Label Construction for Multi-label Feature Selection

Newton Spolaôr, Maria Carolina Monard
Laboratory of Computational Intelligence
Institute of Mathematics and Computer Science
University of São Paulo
São Carlos, Brazil

Grigorios Tsoumakas
Machine Learning and Knowledge Discovery Group
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
{*newtonspolaor,hueidianalee*}*@gmail.com,*
mcmonard@icmc.usp.br,greg@csd.auth.gr

Huei Diana Lee
Laboratory of Bioinformatics
Western Paraná State University
Foz do Iguaçu, Brazil

***Abstract*—Multi-label learning handles datasets where each instance is associated with multiple labels, which are often correlated. As other machine learning tasks, multi-label learning also suffers from the curse of dimensionality, which can be mitigated by dimensionality reduction tasks, such as feature selection. The standard approach for multi-label feature selection transforms the multi-label dataset into single-label datasets before using traditional feature selection algorithms. However, this approach often ignores label dependence. This work proposes an alternative method, *LCFS*, which constructs new labels based on relations between the original labels to augment the label set of the original dataset. Afterwards, the augmented dataset is submitted to the standard multi-label feature selection approach. Experiments using Information Gain as a measure to evaluate features were carried out in 10 multi-label benchmark datasets. For each dataset, the quality of the features selected was assessed by the quality of the classifiers built using the features selected by the standard approach in the original dataset, as well as in the dataset constructed by four *LCFS* settings. The results show that setting *LCFS* with simple strategies using pairs of labels gives rise to better classifiers than the ones built using the standard approach in the original dataset. Moreover, these good results are accomplished when a small number of features are selected.**

***Keywords*-feature ranking; filter feature selection; Binary Relevance; Information Gain; systematic review**

I. Introduction

In multi-label learning, each instance is associated with multiple labels simultaneously. A key difference between multi-label and traditional binary or multi-class single-label learning is that the labels in multi-label learning are not mutually exclusive. Thus, in comparison with traditional single-label learning, multi-label learning is more general and more challenging to solve.

As the labels in multi-label learning are often correlated, a significant challenge is how to explore the label structure to improve classification performance. As other machine learning tasks, multi-label learning also suffers from the "*curse of dimensionality*". Dimensionality reduction (feature selection), which aims to find a small number of features that describes the dataset, as well as, or even better than the original set of features does [1], is an effective way to mitigate the curse of dimensionality.

The standard approach for multi-label Feature Selection (FS), which transforms the multi-label dataset into single-label datasets before using traditional FS algorithms, is implementable within the Binary Relevance (*BR*) approach [2]. A drawback of *BR* is that label dependence is often ignored. An alternative to reduce this problem would be to construct labels based on relations among the original labels and include the new labels during the feature selection phase. The main idea of variable (label or feature) construction is to gather information about the relations among the original variables from data and infer additional variables. Although feature construction methods are less usual than feature selection methods [3], they have already been used for single-label and multi-label learning [4]. Nevertheless, to the best of our knowledge, there is little research on label construction for multi-label data.

In this work, we propose the Label Construction for Feature Selection (*LCFS*) method to build binary variables (new labels) based on label relationships. These variables are then included as new labels in the original dataset and the standard multi-label FS approach is used in the augmented dataset to select features. Afterwards, the dataset described by the selected features and the original labels can be submitted to any multi-label learning algorithm. Experiments in 10 benchmark datasets using the Information Gain (*IG*) measure for FS, suggest that setting *LCFS* with simple strategies in pairs of labels gives rise to better classifiers than the ones built using the standard approach when a small number of features are selected.

The rest of this paper is organized as follows: Section II briefly presents multi-label learning and FS. Section III summarizes related work, which have been found through a systematic literature review. The proposed method *LCFS* is described in Section IV. Section V presents the experimental evaluation. Section VI concludes and highlights future work.

II. Background

This section presents basic concepts and terminology of multi-label learning and feature selection.

A. Multi-label learning

Let D be a dataset composed of N examples $E_i = (\mathbf{x}_i, Y_i)$, $i = 1 \ldots N$. Each example (instance) E_i is associated with a feature vector $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iM})$ described by M features (attributes) X_j, $j = 1 \ldots M$, and its multi-label Y_i, which consists of a subset of labels $Y_i \subseteq L$, where $L = \{y_1, y_2, \ldots, y_q\}$ is the set of q labels. Table I shows this representation. In this scenario, the multi-label classification task consists in generating a classifier H which, given an unseen instance $E = (\mathbf{x}, ?)$, is capable of accurately predicting its multi-label Y, *i.e.*, $H(E) \rightarrow Y$.

