Towards a High-Level Multi-Label Classification from Complex Networks

Vinícius H. Resende and Murillo G. Carneiro
Faculty of Computing
Federal University of Uberlândia
Uberlândia, Brazil
{viniciusresende, mgcarneiro}@ufu.br

***Abstract*—Multi-label learning aims to solve problems in which data items can have multiple class labels assigned simultaneously, e.g., text categorization, image annotation, medical diagnosis, etc. However, as most of multi-label techniques are derived from the single-label ones, existing techniques perform the multi-label classification only based on the physical features of the data (e.g., distance, similarity or distribution), ignoring the semantic meaning of the data, such as the formation pattern. Inspired by recent advances in the use of complex networks for single-label learning, this exploratory work aims to investigate a multi-label solution able to combine existing multi-label classifiers with a high-level classifier based on complex networks measures, aiming to present a new concept of multi-label classification that, besides the physical attributes, also analyzes the topological structure of the data. Experimental results considering both artificial and real-world data sets emphasize respectively the salient features of our technique in comparison to the traditional ones and its potential to improve the predictive performance of those techniques, especially in data sets characterized by higher cardinality and density of labels, which often denote more difficult scenarios to multi-label learning.**

***Index Terms*—Multi-Label Learning, High-Level Classification, Complex Networks, Machine Learning, Complex Network Measures.**

I. Introduction

Supervised learning is one of the main machine learning paradigms, in which data items (or objects) are denoted beyond a set of features, by known information referred as label (or class). The most common supervised learning task is data classification, which aims to learn a function from previously labeled objects in order to predict the label of each new (unknown) object. An example of data classification is the spam detection problem, where each known data item (email) is denoted by a set of features (e.g., sender, subject, words in the email, etc) and a single label (spam or not spam). This way, conventional data classification assumes that each object can be associated with only one class. However, such an assumption does not hold for problems in which the objects can have simultaneously multiple labels [1]. For example: in sentiment classification, the same tweet can represent multiple sentiments, such as worry, surprise and fear [2]; in text categorization, a newspaper article can be tagged as People and Economics [3]; and in image annotation, one image can be tagged with a set of multiple words indicating its contents, such as airplane, sky and grass [4]. Such problems are addressed by the multi-label learning task which assumes that objects can belong to one or more classes.

Multi-label classification has attracted a lot of attention in recent years as it is a very challenging task from both theoretical and practical perspectives. From the theoretical viewpoint, the overwhelming size of output space which grows exponentially as the number of class labels increases makes the learning task much more difficult to solve than conventional classification. From the practical one, the increasing range of complex applications makes the multi-label techniques prominent tools to deal with such problems.

Multi-label techniques can be divided in two major approaches: problem transformation and problem adaptation, with the former aiming to fit data to algorithm and the latter to fit algorithm to data [1].

The most common approach to deal with multi-label classification problems transforms it into another well-established learning scenarios [1]. After that, any conventional classification technique can be used, e.g., Support Vector Machines (SVM), Naive Bayes (NB), etc. Examples of techniques that adopt such an approach include Binary Relevance (BR) [5] and Classifier Chain (CC) [6]. BR is the most popular algorithm for multi-label learning and works by transforming the multi-label problem into independent binary classification problems, where each one of them corresponds to a possible label in the label space. The CC in turn, transform the multi-label problem into a chain of binary classification problems, where subsequent binary classifiers in the chain are built upon the predictions of preceding ones.

Alternatively, some works in literature also adapted conventional learning algorithms to treat the multi-label problem directly. One of the most representative algorithms in this category is Multi-Label k-Nearest Neighbor (ML-kNN) [7], which is an adaptation of lazy learning techniques. The basic idea is to adapt the kNN to handle the multi-label problem. In addition, it also employs Bayesian inference to select assigned labels. Other adapted algorithms include Ranking Support Vector Machines (Rank-SVM) [8] and Collective Multi-Label classifier (CML) [9].

Another categorization of multi-label algorithms takes into account the exploitation of correlations (or dependency) among labels [1]. In this way, the algorithms can be roughly categorized in first-order, second-order and high-order strate-

gies. Algorithms of first-order, such as BR and ML-kNN, decomposes the multi-label problems into independent binary classification problems, ignoring any correlation among labels. Algorithms of second-order, such as Rank-SVM and CML, considers co-occurrence of labels by considering pairwise relations between them (e.g., ranking labels in terms of relevance) or by analyzing the interaction between pairs of labels. Algorithms of high-order, such as CC, considers relations among labels such as taking labels' influences on each label or also addressing connections among random subsets of labels.

Despite the great advance in the last years, a drawback which has been barely investigated is that most of the multi-label learning techniques performs classification considering only the physical features of the input data (e.g., similarity, distance or distribution). This happens because these techniques are designed essentially from the traditional single-label classification ones. Although some of them are also able to analyze dependencies among the labels, this does not necessarily contribute to the detection of the semantic meaning of the data patterns as they are often inherently represented into complex relationships among the features. Figure 1 denotes a misleading problem for most of the current multi-label techniques, which are unable to detect both the straight line (Green/○ markers) and the kind of spherical shape (Red/△) patterns. Indeed, as we have shown in our experiments, they are more propense to classify each test item (denoted as Black/□ markers) only in the Red/△ class, completely ignoring the straight line pattern.

