International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 3 (2011) pp. 218-227
© MIR Labs, www.mirlabs.net/ijcisim/index.html

A Comparative Analysis of Classification Methods to Multi-label Tasks in Different Application Domains

Araken M Santos[1], Anne M P Canuto[2] and Antonino Feitosa Neto[2]

[1] Campus Angicos
Federal Rural University of Semi-Árido (UFERSA) Angicos, RN - BRAZIL, 59515-000
araken@ufersa.edu.br

[2] S Informatics and Applied Mathematics Department
Federal University of Rio Grande do Norte (UFRN) Natal, RN - BRAZIL, 59072-970
anne@dimap.ufrn.br and *antonino_feitosa@yahoo.com.br*

Abstract: **In traditional classification problems (single-label), patterns are usually associated with a single label from a set of two or more classes. When an example can simultaneously belong to more than one class (label), this classification problem is known as multi-label classification problem. Multi-label classification methods have been increasingly used in modern applications, such as music categorization, functional genomics and semantic annotation of images. In addition, the multi-label classification methods can be broadly classified in two groups, which are: problem transformation and algorithm adaptation methods. This paper presents a comparative analysis of some existing multi-label classification methods (from both groups of methods) applied to different problem domains. The main aim of this analysis is to evaluate the performance of such methods in different tasks and using different evaluation metrics.**

Keywords: Multi-label Tasks, Classification methods**.**

I. Introduction

In the machine learning context, a large amount of research has been done in traditional single-label classification method. In these methods, training examples are associated with a single label l from a previously known finite set of disjoint labels L. Hence, a single label dataset D is composed of n examples (x1,l1),(x2,l2),…(x3,l3), where x represents the input data and l represents the single label to which x belongs to [1]. However, there are real classification problems where an example can belong to more than class simultaneously. These problems are known as multi-label classification problems. [1, 2, 3]. In multilabel classification, the examples are associated with a set of labels Y □ L.

Initially, multi-label classification was motivated by application in the context of text categorization and medical diagnosis. Text documents, for instance, usually belong to more than one conceptual class. These applications could be considered as natural multi-label problems. However, nowadays, multi-label classification has attracted significant attention from a lot of researchers, motivated from an increasing number of new applications, such as semantic annotation of images [4, 5, 6] and video [7, 8], functional genomics [9, 10, 11, 12, 13], music categorization into emotions [14, 15, 16, 17], directed marketing [18], among others.

In the literature, different methods have been proposed to be applied to multi-label classification problems, which can be broadly classified as problem transformation and algorithm adaptation methods. Although there are a reasonable number of multi-label classification methods proposed in the literature, there is little effort in comparing the different multi-label methods (using both groups of methods) in different applications. In [2], for instance, the authors presented a comparative analysis of some existing methods and they used different evaluation metrics applied to the protein domain. Nonetheless, with increasing number of possible multi-label applications in different domains, it is important to perform a broader comparison, using different application domains.

As a contribution to this important topic, this paper presents a comparative analysis of some existing multi-label classification methods, using datasets of different domains. In order to do this analysis, nine multi-label classification methods are used, in which seven of them belong to the problem transformation approach (Binary Relevance (BR), Label Powerset (LP), Random k-labelsets (RAkEL), Classifier Chains (CC), Pruned Sets (PS), Ensemble of Classifier Chains (ECC) and Ensemble of Pruned Sets (EPS)) and the remaining two belong to the algorithm adaptation approach (Multi-Label k Nearest Neighbours (ML-kNN) and Back-Propagation Multi-Label Learning (BPMLL)). In addition, these methods will be evaluated using different evaluation metrics. As a result of this analysis, we aim to investigate these classification methods under different circumstances.

II. Multi-label Classification

As already mentioned, in traditional single-label classification, a classifier is built and trained using a set of examples associated with just one single label *l* of a set of disjoint labels *L*, where $|L|>1$. Moreover, in multilabel classification, the examples can be associated with a set of labels $Y \subseteq L$. In the literature, different methods have been proposed to be applied to multi-label classification problems, such as [1, 5]. The next two subsections will describe these two groups in more details.

A. Problem Transformation Methods

In this approach, the main idea is to transform the original multilabel problem into a set of single-label classification problems. It is an algorithm independent approach, since its functioning does not depend directly on the classification method used. Problem transformation approaches have employed classical algorithms such as in [19,20,21]. In this paper, we will use algorithms that belong to this group

There are several problem transformation methods in the literature that can be used to transform a multilabel dataset into a single-label dataset. Thus, any traditional classification algorithm can be used to deal with the multi-label problem. In this paper, we have chosen to compare some the most widely applied in the literature, which are:

- Binary Relevance (BR) is a popular and the most widely-used problem transformation method [1]. BR considers the prediction of each label as an independent binary classification task. Thus, BR builds *M* binary classifiers, one for each different label *L* (where M = L). For the classification of a new instance, BR outputs the union of the labels l_i that are positively predicted by the *M* classifiers. The main drawback of this method is the fact that it assumes that the labels assigned to an example are independent, ignoring the possible correlations among the possible labels.
- Label Powerset (LP) is a simple and less common problem transformation method [1]. LP considers each unique set of labels that exists in a multilabel training set as one of the labels of a new single-label classification task. Given a new instance, the single-label classifier of LP outputs the most likely label, which is actually a set of labels. The main advantage of LP is that it takes the label correlations into account. However, it suffers from the increasing complexity emerged from the large number of label subsets and the majority of these classes are associated with very few examples [23].
- Random k-labelsets (RA*k*EL) constructs an ensemble of LP classifiers [22]. Each LP classifiers is trained using a different small random subset of the set of labels. An average decision is calculated for each label l_i in *L*, and the final decision is positive for a given label if the average decision is larger than a given threshold *t*. The RA*k*EL aims to take into account label correlations and at the same time avoiding the aforementioned problems of LP [23].
- Classifier Chains (CC) [24] involves $|L|$ binary classifiers as in a binary relevance method. Classifiers are linked along a chain where each classifier deals with the binary relevance problem associated with label $l_j \in L$. The feature space of each link in the chain is extended with the *0/1* label associations of all previous links.
- Pruned Sets (PS) [25] for multi-label classification is centred on the concept of treating sets of labels as single labels. This allows the classification process to inherently take into account correlations between labels. By pruning these sets, PS focuses only on the most important correlations, which reduces complexity and improves accuracy.
- Ensembles of Classifier Chains (ECC) [23] trains *m* CC classifiers $C_1, C_2, \ldots, C_m$. Each C_k is trained with a random chain ordering (of *L*) and a random subset of *D*. Hence each C_k model is likely to be unique and able to give different multilabel predictions. These predictions are summed by label so that each label receives a number of votes. A threshold is used to select the most popular labels which form the final predicted multi-label set.
- Ensembles of Pruned Sets (EPS) [25] combine pruned sets in an ensemble scheme. PS is particularly suited to an ensemble due to its fast build times and, additionally, the ensemble counters any over-fitting effects of the pruning process and allows the creation of new label sets at classification time.

a. Algorithm Adaptations Methods

In this approach, extensions of single-label classifiers have been developed, adapting their internal mechanisms to allow their use in multilabel problems. Also, new algorithms can be developed specifically for multilabel problems [3]. It is expected that an algorithm which was developed specifically to solve multi-label problems may have a better performance than methods based on the problem transformation. [2]. However, as it is an algorithm dependent approach, it is not widely used since it has to be used with that specific classification methods to which it was proposed.