Table I
Multi-label data

	X_1	X_2	$\ldots$	X_M	Y
E_1	x_{11}	x_{12}	$\ldots$	x_{1M}	Y_1
E_2	x_{21}	x_{22}	$\ldots$	x_{2M}	Y_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
E_N	x_{N1}	x_{N2}	$\ldots$	x_{NM}	Y_N

1) Categorizing multi-label learning algorithms: Multi-label learning methods can be organized into two main categories [2]: (i) problem transformation methods, where the multi-label learning problem is decomposed into a set of single-label (binary or multi-class) learning tasks; and (ii) algorithm adaptation methods, which adapt specific learning algorithms to handle multi-label datasets directly. The key philosophy of the problem transformation methods is to fit data to algorithms, while for the algorithm adaptation methods is to fit algorithms to data [5].

Another categorization, proposed by Zhang and Zhou [5], organizes multi-label learning methods based on the order of label dependence taken into account, as exploring label dependence during learning can improve its performance. First-order strategies ignore co-existence of other labels. The problem transformation Binary Relevance (*BR*) approach exemplifies this category by transforming a multi-label dataset into q single-label binary datasets, learning from each single-label problem separately and combining the results. Second-order strategies consider pairwise relations between labels, such as interactions between any pair of labels, or the ranking between relevant and irrelevant labels. High-order strategies consider relations among more labels.

Although high-order strategies potentially model wider label dependences, they are usually computationally more demanding. This work focuses on finding second-order relations between single labels from the multi-label dataset and representing them as new labels. The idea is that, by labeling instances with the original and the constructed labels, it will be possible to allow feature selection methods based on *BR* to incorporate label pairwise information.

2) Evaluation Measures: Unlike single-label classification where the classification of a new instance has only two possible outcomes, correct or incorrect, multi-label classification should also take into account *partially* correct classification. A complete discussion on multi-label performance measures is out of the scope of this work, and can be found in [2]. In what follows, we briefly describe the four multi-label evaluation measures used in this work.

F-measure, *Hamming Loss* and *Accuracy*, defined by Equations 1 to 3, are example-based evaluation measures, where Δ represents the symmetric difference of two sets, Y_i and Z_i are the true and the predicted multi-label respectively.

$$F\text{-}measure(H, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{2|Y_i \cap Z_i|}{|Z_i| + |Y_i|}. \quad (1)$$

$$Hamming\,loss(H, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \Delta Z_i|}{|L|}. \quad (2)$$

$$Accuracy(H, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|}. \quad (3)$$

In addition, *Micro-averaged F-measure* (F_b), defined by Equation 4, is a label-based measure, where $T_{P_{y_i}}$, $F_{P_{y_i}}$, $T_{N_{y_i}}$ and $F_{N_{y_i}}$ represent, respectively, the number of true/false positives/negatives for a label $y_j \in L$.

$$F_b(H, D) = \frac{2 \sum_{j=1}^{q} T_{P_{y_j}}}{2 \sum_{j=1}^{q} T_{P_{y_j}} + \sum_{j=1}^{q} F_{P_{y_j}} + \sum_{j=1}^{q} F_{N_{y_j}}}. \quad (4)$$

All these performance measures range in the interval $[0, 1]$. For *Hamming Loss*, the smaller the value, the better the multi-label classifier performance is, while for the other measures, greater values indicate better performance.

B. Feature selection

Regardless of the multi-label learning approach, any FS method addresses a few relevant issues, such as the interaction with the learning algorithm and the feature importance measure. Three approaches determine different interactions between a FS method and the learning algorithm: wrapper, embedded and filter [1]. The first two approaches strongly interact with the learning algorithm. On the other hand, filters use general properties of the dataset to remove unimportant features from it, regardless of the learning algorithm. Thus, the features chosen may not be the best ones for a specific learning algorithm. The FS algorithms used in this work fall within this approach.