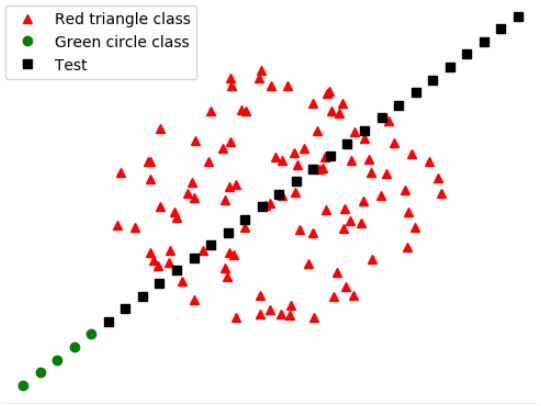


Fig. 1: A toy data set which emphasizes the drawbacks of current multi-label techniques and also the salient features of our technique.

Indeed, the difficult of most traditional classification techniques in the detection of high level patterns (e.g., formation pattern) has been pointed out and discussed by recent works in literature [10]–[15], which have designed techniques able to consider both physical and topological features of the input data in order to overcome such a problem. In common, all these techniques convert vector-based data into a network in order to exploit properties from complex network theory. For example, [10] proposed a high-level framework which combines the associations produced by low-level and high-level techniques. The low-level, which can be any traditional technique (e.g., SVM, NB, kNN, decision tree, etc), captures physical features of the data. The high-level, which is given by a classifier via complex network measures, captures topological features of the data by verifying its pattern conformation. In a few words, an unknown object is inserted into the network components (each one denoting a class label) and is classified in the class which the network structure suffers the smallest variation after its insertion. Such a framework has been one of the main inspirations for our work.

Motivated by those recent works for single-label classification, this paper aims to model a high-level technique able to combine associations produced by existing multi-label classification algorithms with those produced by complex network measures in order to consider besides the physical attributes, the topological structure of the data in the multi-label learning context. Thus, our hypothesis states that such a technique can improve the predictive performance of widely used multi-label algorithms. In order to test this, we selected relevant techniques of each one of the multi-label categorizations discussed before, namely BR (problem transformation and first-order), ML-kNN (problem adaptation and first-order) and CC (problem transformation and high-order). To evaluate our work, experiments have been conducted on artificial and real-world data sets, which confirmed the limitations we pointed out about current multi-label learning as well as revealed a promising scenario to continue our investigation about high-level multi-label learning.

The remainder of this paper is organized as follows. Section II presents a formal definition of the problem as well as a detailed description about the proposed technique. Section III presents results on a illustrative data set which also serve as a motivation for our work. In Section IV we describe the experiments performed, discuss the results of the techniques under comparison and also the analysis about the parameters influence. Finally, Section V concludes the paper.

II. Model Description

In both single-label and multi-label learning, most of the existing algorithms perform classification considering only the physical attributes of the data, ignoring the semantic meaning of the data, such as the formation pattern. By the contrary, it has been shown in the single-label learning literature that one way to capture such a semantic meaning is through of the usage of complex network models [10], [13], [14]. Based on the high-level single-label technique presented in [10], we propose in this paper a high-level technique able to perform multi-label learning. To the best of our knowledge, this is the first attempt to address such a task by using complex networks. Thus, the basic idea here is to combine associations produced by existing multi-label techniques with those provided by a high-level technique based on network measures in a way that the resulting multi-label learning technique is able to consider not only the physical attributes, but also the data topological structure.

The multi-label classification problem addressed here can be divided in two phases: training and test. In the training phase, the proposed algorithm receives as input a given training set denoted by $\mathcal{X} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$, where each data item is represented by the tuple $(\mathbf{x}_i, \mathbf{y}_i)$, with $\mathbf{x}_i = \{x_1, \ldots, x_d\}$ denoting a d-dimensional vector of features and $\mathbf{y}_i = \{y_i^{(1)}, \ldots, y_i^{(L)}\}$ the output domain of possible labels $\mathcal{L} = \{1, \ldots, L\}$ in which a given class label l can be assigned (if $y_i^{(l)} = 1$) or not (if $y_i^{(l)} = 0$) to x_i. The objective here is to learn a multi-label classifier function $f\colon \mathcal{X} \rightarrow 2^{\mathcal{L}}$. In the test phase, the multi-label classifier $f(\cdot)$ is used to predict the labels of new test items $(\mathbf{x}, ?)$, such that $f(\mathbf{x}) \subseteq \mathcal{L}$.

Regarding the proposed technique, the training phase has two major steps which are the construction of a graph for each class label from the input data $\mathcal{X}$ and the calculation of the network measures for each one of the graphs. In the test phase, three major steps are followed: each test item is virtually inserted into the graphs to recalculate the network measures in order to obtain their variations; such variations are then employed to produce the classification probabilities of the test item for each class; and such probabilities are after combined with the classification probabilities provided by traditional multi-label techniques. In the next sections we present a detailed description about each one of these phases in our technique.

