standard deviation over thirty runs using the stratified 10-fold cross-validation process.

Domain	**Meta-data**			**HL**	**DT**	**SVM**
Name	#Instances	#Attributes	#Classes	Acc. ± Std.	Acc. ± Std.	Acc. ± Std.
Ecoli	336	7	8	85.28 ± 6.69	80.78 ± 5.55	**87.23 ± 5.22**
Iris	150	4	3	**97.13 ± 3.82**	93.60 ± 5.59	96.28 ± 4.02
Opt. Digits	5620	64	10	98.59 ± 0.47	90.27 ± 1.27	**99.26 ± 0.33**
SPECTFHeart	267	44	2	**78.76 ± 3.21**	75.41 ± 6.20	78.01 ± 3.92

Following a post-hoc test, Nemenyi test is employed (also considering a significance level of 5%). The results of this test allow concluding that our technique is highly able to find competitive results in comparison with the SVM algorithm (statistically equivalent). Also, Nemenyi test indicates HL and SVM outperform DT algorithm. These results are satisfactory, especially because HL does not need any low level algorithm to perform a good classification. In addition, this suggests complex network measures are able to provide as high level as low level features. Moreover, our structure based on the component efficiency measure offers extra information about the patterns of the data set which help HL in the classification and which low level algorithms are not able to do. Note also that selection of parameters is not necessary in our approach.

IV. Conclusion

This paper has provided a new approach for high level data classification. Differently from previous works, the proposed technique does not perform a combination between low level and high level algorithms. Instead, low level and high level features are embedded in a unique scheme composed only of complex network measures. The advantages are the reduction in the computational cost because no one low level algorithm is required, the absence of parameters and mainly the exploitation of complex network measures in the whole classification process.

The novelty of our technique is the use of a recently proposed complex network measure, named component efficiency. Basically, it computes how efficiently the vertices in each component exchange information among themselves and it uses these resulting values to drive the classification of the new instances. In other words, a test instance will be classified in one of the components (class), in which their local features are in conformity with its insertion. This process also reduces the computational cost because it establishes a heuristic that excludes of the classification those components that are not in conformity with the insertion of new instance into them.

Empirical experiments and statistical tests have been provided to evaluate the proposed technique. Firstly, artificial data sets have been considered. In these data sets, experiments show traditional machine learning techniques, such as DT and SVM, can not to detect the patterns formation of the data because they perform the classification based only on physical attributes. The simulations emphasize some features of the new approach in relation to previous works about high level classification, such as the independence of parameters and the filtering of the components to the prediction phase which makes the classification more powerful. Secondly, real-world data sets have been analyzed too. Although we have tuned the parameters of DT and SVM algorithms, the high level algorithm totally based on complex networks has been able to find competitive results in comparison with these traditional algorithms. Moreover, our technique has outperformed DT algorithm in all data sets used. Forthcoming works include more detailed investigations about each stage of our technique in order to improve its computational efficiency and classification performance. In addition, more comparisons with another start-of-the-art algorithms will be performed.

Acknowledgment

The authors would like to acknowledge São Paulo State Research Foundation (FAPESP) - grant number 2012/07926-3 - and Brazilian National Council for Scientific and Technological Development (CNPq) for the financial support given to this research.

References

[1] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: Structure and dynamics," *Physics Reports*, vol. 424, no. 4-5, pp. 175 – 308, 2006.

[2] S. Boccaletti, M. Ivanchenko, V. Latora, A. Pluchino, and A. Rapisarda, "Detecting complex network modularity by dynamical clustering," *Physical Review Letters*, vol. 75, p. 045102, 2007.

[3] L. D. F. Costa, O. N. Oliveira, G. Travieso, F. A. Rodrigues, P. R. V. Boas, L. Antiqueira, M. P. Viana, and L. E. C. Da Rocha, "Analyzing and modeling real-world phenomena with complex networks: A survey of applications," *Advances in Physics*, vol. 60, no. 3, p. 103, 2007.

[4] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3–5, pp. 75 – 174, 2010.

[5] M. Newman, *Networks: An Introduction*. New York, NY, USA: Oxford University Press, Inc., 2010.

[6] D. Watts and S. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 393, pp. 440–442, 1998.

[7] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[8] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Rev. Mod. Phys.*, vol. 74, pp. 47–97, Jan 2002.

[9] T. Mitchell, *Machine Learning*, ser. McGraw-Hill series in Computer Science. McGraw-Hill, 1997.

[10] V. Vapnik, *The Nature of Statistical Learning Theory*, 1st ed. Springer-Verlag, 1999.

[11] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[12] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. Wiley India Pvt. Limited, 2007.

[13] E. Alpaydin, *Introduction to Machine Learning*, 2nd ed. The MIT Press, 2009.

[14] *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed.

[15] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," in *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering*. IOS Press, 2007, pp. 3–24.

[16] L. Breiman, *Classification and regression trees*. Chapman & Hall/CRC, 1984.

[17] J. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, 1986.

[18] R. E. Neapolitan, *Learning Bayesian Networks*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2003.

[19] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed., P. Hall, Ed. Englewood Cliffs, NJ: Prentice Hall, 1998.

[20] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[21] T. C. Silva and L. Zhao, "Network-based high level data classification," *IEEE Transactions on Neural Networks*, vol. 23, pp. 954–970, 2012.

[22] M. G. Carneiro, J. L. Rosa, A. A. Lopes, and L. Zhao, "Classificação de alto nível utilizando grafo k-associados ótimo," in *Proceedings of the IV International Workshop on Web and Text Intelligence*. Sociedade Brasileira de Computação, 2012, pp. 1–10.

[23] ——, "Network-based data classification: Combining k-associated optimal graphs and high level prediction," *Journal of the Brazilian Computer Society (submitted)*, pp. 1–12, 2013.

[24] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[25] O. Chapelle, B. Schölkopf, A. Zien *et al.*, *Semi-supervised learning*. MIT press Cambridge, MA:, 2006, vol. 2.

[26] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40 –51, 2007.

[27] X. Zhu, "Semi-supervised learning literature survey," University of WisconsinMadison, Tech. Rep., 2008.

[28] A. A. Lopes, J. R. Bertini, and Z. Liang, "Classification based on the optimal k-associated network," in *Proceedings 1st International Conference on Complex Sciences: Theory and Applications (Complex09)*, 2009, p. 1167–1177.

[29] T. H. Cupertino, M. G. Carneiro, and L. Zhao, "Dimensionality reduction with the k-associated optimal graph applied to image classification," in *Proceedings of the 2013 IEEE International Conference on Imaging Systems and Techniques (IST)*, 2013, pp. 1–6.

[30] R. F. S. Andrade, J. G. V. Miranda, S. T. R. Pinho, and T. P. Lobão, "Characterization of complex networks by higher order neighborhood properties," *European Physical Journal B*, vol. 61, no. 2, p. 28, 2006.

[31] M. E. J. Newman, "Assortative mixing in networks," *Physical Review Letters*, vol. 89, p. 208701, 2002.

[32] V. Latora and M. Marchiori, "Efficient behavior of small-world networks," *Phys. Rev. Lett.*, vol. 87, p. 198701, 2001.

[33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[34] K. Bache and M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[35] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, and S. García, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *Multiple-Valued Logic and Soft Computing*, vol. 17, no. 2-3, pp. 255–287, 2011.

[36] J.-H. Kim, "Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap," *Computational Statistics and Data Analysis*, vol. 53, pp. 3735–3745, 2009.

[37] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.