Many measures have been used to estimate the importance of features based on properties of the dataset. A popular single-label FS measure is Information Gain (*IG*), which evaluates each feature according to the dependence between this feature and a single label, as defined by Equation 5.

$$IG(D, X_j) = entropy(D) - \sum_{v} \frac{|D_v| \, entropy(D_v)}{|D|}. \quad (5)$$

In other words, the *IG* of feature X_j, $j = 1 \ldots M$, calculates the difference between the entropy of the dataset D and the weighted sum of the entropy of each subset $D_v \subseteq D$, where D_v consists in the set of examples where X_j has the value v. Therefore, if X_j has 10 distinct values[1] in D, the sum would be applied to 10 different D_v datasets.

III. Related work on multi-label FS

Feature selection has been an active research topic in supervised learning, with many related publications and comprehensive surveys [1]. Although most publications are related to single-label learning, a number of papers have recently reported results to support multi-label learning.

Aiming at capturing a wide, replicable and rigorous overview of the topic, we have instantiated the systematic literature review process [6] for multi-label FS in [7], and recently updated it in [8]. Table II summarizes the 72 publications found in terms of the two categorizations described in Section II: order of label dependence and interaction with the learning algorithm. The 72 references are listed in the supplementary material available at http://www.labic.icmc.usp.br/pub/mcmonard/ExperimentalResults/BRACIS2014.pdf.

Table II
Number of related publications per approach ($total = 72$)

categorization	approach	#publications (%)
order of label dependence	first-order	44 (61,11%)
	second-order	10 (13,89%)
	high-order	9 (12,50%)
	hybrid	3 (4,17%)
	unrecognized	6 (8,33%)
interaction with the learning algorithm	filter	50 (69,44%)
	embedded	10 (13,89%)
	wrapper	7 (9,72%)
	unrecognized	5 (6,94%)

As can be observed, filters and first-order strategies have been the most usual choices in multi-label FS. The proposed method, described next, considers second-order relations between labels.

IV. The proposed method: *LCFS*

Given a multi-label dataset D with the set of single labels $L = \{y_1, y_2, y_3, \ldots, y_q\}$, the main idea of *LCFS* is to construct q' new single labels by combining the original labels within pairs (y_i, y_j), $i \neq j$, $y_i \in L$ and $y_j \in L$. In each iteration, *LCFS* selects a pair of labels (y_i, y_j) from L and combines the labels within this pair to generate a new label y_{ij}. After repeating this procedure q' times, the q' new labels are included in the label set L, such that information about pairwise relationships between original labels can be used by the binary relevance approach for feature selection.

The *LCFS* method consists of two steps, each one concerned with answering a different question:

1) Selection: which pairs of labels (y_i, y_j) should be chosen?
2) Generation: how to combine these labels to generate the new labels y_{ij}?

Figure 1 illustrates these steps for $q' = 1$.

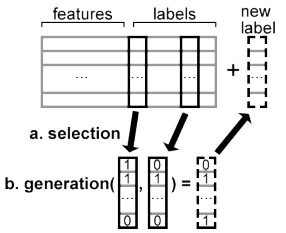

Figure 1. Applying the two steps of *LCFS* to construct $q' = 1$ new labels

Thus, setting *LCFS* involves choosing a strategy to select label pairs and a strategy to combine the labels within each pair. An additional parameter is the number of new labels q' that will be constructed. In what follows, the two *LCFS* steps are described.

A. Step 1: selection

Given the set of labels $L = \{y_1, y_2, y_3, \ldots, y_q\}$ of the dataset D, *LCFS* chooses q' different pairs of labels[2] (y_i, y_j), $i \neq j$, according to a selection strategy. The idea is that these pairs capture some pairwise relationships between the labels to be considered by feature selection.

LCFS supports different selection strategies, such as the simple Random Selection (*RS*), as well as, heuristic strategies based on the number of instances labelled by each original single label (label frequency). In particular, two strategies considering label frequency are Co-occurrence-based Selection (*CS*) and related Labels Selection (*LS*). *CS* sorts in descending order label pairs according to the co-occurrence c_c, *i.e.*, the number of instances labelled by both labels within a pair, and selects the first q' different pairs. On the other hand, *LS* counts (1) the number of instances in which the labels within a pair agree, c_e, and (2) the number of instances in which the labels within a pair disagree, c_d. Then, the pairs are sorted, in descending order, into two lists according to the values of c_e and c_d. The pair with greatest value is selected, removed from the correspondent list and the procedure is repeated until selecting q' different pairs.

B. Step 2: generation

In this step, *LCFS* combines both labels from all previously selected pairs (y_i, y_j), $i \neq j$, to construct the new

[1] Discretization is applied to numerical features before using *IG*.