A. Training Phase

In the training phase, our proposed technique has two major phases which are the graph construction and the calculation of the complex network measures, which are described in the following.

1) Graph Construction: In order to exploit topological properties of the data, they must be represented as a network. Despite most data sets are available in the format of vector of attributes, the literature contains some methods to build up graphs from such a kind of data. The most popular techniques are variants of the nearest neighbor approach, such as the *k-nearest neighbor graph* which connects a given node to its k nearest neighbors and the *ϵ-neighborhood graph* which connects a given node to all other nodes that distances are less than a predefined value ϵ [16]. Another approach is the *b-matching* method that removes edges of the graph in such a way that all vertices have the same degree [17], [18] or yet the usage of particle swarm optimization to build up an optimized graph regarding a given learning task [15].

Different from related works applied to the single-label classification [10], [13], where a unique graph is generated and classes are denoted by network components, our formulation provides a set of graphs $\mathcal{G} = \{\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(L)}\}$ in order to model an inductive multi-label learning setting . Formally, for each label $l \in \mathcal{L}$ is associated a graph $\mathcal{G}^{(l)} = \{\mathcal{V}^{(l)}, \mathcal{E}^{(l)}\}$, where $\mathcal{V}^{(l)}$ and $\mathcal{E}^{(l)}$ denotes the set of nodes and edges, respectively. Each node $v_i \in \mathcal{V}^{(l)}$ represents a data item $x_i \in \mathcal{X}$ which is associated with the label l, i.e., $y_i^{(l)} = 1$. Each edge $e_{ij} \in \mathcal{E}^{(l)}$ represents a link between nodes v_i and v_j. In this paper we have proposed a multi-label variation of the k-nearest neighbors graph in which a connection $e_{ij} \in \mathcal{E}^{(l)}$ exists if x_j is among the nearest neighbors of x_i, $y_i^{(l)} = y_j^{(l)} = 1$ and $\sum_{l=1}^{\mathcal{L}} outdegree(v_i \in \mathcal{V}^{(l)}) \leq k$. In this variation, any data item that belongs to multiple class labels can be inserted into multiple graphs. Notice that the vertices do not necessarily will have the same degree in those graphs as the proposed nearest neighbors graph is asymmetrical, i.e, a vertex u may be in the k neighbors of a vertex v, but the opposite may not happen. After the directed graphs are formed then they are converted to undirected ones so that we can use the complex networks metrics.

2) Complex Network Measures: The study of complex networks has attracted considerable attention in recent years in solving problems of social, biological and communication systems [19]. Complex networks can be understood as networks that have a non-trivial topological structure, which means that their training bias does not follow a specific or totally random criterion [13]. In this study, we employ the following three network measures in order to extract topological properties of the data.

Assortativity. This measure determines how much the vertices tend to connect in an assortative way. The measure can assume values between $[-1, 1]$, so that positive values indicate that pairs of directly connected vertices are more likely to behave in the same way, whereas values negatives indicate a greater probability of connected vertices having different behaviors [20]. Be E the number of edges in the network and i_u, k_u the degrees of the vertices i and k which compose an edge u, the assortativity can be calculated by:

$$r = \frac{E^{-1}\sum_u i_u k_u - [E^{-1}\sum_u \frac{1}{2}(i_u + k_u)]^2}{E^{-1}\sum_u \frac{1}{2}(i_u^2 + k_u^2) - [E^{-1}\sum_u \frac{1}{2}(i_u + k_u)]^2} \quad (1)$$

Average Degree. This is one of the simplest network measures, which quantifies the average degree of the network.

$$AD = \frac{1}{N}\sum_{i=1}^{N} k_i \quad (2)$$

Clustering Coefficient. Clustering coefficient or transitivity quantifies how much the vertices tend to group together. The clustering coefficient of a vertex measures how close it is to a clique. This measure can be obtained by:

$$CC_i = \frac{|e_{us}|}{k_i(k_i - 1)} \ , \quad (3)$$

$|e_{us}|$ represents how many clique of three vertices is formed through vertex i, which means the number of connections shared by direct neighbors of the vertex i. To form one clique, i should be connected to $\{u, s\}$ and u need also to be connected to s. k_i is the degree of the vertex i. The average clustering coefficient of the network can be obtained by:

$$CC = \frac{1}{N}\sum_{i=1}^{N} CC_i \ . \quad (4)$$

B. Test Phase

From the calculations of the network measures on the graph generated in the training phase, the next steps are: the insertion of each test item into a corresponding graph by using the same k-nearest neighbors strategy adopted in the training phase; the calculation of the high-level associations; and the combination of both traditional and high-level associations.