In the literature, various algorithm adaptation methods are proposed, based in different algorithms, such as: decision trees [9], probabilistic methods [23], neural networks [26], support vector machines [19], lazy and associative methods [5], boosting [27], among others. In [9], the most popular decision tree algorithm, named C4.5, was adopted for the handling of multilabel data. In [23], a probabilistic generative model is proposed, which a multilabel document is produced by a mixture of the word distributions of its labels. Extensions of the k-Nearest Neighbors (kNN) lazy learning can be found in various works, such as [5, 16, 21]. In [27], two extensions of AdaBoost algorithm are proposed, AdaBoost.MH and AdaBoost.MR. In this paper, we have chosen to compare two the most widely applied in the literature, which are:

- ML-kNN (Multi-Label k Nearest Neighbours) [5] extends the popular k Nearest Neighbors (kNN) lazy learning algorithm using a Bayesian approach. It uses the maximum a posteriori principle in order to determine the label set of the test instance, based on prior and posterior probabilities for the frequency of each label within the k nearest neighbors [5].
- Back-Propagation Multi-Label Learning (BPMLL) [28] is a neural network algorithm for multi-label learning. Its derived from the popular Back-propagation algorithm. The main modification to the algorithm is the introduction of a new error function that takes multiple labels into account. The new function was defined to capture the

characteristics of multi-label learning, that is, the labels belonging to an instance should be ranked higher than those not belonging to that instance.

III. Evaluation Metrics

The evaluation of multi-label classifiers requires different measures than those used in the case of single-label problems. Unlike the single-label problems, which the classification of an example is correct or incorrect, in a multilabel problem a classification of an example may be partially correct or partially incorrect. This can happen when a classifier correctly assigns an example to at least one of the labels it belongs to, but does not assign to all labels it belongs to. Also, a classifier could also assign to an example to one or more labels it does not belong to [2].

Several measures have been proposed in the literature for the evaluation of multilabel classifiers. According to [1], these measures can be broadly categorized in two groups: bipartition-based and ranking-based. Some of the bipartition-based measures, called example-based-measures, evaluate bipartitions over all examples of the evaluation dataset. Other bipartition-based measures, named label-based measures, decompose the evaluation process into separate evaluations for each label. Furthermore, the ranking-based measures evaluate rankings with respect to the ground truth of multi-label dataset. The next three subsections will described these three types will be described.

However, for the definitions of these measures, let an evaluation dataset of multi-label examples be denoted as (x_i, y_i), $i=1, \ldots, N$, where $Y_i \subseteq L$ is the set of true labels and $L=\{\lambda_j: j=1 \ldots M\}$ is the set of all labels. Given an example x_i, the set of labels that are predicted by an multi-label method is denoted as Z_i, while the rank predicted for a label λ is denoted as $r_i(\lambda)$. The most relevant label receives the highest rank (1), while the least relevant one receives the lowest rank (M) [1].

b. *Example-based Measures*

Hamming Loss: Hamming Loss takes into account prediction errors (incorrect label) and missing errors (label not predicted). Then, hamming loss evaluates the frequency that an example-label pair is misclassified, i.e., an example is associated to the wrong label or a label belonging to the instance is not predicted. The best performance is reached when hamming loss is equal to 0. The smaller the value of hamming loss is, the better the performance is.

$$\text{HammingLoss} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\mathrm{Y_i} \Delta \mathrm{Z_i}|}{M} \qquad \textbf{(1)}$$

Accuracy: Accuracy symmetrically measures how close Y_i *is to* Z_i.

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\mathrm{Y_i} \cap \mathrm{Z_i}|}{|\mathrm{Y_i} \cup \mathrm{Z_i}|} \qquad \textbf{(2)}$$

Precision: Precision can be defined as the percentage of true positive examples from all the examples classified as positive by the classification model.

$$\text{Precision} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\mathrm{Y_i} \cap \mathrm{Z_i}|}{|\mathrm{Z_i}|} \qquad \textbf{(3)}$$

Recall: Recall is the percentage of examples classified as positive by a classification model that are true positive.

$$\text{Recall} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\mathrm{Y_i} \cap \mathrm{Z_i}|}{|\mathrm{Y_i}|} \qquad \textbf{(4)}$$

F-Measure: F-Measure is a combination of Precision and Recall. It is the harmonic average of the two metrics and it is used as an aggregated performance score.

$$\text{F - Measure} = \frac{1}{N}\sum_{i=1}^{N}\frac{2|\mathrm{Y_i} \cap \mathrm{Z_i}|}{|Z_i| - |Y_i|} \qquad \textbf{(5)}$$

Subset Accuracy: Subset Accuracy is a very restrictive accuracy metric, considering a classification as correct if all the labels predicted by a classifier are correct.

$$\text{SubsetAccuracy} = \frac{1}{N}\sum_{i=1}^{N} I(|Z_i| = |Y_i|) \qquad \textbf{(6)}$$

c. *Label-based Measures*

The calculation of these measures for all labels can be done using two averaging operations, known as macro-averaging and micro-averaging. Consider a binary evaluation measure $F(t_p, t_n, f_p, f_n)$ that is calculated based on the number of true positives (t_p), true negatives (t_n), false positives (f_p) and false negatives (f_n). Micro-averaged precision (Mic-P) represents the ratio of examples correctly classified as l (t_p) and incorrectly (f_p) classified as l. Micro-averaged recall (Mic-R) represents the ratio of examples correctly classified as l, and all examples actually pertaining to the class l (f_n). Micro-averaged F-measure (Micro-F1) represents a harmonic mean of Micro-Precision and Micro-Recall. |L| represents the number of labels.

$$\text{Micro - F1}\frac{2 \text{ x (Mic - P) x (Mic - R)}}{\text{(Mic - P + (Mic - R)}} \qquad \textbf{(7)}$$

Where:

$$\text{Mic - P} = \frac{\sum_{l=1}^{|L|} tp_l}{\sum_{l=1}^{|L|}(tp_l + fp_l)} \qquad \text{Mic - R} = \frac{\sum_{l=1}^{|L|} tp_l}{\sum_{l=1}^{|L|}(tp_l + fn_l)}$$

Macro-average precision (Mac-P) is computed firstly by computing the precision for each label separately, and averaging over all labels. The same procedure is used for computing the macro-averaged recall (Mac-R). Macro-averaged F-measure (Macro-F1) represents a harmonic mean of Macro-Precision and Macro-Recall.

