High-Level Classification for Multi-Label Learning

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

***Abstract*—Multi-label learning (MLL) addresses the problem of learning from data items which can be associated with multiple labels simultaneously. As MLL techniques are usually derived of single-label ones, they also share common drawbacks. For example, most MLL techniques perform a low-level classification, i.e., they consider only the physical features of the input data (e.g., distance, distribution, etc) in the classification process, having troubles to detect semantic relationships among the data items, like the formation pattern for example. Recent studies have shown that learning systems based on complex networks have the ability to consider not only the physical features of the data, but also structural and topological features extracted from the network connection patterns, which is known as high-level classification. In this paper, we investigate a MLL framework which combines both low-level and high-level techniques in order to improve the predictive performance of existing MLL techniques. Experiments conducted on artificial and real-world data sets highlighted the salient features of the MLL framework and also attested its good predictive performance in comparison with widely used MLL techniques, indicating that our framework may considerable improve their predictive performance.**

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

I. Introduction

In many real-world problems, the association of an object with a unique word or label is not enough. For example, a gene can have several functional classes [1]; a newspaper can cover different topics; a song can cause different types of emotion [2]; and an image may have multiple objects in your content [3]. In this sense, the multi-label learning (MLL) differs from the multi-class one by assuming that each object can be associated with multiple labels simultaneously.

Nowadays multi-label learning has attracted a lot of attention mostly due to its fast increasing number of applications, and also its challenging characteristics, like the exponential number of label combinations. In the literature, the MLL algorithms are usually divided in two major groups: *problem transformation* and *algorithm adaptation* [4]. In the former are algorithms that handles the MLL problem by transforming it into a set of binary classification problems, e.g., Binary Relevance (BR) [5], Classifier Chain (CC) [6] and Label Powerset (LP) [7]. In the latter are the algorithms adapted from known single-label techniques in order to treat the MLL task directly, e.g., Multi-Label k-Nearest Neighbors (MLkNN) [8], Backpropagation for Multi-Label Learning (BP-MLL) [9] and Ranking Support Vector Machines (Rank-SVM) [1].

BR and CC are among the most popular MLL algorithms. Both transform a multi-label problem into a set of binary classification problems, but BR assumes independence among such binary problems while CC models them as a chain, where subsequent binary classifiers in the chain are built upon the predictions of the preceding ones. Different from BR and CC, LP transforms the MLL problem into a multi-class one, which represents all possible label combinations. In common, BR, CC and LP require a base classifier to deal with the transformed problem, being that any conventional classification technique, such as Naive Bayes (NB), Random Forest (RF) and Support Vector Machine (SVM), can be adopted. Regarding the adapted algorithms, MLkNN is one of the most popular. It consists of an adaptation of the well-known k-nearest neighbors classifier in order to handle the MLL task, and also takes into account Bayesian statistics to select the labels to be assigned. Another relevant algorithm of this algorithm adaptation group is the BP-MLL, which is an adaptation of neural networks for multi-label learning.

As most of the MLL techniques are essentially derived of the single-label ones, they share common advantages, but also common drawbacks. For example, recent studies [10]–[12] have shown that traditional classification techniques have troubles to detect the semantic relationship among the data items by considering only the physical features (e.g., distance, similarity, distribution, etc.) of the input data in the classification process. Furthermore, other study [13] pointed out that the same drawback occurs in MLL techniques. This situation is illustrated by Figure 1, which denotes a misleading case for most of the MLL techniques that are unable to correctly classify the test item (marked as a red/$\diamond$ in the figure) as belonging to both classes (green/$\pentagon$). Instead of it, such methods are much more likely to classify the test item into the class 1 (black/$\triangle$).

To overcome such a drawback, complex networks properties and measures have been incorporated in the design of efficient learning techniques, in which the salient feature is the ability to consider not only the physical features of the data but also the topological ones, which is called high-level classification (HL). Successful examples of HL techniques include the classification via pattern conformation [10], [11] and via characterization of importance [12], [14]. The former classifies a test item into the network component (class) in which its insertion

Thanks to the Brazilian National Council for Scientific and Technological Development - CNPq (grant number 439556/2018-0).

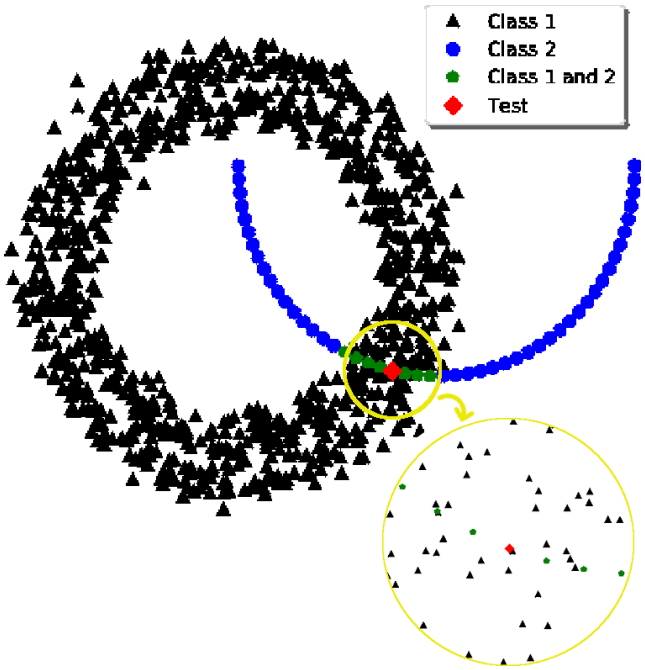


Fig. 1: An illustrative multi-label problem with clear pattern formation. Black/△ markers denote data items of the class 1; blue/○ ones denote data items of the class 2; green/⬠ ones indicate data items of the both classes; and the red/◇ marker represents a test data item, which consists of a misleading case for traditional multi-label techniques, which are unable to correctly detect the pattern formation of the input data.

causes the lowest variation of the complex network measures, while the latter derives the Google PageRank measure to classify a test item into the network component in which it receives more importance. In common, both techniques have two major steps: the construction of a network (graph) from the vector-based data, and the explotation of such a network by taking into account complex network concepts.