1) Variation of the Network Measures: Given a test item $\mathbf{x}$ which is virtually inserted into a graph $\mathcal{G}^{(l)}$ (denoting a class label l). The variation of the network measures in such a graph is given by:

$$f_{\mathbf{x}}^{(l)} = G_{\mathbf{x}}^{(l)}(u)\, p^{(l)}, \tag{5}$$

where $G_{\mathbf{x}}^{(l)}$ represents the variation of the test item $\mathbf{x}$ using the measure u in the graph and $p^{(l)}$ represents the proportion of training objects belonging to l, a strategy for dealing with imbalanced data sets.

2) High-Level Term: After the insertion of a test instance and the calculation of its variation in the corresponding graph, the high-level probabilities can be obtained. If the test item causes great variation, it probably is not in conformation with that class pattern, i.e., it does not belong to that class. Otherwise, it probably is compliant with that network pattern and may be associated with the label in question. The formulation of the high-level term is given by:

$$\mathcal{H}_{\mathbf{x}}^{(l)} = \sum_{u=1}^{Z} \delta(u)[1 - f_{\mathbf{x}}^{(l)}(u)] \ , \tag{6}$$

where $\delta \in [0,1]$, $\sum_u^Z \delta(u) = 1$ represents a weight for each network measure, u represents the chosen network measures for the pattern compliance analysis, and Z represents the number of measures to combine.

3) Combination of Classifiers: At the end, the high-level multi-label classifier performs the combination between low and high-level associations, i.e.:

$$\mathcal{M}_{\mathbf{x}}^{(l)} = \lambda \mathcal{H}_{\mathbf{x}}^{(l)} + (1-\lambda)\mathcal{C}_{\mathbf{x}}^{(l)} \tag{7}$$

where $\mathcal{M}_{\mathbf{x}}^{(l)}$ is the value generated by the junction of the probabilities of an object $\mathbf{x}$ belonging to the label l given by a traditional multi-label classifier, denoted by $\mathcal{C}_{\mathbf{x}}^{(l)}$ and by the multi-label combination of network measure variations, denoted for $\mathcal{H}_{\mathbf{x}}^{(l)}$.

As $\mathcal{M}$ provides the probabilities associated with each label $l \in \mathcal{L}$, a threshold needs to be achieved in order to classify (or not) the test item $\mathbf{x}$ in the class label l. Therefore, the labels to be associated with $\mathbf{x}$ must respect the following criterion:

$$y_{\mathbf{x}}^{(l)} = \begin{cases} 1 & \text{if } \mathcal{M}_{\mathbf{x}}^{(l)} \geq \tau \\ 0 & \text{otherwise.} \end{cases}$$

C. Algorithm and Complexity

The Alg. 1 describes the main steps of the high-level multi-label technique proposed here. In the following we discuss the time complexity inherent to each of those steps. For sake of clarity, we consider the number of nodes $\mid \sum_{l=1}^{\mathcal{L}} \mathcal{V}^{(l)} \mid = \mathcal{O}(nc)$ and the number of edges $\mid \sum_{l=1}^{\mathcal{L}} \mathcal{E}^{(l)} \mid = \mathcal{O}(nc)$ as the graphs are sparse, i.e., ($k \ll n$). Notice that c denotes the cardinality of the data set in terms of labels per objects.

Algorithm 1: PROPOSED TECHNIQUE.

1 **Require:** k, $\mathcal{X}$, $\mathbf{x}$, λ, δ, τ
2 Build up the graphs from $\mathcal{X}$ (refer to SubSect. II-A1)
3 Compute Eqs. 1-4 for the graphs
4 Insert a test item $\mathbf{x}$ in the graphs of its k nearest neigh.
5 Compute Eqs. 5-7
6 Classify $\mathbf{x}$ in any label class l which $\mathcal{M}_{\mathbf{x}}^{l} \geq \tau$

1) The time complexity related to the graph construction lies on $\mathcal{O}(n^2)$ as the euclidean distance is computed between each pair of data items before the nearest neighbor search.
2) The complexity order of the network measures lies on $\mathcal{O}(nc)$ for assortativity, $\mathcal{O}(ncd^2)$ for clustering coefficient, and $\mathcal{O}(nc)$ for the average degree with d denoting the mean of the average degrees. Taking the highest, we have $\mathcal{O}(ncd^2)$.
3) The time complexity of the insertion step lies on $\mathcal{O}(nlog(n))$ as we need to find the k nearest neighbors of the test object.
4) The time complexity associated to calculate the variation of the network measures lies on $\mathcal{O}(cd^2)$ since we can recalculate the measures only for the neighbors of the test object.
5) Finally, the complexity order of the high-level classifier is given by $\mathcal{O}(n^2 + ncd^2 + n\ log(n) + cd^2)$. As usually we have $c \ll n$ and $d \ll n$ (sparse graphs), taking the highest order term results in $\mathcal{O}(n^2)$. However, this time complexity can be reduced by adopting any improvement of the nearest neighbor methods, such as tree-based methods. On the other hand, the time complexity of the low-level classifier depends on the algorithm chosen.