$$\text{Micro - F1}\frac{2 \text{ x (Mac - P) x (Mac - R)}}{|L|} \qquad \textbf{(8)}$$

Where:

$$\text{Mac - P} = \frac{\sum_{l=1}^{|L|} \frac{tp_l}{tp_l + fp_l}}{|L|} \qquad \text{Mac - R} = \frac{\sum_{l=1}^{|L|} \frac{tp_l}{tp_l + fn_l}}{|L|}$$

d. Ranking-based Measures

One-error: One-error evaluates the frequency of the top-ranked label that was not in the set of true labels. The best performance is reached when one-error is equal to 0. The smaller the value of one-error is, the better performance is.

$$\text{One - error} = \frac{1}{N}\sum_{i=1}^{N} \delta\left(\arg\min_{\lambda \in L} r_i(\lambda)\right) \qquad \textbf{(9)}$$

Where:

$$\delta(\lambda) = \begin{cases} 1 \text{ if } \lambda \in \mathrm{Y_i} \\ 0 \text{ otherwise} \end{cases}$$

Coverage: Coverage is defined as the distance to cover all possible labels assigned to a sample *x*. It is loosely related to precision at the level of perfect recall. The smaller the value of coverage is, the better the performance is.

$$\text{Coverage} = \frac{1}{N}\sum_{i=1}^{N} \max_{\lambda \in Y_i} r_i(\lambda) - 1 \qquad \textbf{(10)}$$

Average Precision: Average precision is the average precision taken for all the possible labels and it can evaluate algorithms as a whole. It measures the average fraction of labels ranked above a particular label $l \in Y_i$ which is actually in Y_i. The best performance is reached when average precision is equal to 1. The bigger the value of average precision is, the better performance is.

IV. Experimental Work

A. Datasets

Three different application domains are used in this investigation: Biological, Image and Music. For each application domain, one multi-label dataset was chosen, which are described as follows:

- Yeast: The biological dataset yeast [9] is concerned with protein function classification. This dataset contains micro-array expressions and phylogenetic profiles for 2417 yeast genes. Each gene is annotated with a subset of 14 functional categories from the top level of the functional catalogue(FunCat);
- Scene: The image dataset scene [4] is concerned with semantic indexing of still scenes. This dataset contains 2407 images associated with up to 6 concepts, such as *beach*, *mountain* and *field*;
- Emotions: The music emotions dataset [16] is concerned with the classification of songs according to the emotions they evoke.

Table I illustrates some basic statistics of these datasets, such as the number of examples, the number of numeric (NUM) and discrete (DIS) attributes and the number of labels, along with multilabel data statistics, such as the number of distinct label subsets (DLS), the label cardinality (LC) and the label density (LD) [1]. Label cardinality is the average number of labels per example, while label density is the same number divided by $|L|$. All datasets are available at http://mlkd.csd.auth.gr/multilabel.html.

TABLE I. STANDARD AND MULTILABEL STATISTICS

Datasets	Example	Attributes		Label	DLS	LC	LD
		NUM	DIS				
yeast	2417	103	0	14	198	4.327	0.302
scene	2712	294	0	6	15	1.074	0.179
emotions	593	72	0	6	27	1.868	0.311

B. Methods and Methodology

As already mentioned, nine different multi-label classification methods will be used in this investigation, which seven are problem transformation methods (Binary Relevance (BR), Label Powerset (LP), Random k-labelsets (RA*k*EL), Classifier Chains (CC), Pruned Sets (PS), Ensemble of Classifier Chains (ECC) and Ensemble of Pruned Sets (EPS)) and the remaining two are algorithms adaptation methods (Multi-Label k Nearest Neighbours (ML-kNN) and Back-Propagation Multi-Label Learning (BPMLL)). For each multi-label method of problem transformation approach, we apply five supervised learning algorithms, which are: k nearest neighbor (KNN), decision tree (DT), support vector machines (SVM), naïve bayes (NB), and multilayer perceptron (MLP). These specific classifiers were chosen for being very distinct in their classification procedure, performing, in this way, a broader and wider analysis of the databases.

The experimental results were evaluated using 11 evaluation measures, where 6 were example-based measures (Hamming Loss, Precision, Accuracy, Recall, F-Measure and Subset Accuracy), 2 were label-based measures (Micro F1 and Macro F1) and 3 were ranking-based measures (One-Error, Coverage and Average Precision).

All multilabel classification methods and supervised learning algorithms used in this work are implementations of the Weka-based [30] package of Java classes for multi-label classification, called Mulan [31]. This package includes implementations of some the multi-label classification methods most widely applied in the literature, such as BR, LP and RAKEL. The experiments were conducted using the 10-fold cross-validation methodology. Thus, all results presented in paper refer to the mean over 10 different test sets. Initial experiments were conducted to define the parameters values used in the supervised learning algorithms. In this paper, the results presented represent the best results obtained. In order to compare the obtained from the different learning methods, a statistical test will be applied, which is called hypothesis test (t-test) [21]. In this paper, the confidence level is 95% (α = 0.05).

V. Results and Discussion

This section presents the results obtained from this empirical study. The next three subsections will present the results for the problem transformation, algorithm adaptation methods as well as a comparison of the best method of each approach.

A. Problem Transformation Methods

Table II shows the performance of the problem transformation methods for multi-label classification when applied to yeast dataset. In this table, it is presented the results obtained by each multilabel classification method using all five supervised learning algorithm as base classifier. The results are presented using 11 different measures evaluation. The best results achieved by the learning methods in each measure are in bold. The statistical test compared the best results with the results of the other learning methods, in a two-by-two basis. The results which are statistically significant are underlined.