In this paper, we investigate the high-level classification in the MLL context. In a few words, we propose a MLL framework based on [15], which is responsible to combine the associations produced by a low-level technique (LL) with those produced by a HL one. The LL term can be any of the existing MLL techniques. The HL one is given by a classifier via pattern conformation, which is composed by a set of complex network measures. Such a framework is a refinement of that one previously published in [13]. Here we analyze a set of graph construction methods into our HL technique, consider a much more sophisticated experimental setup which simulates a real scenario for model evaluation and parameter selection, and perform statistical tests in order to support our analyses and discussions. In addition, several experiments were conducted against widely used MLL techniques on artificial and real-world data sets. In summary, it is statistically attested that our HL technique can improve the predictive performance of MLL ones in terms of accuracy and F_1 metrics.

The remainder of this work is organized as follows. Section II presents our multi-label high-level framework (MLL-HL) with a detailed description about the graph construction and the network measures analysis. Sections III and IV discuss respectively the results obtained on artificial and real-world data sets, and Section V concludes the paper.

II. Model Description

As in single-label learning, we can divide the MLL problem in two steps: training and test. In the training phase, the algorithm receives a set of training instances $\mathcal{X} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$, where each data item is formed by a tuple $(\mathbf{x}_i, \mathbf{y}_i)$, with $\mathbf{x}_i = \{x_1, \ldots, x_d\}$ representing a d-dimensional vector of features and $\mathbf{y}_i = \{y_i^{(1)}, \ldots, y_i^{(L)}\}$ the output domain of possible labels $\mathcal{L} = \{1, \ldots, L\}$, where a given class label l is said assigned to x_i if $y_i^{(l)} = 1$ or not if $y_i^{(l)} = 0$. The objective here is to learn a multi-label classifier function $f\colon \mathcal{X} \rightarrow 2^{\mathcal{L}}$. In the test phase, this multi-label function $f(\cdot)$ is adopted to predict the labels of new test items $(\mathbf{x}, ?)$, such that $f(\mathbf{x}) \subseteq \mathcal{L}$.

The MLL-HL framework is responsible by combining the associations produced by low-level and high-level classifiers. Given a test data item $\mathbf{x}$, its probability to belong to a class label l can be defined by:

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

where $\mathcal{M}_{\mathbf{x}}^{(l)}$ is the value generated by the combination of the probabilities given by a traditional MLL classifier, denoted as $\mathcal{C}_{\mathbf{x}}^{(l)}$, and by a high-level classifier, denoted as $\mathcal{H}_{\mathbf{x}}^{(l)}$. The $\lambda \in [0, 1]$ corresponds to a linear combination of both classifiers, in which high values prioritize the structural properties and low values the physical ones.

After $\mathcal{M}$ is calculated, the final output of the multi-label high-level framework is given by:

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

which means that a test item $\mathbf{x}$ receives a label l if the combination between LL and HL associations results in a probability greater than a threshold τ.

As the LL associations can be produced by any traditional MLL technique, we focus the remainder of this section on explaining the HL associations, which are generated by analyzing the pattern conformation of a given test item regarding to each network component (class). The HL technique can be divided in training and test phases. In the training one, the major step is the graph construction from which the network measures are calculated. In the test phase, the major step is the pattern conformation analysis, which verifies if a test item is in compliance with the pattern of each network class by analyzing the variation of the network measures.

A. Graph Construction

In data classification, most of the data sets are available in the feature-vector format. However, in order to exploit spatial, structural and topological relationships among the

input data, a network that represents efficiently such data must be built. In MLL context, an interesting example is [16], which proposes a method to build a graph via clique generation in order to capture label correlations. In this paper we propose four graph construction methods for the MLL task, which are derived from supervised graph construction methods [17], such as k-nearest neighbors (kNN) graph, selective k-nearest neighbors (S-kNN) graph and the ϵ-radius neighborhood (ϵN). In addition, we have also considered the degree k-nearest neighbors (D-kNN) graph, which has already been proposed for the MLL task in [13].

In the graph construction phase of our HL technique, an undirected graph $\mathcal{G}^{(l)}$ is constructed from $\mathcal{X}$ for each class label $l \in \mathcal{L}$, i.e., $g(\mathcal{X}, l) \rightarrow G^{(l)}$. Let $\mathbf{A}^{(l)}$ be the adjacency matrix of $G^{(l)}$ and suppose a given data item $\mathbf{x}_i \in l$, i.e., $y_i^{(l)} = 1$, the five graph construction methods analyzed in this work can be formally defined as follows.