III. RESULTS ON A TOY DATA SET

Here we present an illustrative experiment which emphasizes the salient features of the proposed technique over the traditional multi-label ones. Taking the data set denoted by Fig. 1, we have both classes Green/$\circ$ and Red/$\triangle$ corresponding to clear and distinct patterns. The former denotes a straight line pattern while the latter denotes some kind of spherical shape. The data set has also test items, denoted by Black/$\square$ markers, which seem the continuation of the line pattern, although some of them could also be part of the spherical pattern. Traditional classification techniques, such as decision tree, neural networks, kNN and SVM, are much more propense to classify these test items into the Red/$\triangle$ class as they consider only the physical features of the data, such as distance or distribution, i.e., they have trouble to classify the test items according to their semantic meaning (e.g., formation pattern).

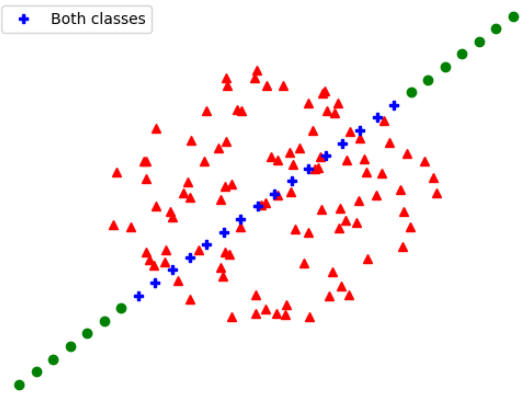
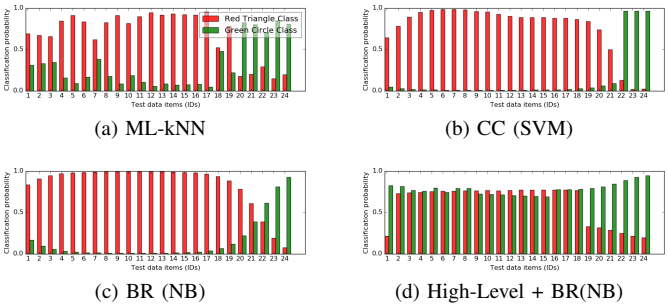


Fig. 2: The correct class of each toy sample.

Despite the multi-label learning task enables to consider the test items in both classes, most techniques designed for such a task is strongly based on the single-label classification ones, which means they should share similar advantages but also similar limitations. In order to demonstrate this, we performed a simple experiment with the illustrative data set considering the three multi-label techniques under study: ML-kNN, CC (SVM) and BR (NB), besides the proposed high-level multi-label technique which is combined with BR (NB) here. In such an experiment, the test data items in Fig. 1 needs to be classified one by one, from left to right, being that the test data item 1 is the leftmost square. To maintain the data distributions during the simulations, whenever a data item is classified, it is incorporated to the training set with the corresponding label(s) and the training and test phases start again. For sake of clarity, Fig. 2 shows the correct class of each toy data set sample used in our experiments. For our analysis we also defined 0.5 as the threshold value τ in order to get a class label from the classification probabilities.

The results obtained by the techniques are shown by Fig. 3 which reveals the difficult of the multi-label techniques in the detection of the related patterns. As shown in Fig. 3(a), ML-kNN is able to classify only the last five test items (rightmost) according to the straight line pattern (Green/$\circ$ class); every other test item was assigned to the spherical one (Red/$\triangle$ class), while no object was associated with both classes. Similarly, Figs. 3(b) and 3(c) shows that CC (SVM) and BR (NB) are able to associate only the three rightmost test items to the straight line pattern, again with no object getting both labels. These results are easy to explain as ML-kNN, CC (SVM) and BR (NB) do not consider topological relations among the data items, only their physical features.

By the contrary, the results of the proposed technique, which are shown in Fig. 3(d), uncover the inherent patterns related to both classes. In this experiment, we have adopted $\lambda = 0.8$, which means that the high-level term had a larger contribution in the final probabilities. By analyzing the figure, one can see that all test items were detected as belonging to the straight line pattern. Interestingly, test items which were in conformation with the spherical pattern were also assigned to that label too. Moreover, such results give evidences about the drawbacks related to the traditional multi-label techniques and establish an important motivation for the design and development of new algorithms to the multi-label learning, including those based on complex network theory. Notice that we do not show the combination of the high-level classifier with other low-level techniques for sake of space. In addition, as those techniques also considers only the physical features of the data, such combinations give us similar results to those presented in the figure.



(a) ML-kNN (b) CC (SVM)

(c) BR (NB) (d) High-Level + BR(NB)

Fig. 3: Classification probabilities of traditional and high-level multi-label techniques for the toy data set presented in Fig. 1.

Notice that other than the toy example, the test object is not inserted as a training instance after being classified, implying that the order in which the objects are tested is not relevant to our algorithm.

IV. Results on Real-World Data Sets

In this section we present the experimental results obtained by the proposed technique, which combines conventional multi-label techniques and complex network measures. The section is organized as follows: Subsect. IV-A describes in detail our experimental setup; Subsect. IV-B presents results in real-world data sets; and Subsect. IV-C discuss the influence of the parameters in the proposed technique.