TABLE II. RESULTS OF PROBLEM TRANSFORMATION METHODS USING YEAST DATASET

Measure			**BR**		
	KNN	**DT**	**SVM**	**NB**	**MLP**
HammingLoss ↓	**0.193**	0.250	0.199	0.301	0.239
Accuracy ↑	**0.522**	0.433	0.499	0.420	0.458
Precision ↑	0.710	0.599	**0.714**	0.529	0.617
Recall ↑	0.602	0.573	0.575	**0.612**	0.590
F-Measure ↑	**0.652**	0.585	0.637	0.568	0.603
SubsetAccuracy ↑	**0.201**	0.060	0.146	0.093	0.107
Micro-F1 ↑	**0.652**	0.581	0.633	0.548	0.597
Macro-F1 ↑	0.402	0.386	0.324	**0.451**	0.438
One-Error ↓	**0.227**	0.393	0.256	0.346	0.290
Coverage ↓	**6.359**	9.337	9.096	7.500	7.293
AveragPrecision ↑	0.188	0.362	**0.460**	0.256	0.213
			LP		
HammingLoss ↓	0.213	0.279	**0.206**	0.242	0.256
Accuracy ↑	0.523	0.412	**0.530**	0.468	0.465
Precision ↑	0.657	0.542	**0.667**	0.593	0.586
Recall ↑	**0.627**	0.541	0.621	0.590	0.594
F-Measure ↑	0.641	0.541	**0.643**	0.591	0.590
SubsetAccuracy ↑	0.245	0.135	**0.260**	0.183	0.194
Micro-F1 ↑	0.638	0.540	**0.643**	0.596	0.584
Macro-F1 ↑	0.437	0.386	0.418	0.440	**0.443**
One-Error ↓	0.269	0.343	**0.267**	0.321	0.334
Coverage ↓	8.149	9.201	**8.065**	8.335	8.501
AveragPrecision ↑	0.430	**0.542**	0.426	0.486	0.491
			RAKEL		
HammingLoss ↓	0.208	0.252	**0.207**	0.279	0.229
Accuracy ↑	**0.493**	0.429	0.487	0.414	0.468
Precision ↑	0.683	0.592	0.690	**0.542**	0.634
Recall ↑	0.575	**0.561**	0.571	0.576	0.582
F-Measure ↑	0.624	0.576	0.625	**0.559**	0.607
SubsetAccuracy ↑	**0.163**	0.083	0.128	0.087	0.123
Micro-F1 ↑	**0.625**	0.573	0.624	0.555	0.605
Macro-F1 ↑	0.380	0.356	0.333	**0.405**	0.394
One-Error ↓	0.259	0.337	**0.255**	0.383	0.302
Coverage ↓	**9.155**	9.616	9.273	9.320	9.184
AveragPrecision ↑	0.438	0.408	0.442	**0.459**	0.398
			CC		
HammingLoss ↓	0.213	0.213	**0.211**	0.272	0.239
Accuracy ↑	**0.521**	0.428	0.489	0.445	0.477
Precision ↑	0.655	0.340	**0.679**	0.547	0.611
Recall ↑	**0.613**	0.549	0.570	0.605	0.585
F-Measure ↑	**0.633**	0.420	0.619	0.575	0.597
SubsetAccuracy ↑	**0.254**	0.153	0.196	0.123	0.193
Micro-F1 ↑	**0.634**	0.550	0.620	0.571	0.597
Macro-F1 ↑	0.434	0.346	0.403	**0.453**	0.000
One-Error ↓	0.272	0.356	**0.256**	0.331	0.386
Coverage ↓	**7.249**	8.842	8.674	7.680	7.992
AveragPrecision ↑	**0.717**	0.629	0.662	0.673	0.672
			PS		
HammingLoss ↓	0.241	0.241	0.205	0.218	**0.094**
Accuracy ↑	0.480	0.411	0.533	0.515	**0.723**
Precision ↑	0.602	0.541	0.670	0.640	**0.756**
Recall ↑	0.596	0.532	0.626	0.624	**0.704**
F-Measure ↑	0.599	0.536	0.647	0.632	**0.729**
SubsetAccuracy ↑	0.212	0.135	0.258	0.234	**0.691**
Micro-F1 ↑	0.599	0.536	0.645	0.633	**0.729**
Macro-F1 ↑	0.445	0.377	0.396	0.436	**0.731**
One-Error ↓	0.321	0.503	0.986	0.335	**0.288**
Coverage ↓	8.313	9.476	11.835	8.446	**1.081**
AveragPrecision ↑	0.666	0.574	0.278	0.661	**0.788**
			ECC		
HammingLoss ↓	0.619	0.644	0.623	0.640	**0.462**
Accuracy ↑	0.296	**0.299**	0.298	0.295	0.270
Precision ↑	0.310	0.308	0.311	0.306	**0.339**
Recall ↑	0.865	**0.909**	0.871	0.891	0.570
F-Measure ↑	0.456	**0.460**	0.458	0.456	0.425
SubsetAccuracy ↑	**0.243**	0.001	0.001	0.001	0.001
Micro-F1 ↑	0.459	**0.462**	0.459	0.458	0.429
Macro-F1 ↑	0.459	0.418	**0.469**	0.417	0.342
One-Error ↓	0.679	0.721	0.732	0.685	**0.629**
Coverage ↓	10.731	10.736	10.848	**10.705**	11,14
AveragPrecision ↑	0.435	0.427	0.425	0.426	**0.463**
			EPS		
HammingLoss ↓	0.242	0.273	**0.207**	0.218	0.248
Accuracy ↑	0.481	0.419	**0.537**	0.516	0.474
Precision ↑	0.600	0.548	**0.664**	0.641	0.591
Recall ↑	0.598	0.544	**0.643**	0.626	0.606
F-Measure ↑	0.599	0.546	**0.654**	0.633	0.598
SubsetAccuracy ↑	0.212	0.146	**0.253**	0.235	0.163
Micro-F1 ↑	0.599	0.545	**0.650**	0.634	0.593
Macro-F1 ↑	0.447	0.382	**0.515**	**0.515**	0.436
One-Error ↓	0.320	0.345	**0.265**	0.278	0.266
Coverage ↓	8.303	9.010	**7.841**	8.221	8.899
AveragPrecision ↑	0.666	0.629	**0.707**	0.689	0.665

In analyzing the performance of the supervised classification methods over the multi-label methods, it is possible to observe that there is no predominance of a supervised method throughout all multi-label methods, since k-NN delivered the best results in most of the cases for BR and CC, while MLP provided the best results for PS and SVM provided the best results for LP and EPS. In contrast, there is no best learning method for ECC and RAKEL.

The statistical test showed a division of the multi-label methods. For BR, LP and RAKEL, the performances of the learning methods are very similar, since the best results are statistically significant in a small number of cases (3 for BR, 1 for LP and 2 for RAKEL). On the other hand, there is a statistical improvement of the best method over the other methods for CC, ECC, PS and EPS, since it had statistical significant improvements of the best method over at least one other method for all used metrics.

Table III shows the performance of the problem transformation methods for multi-label classification using the scene dataset. In analysing Table III, once again, it can be observed that the predominance is divided into k-NN (BR), SVM (LP, PS and EPS) and MLP (RAKEL, CC and ECC). Unlike the previous dataset, there is always a predominance (the best result for the majority of evaluation metrics) of a supervised learning method, which can show that there is always the best choice to be made for all multi-label methods. The statistical test confirmed the predominance of the methods with the best results since these results are statistically significant over at least one other learning method for all evaluation metrics (all multi-label methods).