1) kNN: Let kNN($\mathbf{x}_i$) be the k-nearest neighbors of $\mathbf{x}_i$, $\mathbf{A}^{(l)}$ is then defined as:

$$A_{ij}^{(l)} = \begin{cases} 1, & \text{if } \mathbf{x_i} \in \text{kNN}(\mathbf{x}_j) \text{ and } y_i^{(l)} = y_j^{(l)}, \\ 0, & \text{otherwise.} \end{cases}$$

2) kNN+ϵN: : Let D be a distance matrix where D_{ij} is the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$, $\mathbf{A}^{(l)}$ then can be obtained by:

$$A_{ij}^{(l)} = \begin{cases} 1, & \text{if } \mathbf{x_i} \in \text{kNN}(\mathbf{x}_j) \text{ and } y_i^{(l)} = y_j^{(l)}, \\ 1, & \text{if } D_{ij} \leq \epsilon \text{ and } y_i^{(l)} = y_j^{(l)}, \\ 0 & \text{otherwise.} \end{cases}$$

3) S-kNN: : Let S-kNN($\mathbf{x}_i$,l) be the k-nearest neighbors of $\mathbf{x}_i$ that belong to $l \in \mathcal{L}$, $\mathbf{A}^{(l)}$ then can be obtained as follows:

$$A_{ij}^{(l)} = \begin{cases} 1, & \text{if } \mathbf{x_i} \in \text{S-kNN}(\mathbf{x}_j, l), \\ 0, & \text{otherwise.} \end{cases}$$

4) S-kNN+ϵN: : Consider the previously defined S-kNN(.) and the matrix distance D, $\mathbf{A}^{(l)}$ is then defined by:

$$A_{ij}^{(l)} = \begin{cases} 1, & \text{if } \mathbf{x_i} \in \text{S-kNN}(\mathbf{x}_j, l), \\ 1, & \text{if } D_{ij} \leq \epsilon \text{ and } y_i^{(l)} = y_j^{(l)}, \\ 0 & \text{otherwise.} \end{cases}$$

5) D-kNN: : Let v_i be the node associated to the data item $\mathbf{x_i}$ in the graph, the adjacency matrix is given by:

$$A_{ij}^{(l)} = \begin{cases} 1, & \text{if } \mathbf{x_i} \in \text{kNN}(\mathbf{x}_j) \text{ and } y_i^{(l)} = y_j^{(l)} \text{ and} \\ & \sum_{l \in \mathcal{L}} outdegree(v_i \in V^{(l)}) < k \\ 0, & \text{otherwise.} \end{cases}$$

Figure 2 shows as example the application of the five graph construction methods on an artificial data set with two class labels. The red color circles denotes objects labeled as class 1, blue color circles as class 2 and black color ones as simultaneously both classes 1 and 2. In the figure, we show exclusively the graphs constructed for the class 1, i.e., $G^{(1)}$. One can see that there are considerable differences among the methods. For example, ϵN permits to represent dense regions, S-kNN permits to connect distant points, and D-kNN generates less connections than others. We believe that such a diversity of characteristics make one or other graph construction method more adequate, depending on the problem.

After the construction process, the network measures can then be applied in order to characterize each generated graph (class label). In this study we use three different network measures, namely clustering coefficient, assortativity and average degree, which are presented in the following.

- Clustering coefficient quantifies how much the vertices tend to group together. Basically, it measures how close each vertex of the graph is to form a clique. CC can be obtained by:

$$CC_i^{(l)} = \frac{|e_{us}^{(l)}|}{k_i^{(l)}(k_i^{(l)} - 1)} , \tag{2}$$

  in which $|e_{us}^{(l)}|$ represents the number of connections shared by adjacent neighbors of the vertex i, and k_i the degree of the vertex i. Let $\mathcal{V}^{(l)}$ be the number of vertices in the graph $G^{(l)}$, the average clustering coefficient of the graph can be obtained by:

$$CC^{(l)} = \frac{1}{\mathcal{V}^{(l)}} \sum_{i=1}^{\mathcal{V}^{(l)}} CC_i^{(l)} . \tag{3}$$

- Assortativity quantifies how much the vertices tend to connect with others with similar degree. The measure assumes 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 negative values indicate a higher probability of connected vertices having different behaviors [18]. Let $E^{(l)}$ be the number of edges in the graph $G^{(l)}$ and $i_u^{(l)}, k_u^{(l)}$ the degrees of the vertices i and k which compose an edge u, the assortativity can be calculated by:

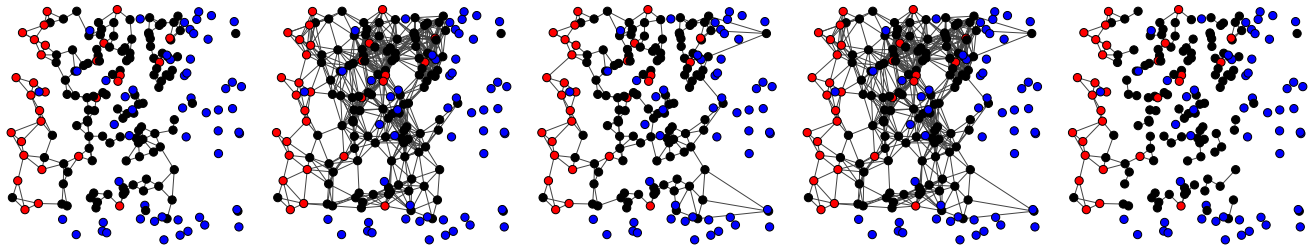
$$r^{(l)} = \frac{\frac{1}{E^{(l)}} \sum_u i_u^{(l)} k_u^{(l)} - [\frac{1}{E^{(l)}} \sum_u \frac{1}{2}(i_u^{(l)} + k_u^{(l)})]^2}{\frac{1}{E^{(l)}} \sum_u \frac{1}{2}(i_u^{2(l)} + k_u^{2(l)}) - [\frac{1}{E^{(l)}} \sum_u \frac{1}{2}(i_u^{(l)} + k_u^{(l)})]^2} \tag{4}$$

- Average Degree: This measure simply quantify the average number of connections in the graph.

$$k^{(l)} = \frac{1}{\mathcal{V}^{(l)}} \sum_{i=1}^{\mathcal{V}^{(l)}} k_i^{(l)} \tag{5}$$