A. Experimental Setup

1) Data sets: In the following we briefly describe the four data sets used in our experiments. The selection was made to encompass diversity on data domains. A numerical summary about the data sets in terms of instances, features and labels is also presented in Table I. The table also presents the cardinality and density of each data set, which are measures that say about "how multi-label a problem is" and that calculate respectively the average number of labels by examples and the average number of labels by examples divided by the total amount of labels each data set has. The division of the data in training and test set, which follows [21], is also presented in the table. In the following we give a brief description about each one of the data sets adopted in this work:

- In the Birds data set it is addressed the following problem: given a recording of an audio, say all species of birds are present there [22].

TABLE I: Brief description of the real-world data sets.

Data set	Domain	#Instances	#Features	#Labels	Cardinality	Density	#Train	#Test
Birds	Audio	645	258	19	1.014	0.053	322	323
Emotions	Music	593	72	6	1.869	0.311	391	202
Scene	Image	2407	294	6	1.074	0.179	1211	1196
Yeast	Biology	2417	103	14	4.237	0.303	1500	917

- The Emotions data set comprises the following problem: given the timbre and the rhythm of a music, what will feel who listen it? The possible labels (feelings) available at the data set are: surprise, happiness, calm, remain quiet, sadness and angry [23].
- The Scene data set has 294 attributes describing a variety of images. The problem given by this data set is: taking an image, what content it shows? Examples of labels are beach, mountain, field and sunset [24].
- Yeast is a data set related to gene expression data and phylogenetic profiles produced via gene microarrays. The problem to be addressed in this data set is: given the genetic expression and phylogenetic information of a yeast, what are its genetic functional classes? The data set has 14 labels for classification [8].

2) Conventional techniques and their parameters: The conventional multi-label classifiers selected for this study are Binary Relevance, Classifier Chain and ML-kNN. With such a selection we cover techniques from both categorization discussed in the Introduction section, namely problem transformation (BR and CC) and problem adaptation (ML-kNN), and first-order (BR and ML-kNN) and high-order (CC) strategies. As BR and CC transforms a multi-label problem in single-label problems, they also requires a base classifier, which often comes from conventional classification literature. In this work, two base classifiers have been evaluated: Naive Bayes which is a simple and widely used technique in multi-label learning; and Support Vector Machine which has been a state-of-the-art classification technique for many domains.

Regarding the parameters, we adopted the values recommended by the authors in their papers. Thus, ML-kNN parameters have been defined as $k = 10$ (number of neighbors) and $s = 1.0$ (smoothing parameter). BR has no parameter besides the base classifier as well as CC once we defined the order of the chain as the order of the labels. About the base classifiers, we assumed the likelihood to be Gaussian in NB; and defined the radial basis function as the kernel in SVM. We also evaluated a small range of values for the kernel (μ) and cost ($\mathcal{C}$) parameters in SVM, but as the results were closely, we kept the standard values of [21].

3) Proposed technique parameters: About our high-level technique, we have considered the variations of three parameters in our experiments. The first parameter which is inherent to $\mathcal{H}$ is the network measures weight δ. As we have selected three network measures for our experiments, namely assortativity, clustering coefficient and average degree, δ has been optimized over the following range $\{(0.1, 0.1, 0.8), (0.1, 0.2, 0.7), \ldots, (0.8, 0.1, 0.1)\}$, which assures that $\sum_{u=1}^{3} \delta(u) = 1$. The second parameter which is related to the convex combination of both traditional multi-label and high-level classifiers, denoted by λ in (6), is optimized over the range $\{0, 0.1, \ldots, 1.0\}$, where $\lambda = 0.8$ means a contribution of 80% of the high-level term in the final prediction. The last parameter named τ is the threshold which a final prediction must achieve in order to get the test data item classified in the label class under evaluation. Such a parameter is also optimized over the range $\{0, 0.1, \ldots, 1.0\}$.

4) Distance and Evaluation Metrics: The Euclidean distance has been adopted as the dissimilarity measure in our simulations as well as the accuracy has been adopted as the evaluation measure.

B. Results

Table II shows the results of conventional multi-label classifiers $\mathcal{C}$ and their respective combination with the high-level classifier via complex network measures, denoted by $\mathcal{M}$. Unlike the subset accuracy, where only objects that had their set of labels perfectly predicted are taken into consideration, the accuracy measure used in this paper takes into account each label that was correctly predicted, which provide a more accurate representation of the model's prediction performance. In the following we analyze both the performance of the traditional multi-label techniques and of the high-level technique.

1) Performance of the traditional techniques: One can see in Table II that the best results varies from technique to technique according to the data sets, which emphasizes the diversity of our selection in terms of data sets and techniques. For example: ML-kNN obtained the best results for Scene and Yeast data sets, but the worse result for Emotions; Binary Relevance and Classifier Chain strategies using Support Vector Machine as the base classifier achieved the best result for Birds, followed closely by ML-kNN; and Classifier Chain strategy using Naive Bayes as the base classifier returned the best result for Emotions, although also the worse results for Yeast data set.