TABLE III. RESULTS OF PROBLEM TRANSFORMATION METHODS USING SCENE DATASET

Measure	BR				
	KNN	DT	SVM	NB	MLP
HammingLoss ↓	**0.094**	0.134	0.106	0.241	0.100
Accuracy ↑	**0.643**	0.539	0.594	0.452	0.625
Precision ↑	**0.668**	0.556	0.612	0.460	0.645
Recall ↑	0.645	0.639	0.643	**0.859**	0.671
F-Measure ↑	0.656	0.595	0.627	0.599	**0.657**
SubsetAccuracy ↑	**0.617**	0.429	0.529	0.167	0.562
Micro-F1 ↑	**0.705**	0.626	0.682	0.558	0.701
Macro-F1 ↑	**0.707**	0.637	0.688	0.576	0.697
One-Error ↓	**0.262**	0.404	0.346	0.535	0.265
Coverage ↓	**0.527**	1.232	0.990	1.003	0.577
AveragePrecision ↑	0.098	0.298	**0.377**	0.201	0.095
			LP		
HammingLoss ↓	0.097	0.147	**0.090**	0.137	0.106
Accuracy ↑	0.713	0.580	**0.735**	0.614	0.695
Precision ↑	0.744	0.601	**0.761**	0.630	0.721
Recall ↑	0.714	0.602	**0.749**	0.680	0.713
F-Measure ↑	0.729	0.602	**0.755**	0.654	0.717
SubsetAccuracy ↑	0.682	0.538	**0.696**	0.531	0.651
Micro-F1 ↑	0.720	0.591	**0.745**	0.638	0.701
Macro-F1 ↑	0.728	0.601	**0.754**	0.646	0.705
One-Error ↓	0.256	0.409	**0.246**	0.403	0.283
Coverage ↓	0.904	1.159	**0.733**	1.049	0.855
AveragePrecision ↑	0.814	0.727	**0.833**	0.742	0.806
			RAKEL		
HammingLoss ↓	**0.095**	0.107	0.097	0.179	0.090
Accuracy ↑	**0.694**	0.639	0.671	0.531	0.705
Precision ↑	**0.724**	0.660	0.692	0.539	0.731
Recall ↑	0.698	0.710	0.720	0.828	**0.742**
F-Measure ↑	0.710	0.684	0.706	0.653	**0.736**
SubsetAccuracy ↑	**0.662**	0.550	0.602	0.257	0.642
Micro-F1 ↑	0.720	0.701	0.724	0.621	**0.743**
Macro-F1 ↑	0.726	0.713	0.734	0.648	**0.749**
One-Error ↓	0.257	0.270	0.237	0.374	**0.229**
Coverage ↓	0.883	0.593	0.637	0.777	**0.553**
AveragePrecision ↑	0.816	0.835	0.847	0.777	**0.857**
			CC		
HammingLoss ↓	0.100	0.100	0.103	0.122	**0.090**
Accuracy ↑	0.701	0.587	0.696	0.582	**0.736**
Precision ↑	0.285	0.608	0.724	0.646	**0.756**
Recall ↑	0.703	0.615	0.705	0.720	**0.736**
F-Measure ↑	0.406	0.611	0.714	0.681	**0.746**
SubsetAccuracy ↑	0.669	0.538	0.659	0.432	**0.698**
Micro-F1 ↑	0.711	0.600	0.705	0.676	**0.746**
Macro-F1 ↑	0.719	0.613	0.714	0.687	**0.752**
One-Error ↓	0.268	0.391	0.278	0.296	**0.232**
Coverage ↓	0.619	1.350	0.894	0.757	**0.526**
AveragePrecision ↑	0.100	0.100	0.103	**0.812**	0.090
			PS		
HammingLoss ↓	0.092	0.092	**0.084**	0.105	0.113
Accuracy ↑	0.729	0.592	**0.751**	0.692	0.673
Precision ↑	0.759	0.611	**0.778**	0.717	0.699
Recall ↑	0.731	0.609	**0.759**	0.699	0.671
F-Measure ↑	0.745	0.610	**0.769**	0.708	0.685
SubsetAccuracy ↑	0.698	0.555	**0.717**	0.660	0.635
Micro-F1 ↑	0.736	0.600	**0.760**	0.700	0.681
Macro-F1 ↑	0.742	0.612	**0.766**	0.704	0.685
One-Error ↓	0.319	0.399	0.904	**0.287**	0.311
Coverage ↓	1.245	1.143	4.255	**0.845**	0.993
AveragePrecision ↑	0.092	0.092	0.084	0.105	**0.113**
			ECC		
HammingLoss ↓	0.470	0.494	0.470	0.543	**0.462**
Accuracy ↑	0.148	0.168	0.159	0.119	**0.270**
Precision ↑	0.152	0.176	0.162	0.163	**0.339**
Recall ↑	0.392	0.471	0.423	0.486	**0.570**
F-Measure ↑	0.219	0.256	0.235	0.244	**0.425**
SubsetAccuracy ↑	0.006	0.010	**0.007**	0.002	0.002
Micro-F1 ↑	0.233	0.257	0.247	0.244	**0.339**
Macro-F1 ↑	0.219	0.253	0.243	0.258	**0.308**
One-Error ↓	0.775	0.801	0.775	0.796	**0.629**
Coverage ↓	2.720	2.725	2.662	2.795	**0.463**
AveragePrecision ↑	0.470	0.494	0.470	0.543	**0.471**
			EPS		
HammingLoss ↓	0.092	0.143	**0.085**	0.105	0.115
Accuracy ↑	0.729	0.592	**0.751**	0.692	0.677
Precision ↑	0.759	0.611	**0.778**	0.717	0.672
Recall ↑	0.731	0.609	**0.760**	0.699	0.720
F-Measure ↑	0.745	0.610	**0.769**	0.708	0.695
SubsetAccuracy ↑	0.698	0.555	**0.715**	0.660	0.609
Micro-F1 ↑	0.736	0.600	**0.759**	0.700	0.687
Macro-F1 ↑	0.742	0.612	**0.765**	0.704	0.695
One-Error ↓	0.242	0.398	**0.225**	0.287	0.299
Coverage ↓	0.858	1.142	**0.689**	0.843	0.877
AveragePrecision ↑	0.824	0.733	**0.846**	0.809	0.798

Table IV shows the performance of the problem transformation methods for multi-label classification using the emotions dataset. For this dataset, once again, it can be observed that the predominance is divided into k-NN (BR and CC), SVM (LP, RAKEL, PS and EPS) and MLP (ECC). The results of the statistical test showed that, unlike the previous dataset, the predominance of the methods with the best results occurred only for CC, PS, ECC and EPS. For BR, LP and RAKEL, although there is always a learning method with the best result, there is no statistical evidence to state that this result is significant different from the results provided by the other learning methods for the majority of the cases.