B. High-Level Classification

In the test phase, the HL classifier calculates the variation of the network measures in order to classify a given test item into the class labels in which its insertion causes small (or even none) changes. Formally, let u denote a given network measure, and let $m^{(l)}$ and $m'^{(l)}$ be the result of applying u for a given graph $G^{(l)}$ respectively before and after the insertion of a test item $\mathbf{x}$, the variation of the network measure u can be defined as:



(a) kNN graph (b) kNN+ϵN graph (c) S-kNN graph (d) S-kNN+ϵN graph (e) D-kNN graph

Fig. 2: Comparative analysis of the five graph construction methods considered for our high-level technique in this study.

$$\Delta G_{\mathbf{x}}^{(l)}(u) = \frac{|m^{(l)} - m'^{(l)}|}{\sum_{q \in \mathcal{L}} |m^{(q)} - m'^{(q)}|} \ . \tag{6}$$

Basically, if the insertion of the test item causes high variation of the complex network measures in that graph, then it is probably not compliant with the class pattern represented by such a network (class). Otherwise, in the case of such a variation is low (or even does not exist), the test item probably belongs to that class. Notice that u can be any of the network measures defined before (clustering coefficient, assortativity or average degree), with m denoting the results obtained by the selected measure.

There are also a special case in the calculation of the network measures that need to be treated individually. Taking the assortativity measure, for example, it may happen that the test item does not connect to any vertex in some of the graphs. Therefore, its variation would be zero and supposedly would be perfectly in compliance with such a network class. However, this should be exactly the opposite situation. Thus, to handle such special cases we define the measure variation value as the worst possible, i.e., the maximum difference between the measure ranges, indicating that the test item caused the highest variation. Thus, the value of $CC^{(l)}$ is set to 1 and $CC'^{(l)}$ to 0 for the clustering coefficient; $r^{(l)}$ is set to 1 and $r'^{(l)}$ to 0 for the assortativity; and for the average degree is defined as $k^{(l)} = \max(k'^{(l)} - \min(k_i'^{(l)}), \max(k_i'^{(l)}) - k'^{(l)})$.

After the variations of the network measures are calculated, we proceed with the convex linear combination of such results as follows:

$$f_{\mathbf{x}}^{(l)}(u) = \Delta G_{\mathbf{x}}^{(l)}(u)\, p^{(l)}, \tag{7}$$

where $p^{(l)}$ is the proportion of items with label l, a strategy to deal with imbalanced data sets.

At the end of our HL technique, the final probability of a test item $\mathbf{x}$ be associated with a class l is given by the linear combination of the network measures variations, which is defined by:

$$\mathcal{H}_{\mathbf{x}}^{(l)} = \sum_{u=1}^{Z} \delta(u)[1 - f_{\mathbf{x}}^{(l)}(u)] \ , \tag{8}$$

with Z denoting the number of network measures adopted and $\delta \in \sum_{u=1}^{Z} \delta(u) = 1$ the weights for the variation results provided by each network measure.

III. Results on Artificial Data

In this section we present some experiments with the toy data set presented in Fig. 1. Such experiments aim at providing important insights about the salient features of our high-level technique, about the limitations of existing MLL algorithms and also about how both techniques can be combined in order to improve their results.

In the following experiments, we consider BR, CC and MLkNN low-level techniques. For sake of clarity, BR and CC are evaluated with different base algorithms, namely SVM and RF, respectively. The parameters of the algorithms were defined as follows: for MLkNN the value of k was set to 10, for SVM, C = 2^{10} and kernel = rbf; for Random Forest classifier, the number of trees was set to 100; for the high-level technique, the values of δ was set as $\frac{1}{3}$ for each of the complex network measures and the graph construction method was the S-kNN, with k = 10. The algorithms were trained with the entire data set, with the exception of the test item.

Given the test item presented in Fig. 1, Tab. I shows the combined probabilities given by the MLL framework. One can see that when $\lambda = 0$ (i.e., only the low-level classifier is considered) none of the algorithms identified the formation pattern related to class 2, giving almost zero probability to the association of the test item to such a class. This is because LL algorithms considers only the physical features of the data. As the number of "class 1" objects around the test instance is much higher than the number of "class 2" objects, this influences directly in their classification.

TABLE I: Classifiers probability obtained by our MLL-HL framework for the test item presented in the illustrative data set of Fig. 1. $\mathcal{C}_1$ corresponds to the probability given for class 1 and $\mathcal{C}_2$ for class 2.

LL Alg.	$\lambda = 0$		$\lambda = 0.30$		$\lambda = 0.65$	
	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_1$	$\mathcal{C}_2$
BR (SVM)	1.000	0.057	0.770	0.270	0.502	0.518
CC (RF)	1.000	0.030	0.770	0.251	0.502	0.508
MLkNN	0.999	0.136	0.770	0.325	0.502	0.546

Otherwise, one can see in Tab. I that with $\lambda = 0.65$ and $\tau = 0.5$, the high-level term combined with any other algorithm would classify the object correctly, i.e., simultaneously in both class 1 and 2. Proceeding with the analysis, one can see that the higher the λ, the lower the probability of class 1 and the greater of class 2. This can be explained by the HL classifier bias, which also analyzes the structural and topological properties of the data instead of considering only the physical ones. Thus, it gives a higher probability to the classes in which the test item is in high compliance with their patterns.

IV. Results on real-world data

In this section we present the results obtained by our MLL-HL framework in the four real-world data sets described in Table II. The selection was made to encompass diversity on a range of metrics, such as domain and number of instances, features and labels. Birds is a data set of audio recording from birds species [19]. Emotions aims at modeling music feelings given the timbre and the rhythm of a music [2]. Scene contains thousands of images in which their content must be classified [3]. Yeast is a biological data set related to gene expression and phylogenetic profiles [1].