2) Performance of the high-level technique: The results in Table II showed that the high-level multi-label technique contributed to improve the performance of the traditional multi-label ones. Analyzing each data set in separate, one can see that such a combination in Emotions resulted in better results for all five algorithms; the same happened in Birds (although with very slight improvement), where a high value for the τ parameter improved the results of the conventional algorithms; in Scene and Yeast data sets, the proposed technique was not able to improve considerably the results of ML-kNN,

although have achieved this for the other four techniques under evaluation.

TABLE II: Accuracy values of $\mathcal{C}$ and $\mathcal{M}$ for each dataset.

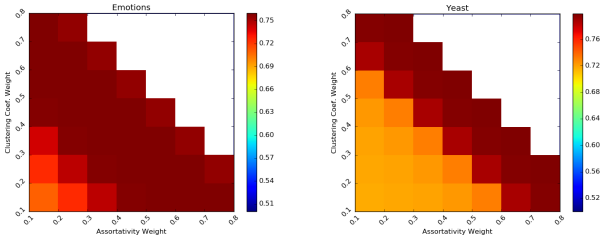
Alg.	Birds		Emotions		Scene		Yeast	
	$\mathcal{C}$	$\mathcal{M}$	$\mathcal{C}$	$\mathcal{M}$	$\mathcal{C}$	$\mathcal{M}$	$\mathcal{C}$	$\mathcal{M}$
ML-kNN	94.6	**94.9**	69.6	72.1	90.9	**91.0**	79.0	79.1
BR (SVM)	94.8	**94.9**	73.5	74.3	85.9	90.1	76.7	**79.8**
CC (SVM)	94.8	**94.9**	73.5	74.5	86.3	88.8	76.7	79.1
BR (GNB)	74.1	**94.9**	72.7	74.2	75.6	85.7	69.9	72.4
CC (GNB)	75.3	**94.9**	73.9	**75.9**	79.3	86.1	68.6	70.5

3) Complexity of the data sets: Another point for our analysis is the complexity inherent to each data set. Despite Emotions has 6 labels (see Table I), the predictive accuracy obtained in such a data set is usually smaller than that obtained for the other data sets (with 14 or 19 labels, for example). This can be partially explainable by two multi-label metrics, cardinality and density, which informs that such a data set has more occurrences of multi-label items than Birds and Scene, for example. Yeast data set has yet a higher cardinality value than Emotions. Interestingly, the high-level multi-label technique achieved better improvement in both data sets. Therefore, such a result may suggests that the classification via network measures can be a promising technique to achieve better performance in such a difficult scenario.

C. Parameter Analysis

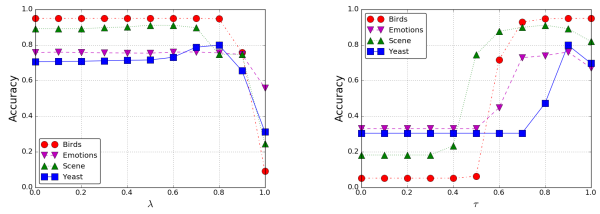
Now we move on to analyze the influence of each parameter of the high-level multi-label technique. The first parameter we analyze is the network measures weight δ. For such an analysis we take the best result for each data set (see Table II) and vary the weights of the network measures in order to evaluate the predictive performance of the technique. Notice that we do not change λ and τ values in such an analysis. Figure 4 shows the heatmaps of each network measure for Emotions and Yeast data sets. To interpret such heatmaps, it is necessary to consider the complement of the sum between the assortativity (axis x) and clustering coefficient (axis y) weights as the average degree weight. One can see that assortativity and clustering coefficient have equivalent contribution in both data sets with larger values of both network measures providing better results than larger values of average degree. Indeed, as the weight of average degree increases worst are the predictive results. Our analysis with Birds and Scene data sets are not exhibited due to the small diference in terms of performance when varying the network measures weights, i.e., the three network measures are closely equivalent in such data sets.

The second parameter analyzed here is the λ, which denotes the linear convex combination among the associations produced by both classifiers. Again, we take the best result for each data set in Table II and vary only the values of λ in order to evaluate the predictive performance of the technique. The results are shown by Fig. 5a which demonstrates that λ have caused insignificant improvement for Birds and Emotions, small improvement for Scene and considerable improvement for the Yeast data set. Notice that results of $\mathcal{M}$ with $\lambda = 0$ (i.e., without any contribution of the high-level classifier) can be different from results of $\mathcal{C}$ (low-level classifier) as the post-processing of $\mathcal{M}$ includes an additional step related to the application of a threshold τ. Another point one can observe in this figure is that $\lambda = 1.0$ (only the high-level term is adopted) provides the worst predictive results, which can be easily explained: despite network measures can detect topological properties of the data, the information from the traditional techniques continues very important, as they detect physical properties of the data related to distance or distribution, for example.



(a) Emotions (b) Yeast

Fig. 4: Analysis of the network measures weight parameter δ .



(a) λ parameter (b) τ parameter

Fig. 5: Analysis of (a) the linear convex combination parameter λ; and (b) the threshold parameter τ.