TABLE IV. RESULTS OF PROBLEM TRANSFORMATION METHODS USING EMOTIONS DATASET

Measure	BR				
	KNN	DT	SVM	NB	MLP
HammingLoss ↓	**0.188**	0.261	0.194	0.225	0.218
Accuracy ↑	**0.551**	0.432	0.532	0.549	0.516
Precision ↑	**0.687**	0.543	0.671	0.618	0.626
Recall ↑	0.641	0.566	0.617	**0.755**	0.645
F-Measure ↑	0.663	0.553	0.642	**0.679**	0.635
SubsetAccuracy ↑	**0.307**	0.164	0.280	0.248	0.241
Micro-F1 ↑	**0.678**	0.577	0.661	0.675	0.647
Macro-F1 ↑	0.653	0.566	0.626	**0.655**	0.626
One-Error ↓	**0.256**	0.403	0.292	0.307	0.294

Coverage ↓	**1.775**	2.622	2.401	1.851	1.860
AveragePrecision ↑	**0.806**	0.687	0.750	0.789	0.785
			LP		
HammingLoss ↓	0.215	0.263	**0.198**	0.229	0.236
Accuracy ↑	0.560	0.463	**0.584**	0.519	0.516
Precision ↑	0.649	0.574	**0.677**	0.640	0.632
Recall ↑	0.671	0.575	**0.698**	0.635	0.624
F-Measure ↑	0.659	0.574	**0.687**	0.637	0.628
SubsetAccuracy ↑	0.336	0.216	**0.351**	0.263	0.268
Micro-F1 ↑	0.661	0.578	**0.688**	0.630	0.622
Macro-F1 ↑	0.647	0.561	**0.675**	0.616	0.606
One-Error ↓	0.365	0.439	**0.310**	0.393	0.365
Coverage ↓	2.319	2.608	**2.235**	2.345	2.423
AveragePrecision ↑	0.395	**0.495**	0.369	0.438	0.444
			RAKEL		
HammingLoss ↓	0.198	0.234	**0.186**	0.251	0.209
Accuracy ↑	0.577	0.502	**0.592**	0.518	0.556
Precision ↑	0.679	0.612	**0.710**	0.596	0.670
Recall ↑	0.699	0.631	0.703	**0.711**	0.677
F-Measure ↑	0.688	0.621	**0.706**	0.647	0.673
SubsetAccuracy ↑	0.332	0.238	**0.341**	0.218	0.295
Micro-F1 ↑	0.686	0.630	**0.701**	0.635	0.666
Macro-F1 ↑	0.664	0.619	**0.681**	0.617	0.650
One-Error ↓	0.292	0.326	**0.260**	0.326	0.289
Coverage ↓	2.145	**1.986**	1.989	2.102	1.964
AveragePrecision ↑	0.776	0.766	**0.798**	0.761	0.787
			CC		
HammingLoss ↓	**0.197**	**0.197**	0.207	0.223	0.211
Accuracy ↑	**0.584**	0.470	0.554	0.556	0.553
Precision ↑	**0.691**	0.574	0.649	0.575	0.663
Recall ↑	0.695	0.578	0.661	**0.753**	0.665
F-Measure ↑	**0.693**	0.576	0.655	0.652	0.664
SubsetAccuracy ↑	**0.349**	0.248	0.310	0.260	0.308
Micro-F1 ↑	**0.689**	0.588	0.663	0.677	0.663
Macro-F1 ↑	0.652	0.576	0.633	**0.659**	0.648
One-Error ↓	**0.283**	0.435	0.347	0.315	0.307
Coverage ↓	**1.756**	2.535	2.318	1.813	1.800
AveragePrecision ↑	0.801	**0.683**	0.741	0.786	0.788
			PS		
HammingLoss ↓	0.203	0.203	**0.192**	0.226	0.211
Accuracy ↑	0.579	0.455	**0.599**	0.541	0.572
Precision ↑	0.670	0.564	**0.675**	0.636	0.661
Recall ↑	0.698	0.561	**0.728**	0.679	0.698
F-Measure ↑	0.684	0.562	**0.701**	0.657	0.679
SubsetAccuracy ↑	0.351	0.218	**0.367**	0.286	0.337
Micro-F1 ↑	0.681	0.571	**0.704**	0.650	0.673
Macro-F1 ↑	0.666	0.554	**0.692**	0.631	0.656
One-Error ↓	0.723	0.469	0.902	**0.427**	0.791
Coverage ↓	3.801	2.590	4.364	**2.331**	3.944
AveragePrecision ↑	0.469	0.671	0.344	**0.713**	0.433
			ECC		
HammingLoss ↓	0.630	0.651	0.640	0.645	**0.494**
Accuracy ↑	0.275	**0.282**	0.268	0.282	0.279
Precision ↑	0.293	0.294	0.284	0.306	**0.320**
Recall ↑	0.727	**0.792**	0.729	0.816	0.552
F-Measure ↑	0.418	0.428	0.409	**0.445**	0.405
SubsetAccuracy ↑	0.003	0.022	0.002	0.002	**0.007**
Micro-F1 ↑	0.426	0.438	0.422	**0.448**	0.428
Macro-F1 ↑	0.419	0.432	0.416	**0.442**	0.430
One-Error ↓	0.863	0.802	0.836	0.879	**0.792**
Coverage ↓	3.892	3.817	3.975	4.003	**3.293**
AveragePrecision ↑	0.420	0.438	0.420	0.412	**0.480**
			EPS		
HammingLoss ↓	0.203	0.276	**0.193**	0.224	0.213
Accuracy ↑	0.579	0.440	**0.599**	0.541	0.579
Precision ↑	0.670	0.555	0.673	0.641	0.637
Recall ↑	0.698	0.553	**0.735**	0.676	0.733
F-Measure ↑	0.684	0.554	**0.703**	0.658	0.682
SubsetAccuracy ↑	0.351	0.201	**0.366**	0.283	0.329
Micro-F1 ↑	0.681	0.554	**0.705**	0.652	0.681
Macro-F1 ↑	0.666	0.536	**0.691**	0.632	0.666
One-Error ↓	0.336	0.442	**0.300**	0.398	0.322
Coverage ↓	2.211	2.599	**2.138**	2.258	2.160
AveragePrecision ↑	0.757	0.677	**0.775**	0.729	0.765

After analyzing all three datasets individually, we will do a general analysis, taking into consideration all three datasets together. First of all, there is a consensus about the best supervised learning for two multi-label methods, BR (with k-NN being the best result for all three datasets) and LP (SVM). However, these best results were not statistically significant for the marjority of the cases (only for the scene dataset). On the other hand, for PS, EPS, CC and ECC, there is the same best result for two datasets (SVM for PS and EPS, MLP for ECC and k-NN for CC). RAKEL was the only multi-label method in which there is no predominance of a supervised learning method.

The overall best supervised learning (summing the best results of all three datasets) was SVM since it provided the highest number of best results (98 out of 231), followed by MLP (51) and *k*-NN (47). The other learning methods had performance much poorer than these two methods. Figure 1 illustrates the number of best results for each dataset. When analyzing the best results for the different datasets, SVM provided the highest number of the best results for all datasets.

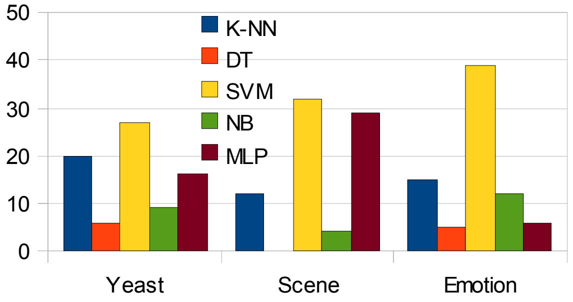

Figure 1. The performance (best results) of the learning algorithms, separated by multi-label classification method.