TABLE II: Meta-description of the real-world data sets in terms of the number of instances (#Inst.), features (#Feat.), labels (#Lab.), training (#Train) and test (#Test) data, and the cardinality (#Car.) and density (#Den.) estimations.

Name (Domain)	#Inst.	#Feat.	#Lab.	#Train	#Test	Car.	Den.
Birds (Audio)	645	258	19	322	323	1.01	0.05
Emotions (Music)	593	72	6	391	202	1.87	0.31
Scene (Image)	2407	294	6	1211	1196	1.07	0.18
Yeast (Biology)	2417	103	14	1500	917	4.24	0.30

For sake of comparisons, we evaluate the proposed technique with three widely adopted multi-label techniques: BR, CC and MLkNN. By transforming a multi-label problem into binary ones, BR and CC requires a base classifier to deal with such problems. In this work, three base classifiers have been evaluated: Naive Bayes (NB) which is a simple and widely used technique in multi-label learning; and Random Forest (RF) and Support Vector Machine (SVM), which are state-of-the-art classification techniques for many domains.

In order to provide a fair comparison, we considered a wide range of parameters for the techniques. For BR and CC, the number of trees in RF is selected over the set $\{2^4, 2^6, 2^8, 2^{10}\}$; the kernel function in SVM can be {linear, rbf}, with the penalty parameter being selected over the set $\{2^{-5}, 2^{-3}, \ldots, 2^3\}$; and in NB, the likelihood is assumed to be Gaussian. For MLkNN, the number of neighbors is selected over the set $\{5, 10, \ldots, 30\}$.

Regarding our high-level technique, in the graph construction phase we have selected parameters k in kNN-based graphs and ϵ in ϵN graph respectively over the sets $\{2, 3, \ldots, 10\}$ and $\in \{0.1, 0.2, \ldots, 0.5\} \cdot \bar{d}$, in which $\bar{d}$ is the average distance between all pairs of samples. For the classification phase, we have λ and τ. The former is selected over the set $\{0.1,0.2,\ldots,1.0\}$, in which $\lambda = 0.7$, for example, means a contribution of 70% of the high-level term in the final prediction. The latter is selected over the set $\{0.5,0.6,\ldots,0.9\}$ and indicates the threshold which a final prediction must achieve in order to get that label class. Three complex network measures were considered in our high-level technique: assortativity, clustering coefficient and average degree. The contribution of each one of them were respectively defined as 0.4, 0.4 and 0.2.

It is worthwhile to mention that as the data sets were already divided in training and test sets, we select all parameters by running a grid search method on a 5-fold cross-validation exclusively on the training set. In the following, we evaluate the predictive performance of the techniques over three multi-label metrics: accuracy, subset accuracy and F_1-weighted. For sake of clarity, the accuracy score used here is the complement of the hamming loss metric [4].

A. Accuracy Evaluation

Table III presents the accuracy results of the multi-label techniques under comparison. In the table, "LL Alg." denotes the low-level algorithms considered (BR, CC and MLkNN); "LL Base" shows the base classifier equipped with the problem transformation technique; and "HL Graph" indicates what graph construction method has been adopted to perform the high-level classification. Notice that the symbol "-" has different meanings in the table: in "LL Base" it means that MLkNN does not require a base classifier; and in "HL Graph" it means that the classification have been performed exclusively by a low-level technique (i.e., no high-level term). Taking into account the low-level algorithms and the base classifiers, the best local results in the table are underlined and the best global results are boldfaced. For the Birds data set, the best results were achieved by BR/CC with RF as base classifier combined with the HL classification provided from the S-kNN graph. For the Emotions data set, CC(SVM) achieved the best results after being combined with the HL classification provided from both kNN and kNN+ϵN graphs. For the Scene data set, BR(SVM) provided the best results when combined with HL(kNN+ϵN) or HL(S-kNN+ϵN). In the Yeast data set, the best result was achieved out of our framework by CC(SVM).

In order to analyze statistically the effectiveness of our multi-label high-level framework, we selected the Friedman test as it permits to compare multiple techniques over multiple data sets [20]. Given the accuracy results presented in Table III, we want to know if there is any evidence that the predictive performance of the low-level classifiers is different when combined or not with the high-level ones. Thus, the null hypothesis say that they are statistically equivalent. After calculating the Friedman test under the significance level α at 0.05, the null hypothesis is rejected, i.e., at least one of the methods differs from the rest. The Nemenyi posthoc test is then applied considering again the significance level α at 0.05. The test indicates that the accuracy results obtained by the low-level techniques in combination with the high-

TABLE III: Comparative performance among widely used ML techniques and our high-level classification in terms of ML Accuracy. "LL. Alg." denotes the low-level algorithms, "LL. Base" the base classifiers, "HL. Graph" the graph construction in the high-level framework. Best local results are underlined and best global results are boldfaced.