The threshold τ is the last parameter we evaluate here. In order to conduct its analysis we performed the same preparation as for δ and λ. Figure 5b presents the results of the variation of such a parameter. One can see that low values of such a parameter results in poor performance. Otherwise, the figure clearly suggests that best values for τ were 0.8 and 0.9.

V. Conclusions

In this paper we extend the high-level framework to the multi-label learning task aiming to take into account both physical and topological features of the input data in the multi-label classification process. In order to accomplish that, we presented a new formulation for the high-level framework which deal with the particularities of the multi-label task. The high-level term (topological features) is provided by generating probabilities from the variation of a set of complex network measures, while the low-level term (physical features) can be provided by any traditional multi-label technique.

Experiments were performed in artificial and real-world data sets. The results in the former emphasized salient features of our approach in comparison to the traditional ones, such as the ability to detect the multiple formation patterns of the data. The results on real-world data sets showed our proposed technique has potential to improve the predictive performance of most of the techniques under comparison, especially in data sets characterized by higher cardinality and density values, which often denote more difficult scenarios to multi-label learning. This is a promising result that allows us to consider further investigations of the use of complex networks for multi-label classification.

Forthcoming works include: the evaluation of other network measures into the high-level technique, as we noticed that the average degree has not contributed to improve the predictive performance; the evaluation of more data sets in order to better analyze the efficiency of our technique as the cardinality and density values increase; the investigation of efficient graph construction methods in order to better detect the inherent characteristics related to both features and labels dependency; and also the usage of other multi-label metrics of performance beyond the accuracy. We also intend to exploit another concepts of complex networks besides of the high-level one, which can be related, for example, to characterization of importance [13], ease of access [12] or community detection [25].

ACKNOWLEDGMENT

Authors thank the financial support given by the Brazilian National Council for Scientific and Technological Development - CNPq (grant number 439556/2018-0). Authors also acknowledge support from the Brazilian Coordination for the Improvement of Higher Education - CAPES.

REFERENCES

[1] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.

[2] S. M. Liu and J.-H. Chen, "A multi-label classification based approach for sentiment classification," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1083–1093, 2015.

[3] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine learning*, vol. 39, no. 2-3, pp. 135–168, 2000.

[4] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2285–2294.

[5] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining (IJDWM)*, vol. 3, no. 3, pp. 1–13, 2007.

[6] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine learning*, vol. 85, no. 3, p. 333, 2011.

[7] M.-L. Zhang and Z.-H. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *Pattern recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[8] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Advances in neural information processing systems*, 2002, pp. 681–687.

[9] N. Ghamrawi and A. McCallum, "Collective multi-label classification," in *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, 2005, pp. 195–200.

[10] 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.

[11] M. G. Carneiro, J. L. Rosa, A. A. Lopes, and L. Zhao, "Network-based data classification: combining k-associated optimal graphs and high-level prediction," *Journal of the Brazilian Computer Society*, vol. 20, no. 1, p. 14, 2014.

[12] T. H. Cupertino, L. Zhao, and M. G. Carneiro, "Network-based supervised data classification by using an heuristic of ease of access," *Neurocomputing*, vol. 149, pp. 86–92, 2015.

[13] M. G. Carneiro and L. Zhao, "Organizational data classification based on the importance concept of complex networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3361–3373, 2018.

[14] T. H. Cupertino, M. G. Carneiro, Q. Zheng, J. Zhang, and L. Zhao, "A scheme for high level data classification using random walk and network measures," *Expert Systems with Applications*, vol. 92, pp. 289–303, 2018.

[15] M. G. Carneiro, R. Cheng, L. Zhao, and Y. Jin, "Particle swarm optimization for network-based data classification," *Neural Networks*, 2018.

[16] M. Carneiro and L. Zhao, "Analysis of graph construction methods in supervised data classification," in *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE, 2018, pp. 390–395.

[17] A. Khoreva, F. Galasso, M. Hein, and B. Schiele, "Classifier based graph construction for video segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 951–960.

[18] T. Jebara, J. Wang, and S.-F. Chang, "Graph construction and b-matching for semi-supervised learning," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 441–448.

[19] H. Jeong, Z. Néda, and A.-L. Barabási, "Measuring preferential attachment in evolving networks," *EPL (Europhysics Letters)*, vol. 61, no. 4, p. 567, 2003.

[20] M. G. Carneiro, "Redes complexas para classificação de dados via conformidade de padrão, caracterização de importância e otimização estrutural," Ph.D. dissertation, Universidade de São Paulo, 2017.

[21] P. Szymański and T. Kajdanowicz, "A scikit-based Python environment for performing multi-label classification," *ArXiv e-prints*, Feb. 2017.

[22] F. Briggs, B. Lakshminarayanan, L. Neal, X. Z. Fern, R. Raich, S. J. Hadley, A. S. Hadley, and M. G. Betts, "Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach," *The Journal of the Acoustical Society of America*, vol. 131, no. 6, pp. 4640–4650, 2012.

[23] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. P. Vlahavas, "Multi-label classification of music into emotions." in *ISMIR*, vol. 8, 2008, pp. 325–330.

[24] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.

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