B. *Algorithm Adaptation Methods*

Table V shows the performance of the algorithms adaptation methods for multi-label classification when applied to yeast dataset. From Table V, it can be observed that ML-kNN has provided the best results for 10 evaluation metrics (coverage was the only exception). Of these 10 best results, 8 of them were statistically significant. Based on this, we can say that ML-kNN had a better performance than BPMLL for most of the evaluation metrics, when applying the yeast dataset.

TABLE V. RESULTS OF ALGORITHMS ADAPTARION METHODS USING YEAST DATASET

	ML-kNN	BPMLL
HammingLoss ↓	**0.193**	0.322
Accuracy ↑	**0.520**	0.185
Precision ↑	0.718	0.189
Recall ↑	**0.600**	0.236
F-Measure ↑	**0.654**	0.210
SubsetAccuracy ↑	**0.189**	0.185
Micro-F1 ↑	**0.651**	0.202
Macro-F1 ↑	**0.476**	0.459
One-Error ↓	**0.234**	0.805
Coverage ↓	6.301	**2.523**
AveragePrecision ↑	**0.762**	0.428

Table VI shows the performance of the algorithms adaptation methods for multi-label classification when applied to scene dataset. As in the previous dataset, ML-kNN provided the best results for all evaluation metrics and these best results were statistically significant in 10 cases (out of 11).

TABLE VI. RESULTS OF ALGORITHMS ADAPTATION METHODS USING SCENE DATASET

	ML-kNN	BPMLL
HammingLoss ↓	**0.085**	0.579
Accuracy ↑	**0.691**	0.212
Precision ↑	**0.811**	0.629
Recall ↑	**0.714**	0.700
F-Measure ↑	**0.759**	0.663
SubsetAccuracy ↑	**0.643**	0.212
Micro-F1 ↑	**0.747**	0.233
Macro-F1 ↑	**0.750**	0.219
One-Error ↓	**0.226**	0.466
Coverage ↓	**0.456**	7.447
AveragePrecision ↑	**0.867**	0.629

Table VII shows the performance of the algorithms adaptation methods for multi-label classification when applied to emotion dataset. The same behavior of the previous datasets, ML-kNN with the best results and most of these best results were statistically significant.

TABLE VII. RESULTS OF ALGORITHMS ADAPTARION METHODS USING EMOTIONS DATASET

	ML-kNN	BPMLL
HammingLoss ↓	**0.262**	0.433
Accuracy ↑	**0.366**	0.276
Precision ↑	**0.624**	0.347
Recall ↑	**0.408**	0.443
F-Measure ↑	**0.493**	0.389
SubsetAccuracy ↑	0.143	**0.276**
Micro-F1 ↑	**0.489**	0.381
Macro-F1 ↑	**0.469**	0.426
One-Error ↓	**0.386**	0.668
Coverage ↓	**2.327**	3.159
AveragePrecision ↑	**0.708**	0.542

C. Problem Transformation Versus Algorithm Adaptation

In this subsection, the best problem transformation method will be compared with the best algorithm adaptation method. In order to do this analysis, the best result for each evaluation metric was pick for each approach. For instance, the best result for hamming loss in the problem transformation methods was obtained by BR using kNN. In this case, this value was used to compare with the best result of this measure obtained by the algorithm adaptation. Table VII presents the best results for problem transformation methods and algorithm adaptation methods, for all three datasets.

TABLE VIII. RESULTS OF THE BEST RESULTS FOR PROBLEM TRANSFORMATION AND ALGORITHM ADAPTATION METHODS, FOR ALL THREE DATASETS

	Yeast dataset	
	Adaptation	Transformation
HammingLoss ↓	0,193	**0,094**
Accuracy ↑	0,520	**0,723**
Precision ↑	0,718	**0,756**
Recall ↑	0,600	**0,704**
F-Measure ↑	0,654	**0,729**
SubsetAccuracy ↑	0,189	**0,691**
Micro-F1 ↑	0,651	**0,729**
Macro-F1 ↑	0,476	**0,731**
One-Error ↓	**0,234**	0,256
Coverage ↓	2,523	**1,081**
AveragePrecision ↑	**0,762**	0,717
	Scene dataset	
HammingLoss ↓	0,085	**0,084**
Accuracy ↑	0,691	**0,751**
Precision ↑	**0,811**	0,778
Recall ↑	0,714	**0,760**
F-Measure ↑	0,759	**0,769**
SubsetAccuracy ↑	0,643	**0,717**
Micro-F1 ↑	0,747	**0,760**
Macro-F1 ↑	0,750	**0,766**
One-Error ↓	0,226	**0,225**
Coverage ↓	**0,456**	0,463
AveragePrecision ↑	**0,867**	0,846
	Emotion dataset	
HammingLoss ↓	0,262	**0,192**
Accuracy ↑	0,366	**0,599**
Precision ↑	0,624	**0,691**
Recall ↑	0,408	**0,816**
F-Measure ↑	0,493	**0,703**
SubsetAccuracy ↑	0,276	**0,367**
Micro-F1 ↑	0,489	**0,705**
Macro-F1 ↑	0,469	**0,692**
One-Error ↓	0,386	**0,283**
Coverage ↓	2,327	**1,756**
AveragePrecision ↑	0,708	**0,713**

In analyzing Table VIII, it can be observed that the algorithm adaptation provided the best results, when compared with the problem transformation method in 28 out of 33 analyzed cases. Of these 33 cases, 22 cases where statistically significant.

The results obtained in this table show that the use of algorithm adaptation methods has been the best option for mutlti-label methods, when compared with the problem transformation methods.

VI. Final Remarks

This paper presented a comparison between different methods of multi-label classification for different domain application, more specifically three different domains (datasets). As they are algorithm-independent approaches, five different learning algorithms were used. Finally, these methods were analyzed using 11 different evaluation metrics

In the experimental results, it can be observed that there are some useful information about the choice of the supervised learning method related to a multi-label method. For instance, the best supervised learning method for BR was always k-NN and for LP was SVM. There are also some supervised learning methods which were the best for most of the analyzed cases (two out of three datasets), SVM for PS and EPS, MLP for ECC and k-NN for CC. This may be a good indication of which supervised learning method to be used when we choose a multi-label method.

Of the problem transformation methods, the ones using SVM usually presented the overall best results for all datasets. This may have occurred because SVM manages better with features from the datasets and methods used. Of the algorithm adaptation methods, ML-kNN has provided the best results in almost all analyzed cases. In order to analyze which is the best multi-label method, the best problem transformation result was comparatively analyzed with the best algorithm adaptation result and the algorithm adaptation methods had provided the best results for almost all evaluation metrics. This results may be an indication theat the use of algorithm adaptation methods can be a better choice than the problem transformation methods, even when we use a wide range of supervised learning methods.