LL	LL	HL	Datasets			
Alg.	Base	Graph	Birds	Emotions	Scene	Yeast
BR	NB	-	95.2	71.3	75.4	70.0
BR	NB	kNN	95.2	77.0	88.2	74.6
BR	NB	kNN+ϵN	95.1	75.7	88.5	74.6
BR	NB	S-kNN	95.1	76.6	87.9	74.3
BR	NB	S-kNN+ϵN	95.1	77.0	87.9	74.3
BR	NB	D-kNN	95.2	76.2	88.2	74.8
CC	NB	-	95.2	72.8	79.0	68.4
CC	NB	kNN	95.7	77.4	87.7	73.5
CC	NB	kNN+ϵN	95.1	76.4	87.8	73.5
CC	NB	S-kNN	95.1	77.1	87.6	73.5
CC	NB	S-kNN+ϵN	95.1	77.3	87.7	73.7
CC	NB	D-kNN	95.2	77.3	87.7	73.4
BR	RF	-	95.7	77.9	90.9	80.7
BR	RF	kNN	95.8	77.4	91.7	80.9
BR	RF	kNN+ϵN	96.0	78.5	91.5	80.9
BR	RF	S-kNN	96.1	78.2	91.6	80.9
BR	RF	S-kNN+ϵN	96.0	78.9	91.5	80.7
BR	RF	D-kNN	**96.2**	78.6	91.4	81.0
CC	RF	-	95.6	78.0	91.1	80.6
CC	RF	kNN	95.8	79.0	91.8	80.7
CC	RF	kNN +ϵN	95.8	78.4	91.6	81.0
CC	RF	S-kNN	96.1	78.5	91.7	80.8
CC	RF	S-kNN+ϵN	96.1	78.5	91.6	80.6
CC	RF	D-kNN	96.1	78.1	91.2	80.9
BR	SVM	-	95.7	79.5	91.9	80.8
BR	SVM	kNN	95.6	80.1	91.9	81.3
BR	SVM	kNN+ϵN	95.5	80.4	**92.0**	81.1
BR	SVM	S-kNN	95.6	80.0	91.9	81.1
BR	SVM	S-kNN+ϵN	95.6	80.0	**92.0**	81.1
BR	SVM	D-kNN	95.6	80.0	91.9	81.3
CC	SVM	-	95.7	80.4	91.3	**81.4**
CC	SVM	kNN	95.6	**80.6**	91.8	80.8
CC	SVM	kNN+ϵN	95.5	**80.6**	91.8	80.8
CC	SVM	S-kNN	95.6	80.4	91.8	81.1
CC	SVM	S-kNN+ϵN	95.6	80.4	91.8	81.1
CC	SVM	D-kNN	95.6	79.9	91.8	81.1
MLkNN	-	-	94.9	78.3	90.4	79.5
MLkNN	-	kNN	95.0	78.0	90.7	79.0
MLkNN	-	kNN+ϵN	95.1	78.1	91.0	79.1
MLkNN	-	S-kNN	94.9	77.9	90.7	78.8
MLkNN	-	S-kNN+ϵN	94.9	77.9	90.7	78.9
MLkNN	-	D-kNN	95.0	78.0	90.7	78.7

level ones provided from both kNN and kNN+ϵN graphs outperform the accuracy results obtained exclusively by the low-level techniques. The critical difference diagram found by the Nemenyi post-hoc test is shown by Fig. 3.

B. Subset Accuracy Evaluation

Now we move on to analyze the predictive performance of the techniques in terms of the subset accuracy metric. Table IV presents the results obtained for both low-level and high-level classifications. One can observe that the results here are very

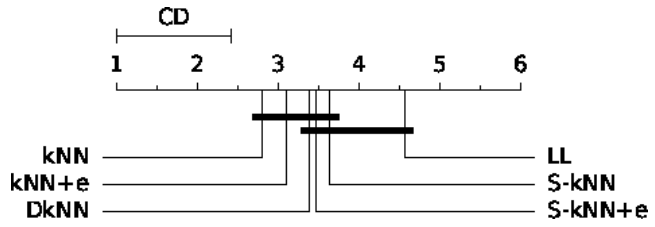


Fig. 3: Critical difference diagram obtained from the Nemenyi post-hoc test over the accuracy results presented in Table III.

smaller than those presented in Table III as this metric only considers a test data item correctly classified if the algorithm predicts correctly all of your label classes. For the Birds data set, the best result was provided by CC(RF) combined with the HL classification taken from the S-kNN graph. For the Emotions data set, again CC(SVM) obtained the best results after the combination with the HL classification provided from both kNN and kNN+ϵN graphs. For the Scene data set, CC(SVM) provided the best results after being combined with HL(kNN+ϵN). In the Yeast data set, the best result was provided exclusively by the low-level technique CC(SVM).

The Friedman test has been adopted here to provide the statistical analysis of the results showed by Table IV. Again, the null hypothesis states that the subset accuracy results obtained exclusively by the low-level techniques are equivalent to those obtained by such techniques when combined with our high-level techniques. Under the significance level α at 0.05, the test failed to reject the null hypothesis, which means that for the significance level considered both techniques (combined or not) are equivalent.

C. F_1-weighted Evaluation

Table V presents the predictive results obtained by the techniques under comparison in terms of F_1-weighted measure. The results here do not follow a pattern in relatation to the results of accuracy or subset accuracy. For example, the F_1 results in the Birds data set are worse than those of subset accuracy, although the F_1 results in Emotions and Yeast data sets are better than those of subsect accuracy. This may be explained by looking at the density of such data sets: Birds has very low density, which means there are very few labels by instance. Otherwise, Emotions and Scene have higher density, which improves the F_1 but makes much more difficult to achieve a high subset accuracy.

Regarding the overall performance in Table V, the best result for the Birds data set was achieved by CC(RF) combined with the HL classification provided from the D-kNN graph. For the Emotions data set, BR(SVM) achieved the best performance when combined with the HL(kNN+ϵN). For the Scene data set, CC(RF) provided the best results when combined with HL(kNN). In the Yeast data set, the best result was achieved by combining the high-level classification provided by HL(kNN) with the low-level classification provided by BR(RF).