Acknowledgment

This work was supported in part by the CNPq, under processes number 200.755/2009-9.

References

[1] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining Multi-label Data", Data Mining and Knowledge Discovery Handbook, O. Maimon, L. Rokach (Ed.), Springer, 2nd edition, 2010.

[2] R. Cerri, R. R. Silva, and A. C. Carvalho,. "Comparing Methods for Multilabel Classification of Proteins Using Machine Learning Techniques", BSB 2009, LNCS 5676, 109-120, 2009.

[3] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification", Proc of Eur conf on Mach Learning and Knowledge Discovery in Databases, LNAI 5782(254-269), 2009.

[4] M. Boutell, J. Luo, X. Shen, and C. Brown, "Learning multi-label scene classification", Pattern Recognition 37, 1757–1771, 2004.

[5] M. L. Zhang, and Z. H. Zhou, "Ml-knn: A lazy learning approach to multi-label learning", Pattern Recognition 40, 2038–2048, 2007.

[6] S. Yang, S. K. Kim, and Y. M. Ro, "Semantic home photo categorization", IEEE T on Circuits and Systems for Video Technology, 17, 324–335, 2007.

[7] G. J. Qi, X. S. Hua, Y. Rui, J. Tang, T. Mei, and H. J. Zhang, "Correlative multi-label video annotation", Proceedings of the 15th international conference on Multimedia, New York, NY, USA, 2007.

[8] C. G. M. Snoek, M. Worring, J. C. van Gemert, J. M. Geusebroek, and A. W. M. Smeulders, "The challenge problem for automated detection of 101 semantic concepts in multimedia", Proc of the 14th ACM int conf on Multimedia, New York, NY, USA, 2006.

[9] A. Clare, and R. King, "Knowledge discovery in multi-label phenotype data", Proc of the 5th Eur Conf on Principles of Data Mining and Knowledge Discovery, Freiburg, Germany , 2001.

[10] A. Elisseeff, and J. Weston, "A kernel method for multi-labelled classification", Adv in Neural Inf Processing Systems 14, 2002.

[11] H. Blockeel, L. Schietgat, J. Struyf, S. Džeroski, and A. Clare, "Decision trees for hierarchical multilabel classification: A case study in functional genomics", LNCS 4213 (2006) 18–29, 2006.

[12] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni,, "Hierarchical classification: combining bayes with svm", Proceedings of the 23rd international conference on Machine learning, 177–184, 2006.

[13] Z. Barutcuoglu, R Schapire, and O Troyanskaya, "Hierarchical multi-label prediction of gene function", Bioinformatics 22, 830–836, 2006.

[14] T. Li, and M. Ogihara, M. "Detecting emotion in music", Proceedings of the International, 239–240, 2003.

[15] T. Li, and M. Ogihara, "Toward intelligent music information retrieval", IEEE Transactions on Multimedia 8, 564–574, 2006.

[16] A. Wieczorkowska, P. Synak, and Z. Ras, "Multi-label classification of emotions in music", Proc of the International Conference on Intelligent Information Processing and Web Mining , 307–315, 2006.

[17] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas, "Multilabel classification of music into emotions", Proc. 9th Int Conference on Music Information Retrieval, Philadelphia, PA, USA, 2008.

[18] Y. Zhang, S. Burer, and W. N. Street, "Ensemble pruning via semi-definite programming", J of Mach Learn Res 7, 1315–1338, 2007.

[19] S. Godbole, and S. Sarawagi, "Discriminative methods for multi-labeled classification", 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2004.

[20] D. Vilar, M. J. Castro and E. Sanchis, "Multi-Label Text Classification Using Multinomial Models," Proc. Fourth Int'l Conf. España for Natural Language Processing (EsTAL '04), 2004.

[21] X. Luo, and N. A. Zincir-Heywood, "Evaluation of two systems on multi-class multi-label document classification" International Syposium on Methodologies for Intelligent Systems, 161–169, 2005.

[22] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-Labelsets for Multi-Label Classification", IEEE T on Knowledge and Data Engineering, 2010.

[23] A. McCallum, "Multi-label text classification with a mixture model trained by em", Proc of AAAI' 99 Workshop on Text Learning, 1999.

[24] J. Read, B. Pfahringer, G. Holmes and E. Frank, "*Classifier chains for multi-label classification*", Proc of 13th European Conference on Principles and Practice of Knowledge Discovery in Databases and 20th European Conference on Machine Learning, Bled, Slovenia. Springer, 2009.

[25] J. Read, B. Pfahringer and G. Holmes, "*Multi-label classification using ensembles of pruned sets*", Proc 8th IEEE International Conference on Data Mining, Pisa, Italy, pages 995-1000. IEEE Computer Society, 2008.

[26] M. L. Zhang, "ML-RBF: RBF neural networks for multi-label learning", Neural Processing Letters, 29(2): 61-74, 2009.

[27] R. E. Schapire, and Y. Singer, "BoosTexter: A Boosting-based System for Text Categorization", Mac. Learn. 39(2-3) 135-168, 2000.

[28] M. L. Zhang, Z. H. Zhou, "*Multi-label neural networks with applications to functional genomics and text categorization*", IEEE Transactions on Knowledge and Data Engineering 18 (2006) 1338–1351.

[29] M. Zhang, and Z. Zhou, "Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization", IEEE Trans. on Knowl. and Data Eng. 18, 10, 1338-1351, 2006.

[30] I. H.Witten, ans E. Frank, "Data Mining: Practical Machine Learning tools and techniques", Morgan Kaufmann, 2005.

[31] G. Tsoumakas, R. Friberg, E. Spyromitros-Xiou, I, Kataks, and J. Vilcek, "Mulan software - java classes for multi-label classification Available at: http://mlkd.csd.auth.gr/multilabel.html#Software

Author Biographies

Araken M. Santos received the B.S. degree from the Federal University of Rio Grande do Norte, Brazil, in 2004 and the M.Sc. degree from the Federal University of Rio Grande do Norte, Brazil, in 2008. Currently, he is an D.Sc. student at the Federal University of Rio Grande do Norte, Brazil, and associate professor in the Federal Rural University of Semi-Árido, Brazil. His interests include pattern recognition, classifier combination and neural networks.

Anne M P Canuto received the BS degree from the Federal University of Rio Grande do Norte, Brazil, in 1992, the MSc degree from the Federal University of Pernambuco, Brazil, in 1995, and the PhD degree from the University of Kent, in 2001. Currently, she is an associate professor in the informatics and applied mathematics department, Federal

University of Rio Grande do Norte. She has published over 50 articles in scientific journals and conferences. Her interests include pattern recognition, classifier combination, and multi-agent systems.

Antonino A Feitosa Neto received the B.S. degree from the Federal University of Rio Grande do Norte, Brazil, in 2010. Currently, he is a M.Sc. student at the Federal University of Rio Grande do Norte, Brazil. His interests include pattern recognition, classifier combination and optimization techniques.