In order to analyze statistically the results showed in Table V, we adopted again the Friedman test. The null hypothesis

TABLE IV: Comparative performance among widely used ML techniques and our high-level classification in terms of ML Subset Accuracy. "LL. Alg." denotes the low-level algorithms, "LL. Base" the base classifiers, "HL. Graph" the graph construction in the high-level framework. Best local results are underlined and best global results are boldfaced.

LL	LL	HL	Datasets			
Alg.	Base	Graph	Birds	Emotions	Scene	Yeast
BR	NB	-	47.4	18.8	17.1	10.5
BR	NB	kNN	47.7	21.8	50.7	11.2
BR	NB	kNN+ϵN	46.7	18.8	51.8	11.7
BR	NB	S-kNN	47.1	22.8	48.8	11.9
BR	NB	S-kNN+ϵN	47.1	23.3	48.8	11.9
BR	NB	D-kNN	47.7	21.3	50.7	11.2
CC	NB	-	47.4	19.8	28.3	9.2
CC	NB	kNN	47.7	21.8	44.0	11.2
CC	NB	kNN+ϵN	46.7	18.8	43.1	11.0
CC	NB	S-kNN	47.1	22.8	44.6	10.5
CC	NB	S-kNN+ϵN	47.1	23.3	45.1	10.5
CC	NB	D-kNN	47.7	22.8	44.0	11.3
BR	RF	-	50.2	24.8	53.1	16.6
BR	RF	kNN	48.9	24.3	60.5	18.9
BR	RF	kNN+ϵN	52.9	27.7	61.0	18.9
BR	RF	S-kNN	51.7	25.2	60.2	20.1
BR	RF	S-kNN+ϵN	51.4	29.7	60.2	18.9
BR	RF	D-kNN	51.7	28.2	59.3	19.5
CC	RF	-	49.2	27.7	55.1	21.6
CC	RF	kNN	48.3	31.2	62.4	20.2
CC	RF	kNN +ϵN	49.5	28.7	61.5	21.7
CC	RF	S-kNN	**53.3**	28.7	61.7	20.3
CC	RF	S-kNN+ϵN	52.0	26.7	60.5	19.7
CC	RF	D-kNN	51.7	29.7	60.6	20.9
BR	SVM	-	50.8	28.2	62.6	20.0
BR	SVM	kNN	49.5	33.7	63.5	21.9
BR	SVM	kNN+ϵN	51.1	33.2	63.0	20.8
BR	SVM	S-kNN	50.2	33.7	63.4	21.0
BR	SVM	S-kNN+ϵN	50.2	33.7	63.4	21.0
BR	SVM	D-kNN	49.8	33.2	63.5	21.7
CC	SVM	-	50.8	30.7	62.9	**23.0**
CC	SVM	kNN	49.5	**34.7**	63.7	21.4
CC	SVM	kNN+ϵN	51.1	**34.7**	**64.0**	21.4
CC	SVM	S-kNN	50.2	34.2	63.7	22.0
CC	SVM	S-kNN+ϵN	50.2	34.2	63.9	22.0
CC	SVM	D-kNN	49.8	32.2	63.7	21.9
MLkNN	-	-	47.4	24.3	62.0	19.2
MLkNN	-	kNN	46.7	25.7	61.5	16.7
MLkNN	-	kNN+ϵN	47.7	26.2	60.2	16.6
MLkNN	-	S-kNN	44.9	25.2	61.9	17.0
MLkNN	-	S-kNN+ϵN	44.9	25.2	61.9	16.4
MLkNN	-	D-kNN	46.4	26.2	61.5	16.9

states that the F_1 results obtained exclusively by the low-level techniques are equivalent to those obtained by combining such techniques with our high-level techniques. Under the significance level α at 0.05, the null hypothesis is rejected. The Nemenyi posthoc test is then applied. The critical difference diagram is shown by Fig. 4 and indicates that the F_1 results obtained by the high-level techniques outperform the F_1 results obtained exclusively by the low-level techniques, regardless of the graph construction method adopted by the high-level technique.

TABLE V: Comparative performance among widely used ML techniques and our high-level classification in terms of ML F_1-weighted. "LL. Alg." denotes the low-level algorithms, "LL. Base" the base classifiers, "HL. Graph" the graph construction in the high-level framework. Best local results are underlined and best global results are boldfaced.

LL	LL	HL	Datasets			
Alg.	Base	Graph	Birds	Emotions	Scene	Yeast
BR	NB	-	13.0	62.1	56.2	57.8
BR	NB	kNN	27.9	63.6	66.8	56.5
BR	NB	kNN+ϵN	24.6	60.8	66.7	56.5
BR	NB	S-kNN	21.7	63.4	66.6	55.4
BR	NB	S-kNN+ϵN	21.7	64.2	66.6	55.4
BR	NB	D-kNN	27.9	61.9	66.8	56.9
CC	NB	-	13.0	62.5	51.7	56.8
CC	NB	kNN	21.7	64.1	61.3	56.1
CC	NB	kNN+ϵN	24.6	60.0	60.1	54.4
CC	NB	S-kNN	21.7	63.8	61.2	56.1
CC	NB	S-kNN+ϵN	21.7	64.1	61.3	56.1
CC	NB	D-kNN	27.9	63.2	60.8	54.7
BR	RF	-	25.5	59.0	67.4	55.4
BR	RF	kNN	49.7	60.6	75.3	73.7
BR	RF	kNN+ϵN	55.2	66.4	74.9	73.4
BR	RF	S-kNN	53.4	65.1	74.8	73.5
BR	RF	S-kNN+ϵN	52.9	66.9	74.0	73.1
BR	RF	D-kNN	55.9	65.7	74.1	**73.8**
CC	RF	-	24.4	59.6	68.3	56.7
CC	RF	kNN	49.7	65.6	**76.3**	73.0
CC	RF	kNN +ϵN	52.5	65.8	75.3	73.6
CC	RF	S-kNN	53.5	64.2	75.6	73.1
CC	RF	S-kNN+ϵN	53.6	64.1	73.8	72.8
CC	RF	D-kNN	**56.1**	65.9	74.4	73.2
BR	SVM	-	35.9	65.4	74.7	60.2
BR	SVM	kNN	45.2	68.3	75.9	73.2
BR	SVM	kNN+ϵN	47.7	**69.5**	75.9	72.8
BR	SVM	S-kNN	47.3	67.6	75.9	72.8
BR	SVM	S-kNN+ϵN	47.3	67.3	76.0	72.8
BR	SVM	D-kNN	46.3	68.1	75.9	73.2
CC	SVM	-	35.9	65.6	74.0	59.2
CC	SVM	kNN	45.0	69.3	76.0	70.5
CC	SVM	kNN+ϵN	47.7	69.3	76.0	70.5
CC	SVM	S-kNN	47.2	68.7	75.9	71.3
CC	SVM	S-kNN+ϵN	47.2	68.7	75.9	71.3
CC	SVM	D-kNN	46.0	68.2	76.0	71.3
MLkNN	-	-	12.6	60.4	72.7	58.2
MLkNN	-	kNN	20.5	64.2	72.1	66.5
MLkNN	-	kNN+ϵN	27.1	64.4	72.2	66.3
MLkNN	-	S-kNN	31.6	63.3	72.5	66.0
MLkNN	-	S-kNN+ϵN	31.6	63.4	72.5	65.9
MLkNN	-	D-kNN	27.6	64.1	72.1	66.0

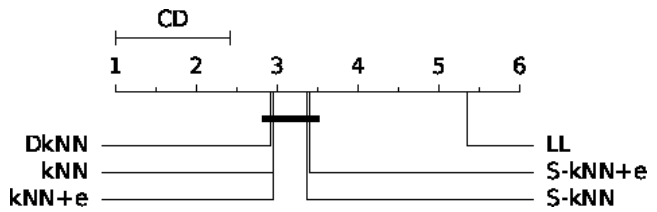


Fig. 4: Critical difference diagram related to the F_1-weighted results presented in Table V.

D. Discussion

In this section, we evaluated the contribution of our high-level technique in relation to traditional multi-label ones. The experiments were performed on real-world data sets and their results were analyzed in terms of three multi-label metrics: accuracy, subset accuracy and F_1-weighted. For every metric, a statistical test was calculated over the results to support our discussion. Such tests revealed, with a confidence level at 95%, that the classification provided by the combination between high-level and traditional techniques outperform that provided exclusively by traditional techniques when considering accuracy and F1 metrics. This is an exciting result, especially if we consider that the traditional techniques had their parameters rigorously tuned. Thus, even an small improvement is very difficult to achieve. However, differently from the existing multi-label techniques, ours is able to consider not only the physical attributes, but also the topological structure of the data, which may explain such results.

Another point is that despite the high-level technique has a considerable number of parameters (k, ϵ, λ and τ), the selection of such parameters, which took into account only the training data, is uncomplicated. In addition, the predictive performance of such a technique could also be improved by selecting other complex network measures and also by tuning their weights, which have not been considered in this paper.

V. Conclusion

The framework presented in this work is a new kind of multi-label technique able to consider both physical and topological properties of the input data through of the combination between low-level and high-level associations. In the low-level classification, which is focused on the physical features of the data, the prediction probabilities of any traditional MLL technique can be adopted. In the high-level classification, which analyzes structural and topological features of the data, the prediction probabilities are generated by a set of complex networks measures, which is responsible to verify the compliance of a given test item with the patterns associated to each network (class label).

The HL classifier is composed by two major steps: graph construction and network measures analysis. In the graph construction step, our investigation contributed with the design of four MLL graph construction methods. In the network measures analysis step, our main contribution was related to the experimental setup, which evidenced a favorable scenario for model evaluation and parameter selection. In addition, experiments conducted on artificial and real-world data sets highlighted the salient features of our MLL-HL framework. Moreover, statistical tests attested its good predictive performance in comparison with traditional MLL techniques like BR, CC and MLkNN, indicating that our framework may improve their predictive performance.

In a future work, we expect to consider more data sets and also evaluate other complex network measures. We also intend to investigate MLL techniques based on other concepts derived from complex networks, such as characterization of importance [12] or ease of access [21].

References

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

[2] 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.

[3] 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.

[4] 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.

[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] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *Data mining and knowledge discovery handbook*. Springer, 2009, pp. 667–685.

[8] 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.

[9] ——, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1338–1351, 2006.

[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] 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.

[12] 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.

[13] V. H. Resende and M. G. Carneiro, "Towards a high-level multi-label classification from complex networks," in *IEEE International Conference on Tools with Artificial Intelligence*, 2019, pp. 1140–1147.

[14] M. G. Carneiro, R. Cheng, L. Zhao, and Y. Jin, "Particle swarm optimization for network-based data classification," *Neural Networks*, vol. 110, pp. 243–255, 2019.

[15] 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.

[16] M. Tan, Q. Shi, A. van den Hengel, C. Shen, J. Gao, F. Hu, and Z. Zhang, "Learning graph structure for multi-label image classification via clique generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4100–4109.

[17] 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.

[18] M. G. Carneiro, L. Zhao, R. Cheng, and Y. Jin, "Network structural optimization based on swarm intelligence for highlevel classification," in *IEEE International Joint Conference on Neural Networks*. IEEE, 2016, pp. 3737–3744.

[19] 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.

[20] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1–30, 2006.

[21] 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.