[2] In this work, two label pairs are considered different if they do not have a common label.

labels y_{ij}. The idea is that the values of y_{ij} represent a pairwise relationship between y_i and y_j. In the end, all instances in D are labeled by the q original labels and the q' new labels. *LCFS* supports different combination strategies between binary variables (labels). In this work, we use three simple logical operators to generate the values of the new labels of each instance in D. The logical operators are:

AND : $y_{ij} = 1$ iff $y_i = y_j = 1$; $y_{ij} = 0$ otherwise.
XOR : $y_{ij} = 1$ iff $y_i \neq y_j$; $y_{ij} = 0$ otherwise.
XNOR: $y_{ij} = 1$ iff $y_i = y_j$; $y_{ij} = 0$ otherwise.

The AND operator clearly highlights co-ocurring labels. XNOR, also known as the coincidence function, assigns the value 1 to y_{ij} *iff* the labels y_i and y_j agree, whereas XOR does the opposite.

After generating the q' new labels, the traditional *BR* feature selection approach can be applied to the dataset now labeled by the $q+q'$ labels. Note that, by combining *BR* with *LCFS*, any single-label FS algorithm can be applied to the augmented dataset with second-order label information [5].

The *LCFS* method has been implemented in Mulan[3], a multi-label learning package based on Weka[4].

V. Experimental evaluation

In this work, we use the *lazy* multi-label learning algorithm *BRkNN-b* to evaluate the quality of the features selected, as *lazy* algorithms are sensitive to irrelevant features. *BRkNN-b*, which is implemented in Mulan, is an improved adaptation of the single-label k-Nearest Neighbor (*kNN*) algorithm to classify multi-label examples [9].

In the experiments, a filter FS approach based on Information Gain combined with Binary Relevance (*IG-BR*) is performed (1) in the dataset with the original set of labels (standard approach) and (2) in the dataset with the original labels and the ones constructed by a *LCFS* setting. Regardless of the label set used, *IG-BR* transforms the multi-label dataset into single-label datasets, applies *IG* in each single-label dataset and averages the *IG* score of each feature X_j, $j = 1 \ldots M$, across all labels. The resulting feature ranking sorts the M averaged *IG* values in descending order [10]. Recall that the labels constructed by *LCFS* are only used to select features.

Afterwards, the subsets of features $X' \subset X$, $|X'| = 10\%M, 20\%M, \ldots, 90\%M$, ranked by each FS method are used to describe the dataset, which is submitted to *BRkNN-b*.

Regarding *LCFS*, four settings combining different Selection (**S**) and Generation (**G**) strategies — Sections IV-A and IV-B — are considered:

LS-X. **S**: *LS*, **G**: XOR or XNOR is chosen based on the lists sorted by the values of c_e and c_d
CS-A. **S**: *CS*, **G**: AND
RS-A. **S**: *RS*, **G**: AND
RS-X. **S**: *RS*, **G**: XOR or XNOR is randomly chosen

Recall that the *LS* strategy sorts the label pairs based on the values of c_e and c_d. For a given label pair (y_i, y_j), *LS-X* applies the XNOR operator to generate the new label y_{ij} if the pair was selected from the list sorted by c_e; otherwise, it applies the XOR operator. *RS-X* randomly selects the XOR or XNOR operator. See the supplementary material for an illustrative example.

We set the number of new labels $q' = \lfloor \frac{q}{2} \rfloor$, *i.e.*, every single label is selected once if q is even, or one single label is left out if q is odd.

A. Multi-label datasets

Table III summarizes the characteristics of the 10 datasets used in this work. For each dataset it shows: dataset name (Dataset); dataset domain (Domain); number of instances (N); number of features (M); feature type ($Type$); number of labels ($|L|$); label cardinality (LC), which is the average number of labels associated with each example; label density (LD), which is the cardinality normalized by $|L|$; and the number of different multi-labels (#Diff).

Table III
Dataset description

Dataset	Domain	N	M	$Type$	$\|L\|$	LC	LD	#Diff
1-*Cal500*	music	502	68	numeric	174	26.044	0.150	502
2-*Corel5k*	image	5000	499	discrete	374	3.522	0.009	3175
3-*Corel16k001*	image	13766	500	discrete	153	2.859	0.019	4803
4-*Emotions*	music	593	72	numeric	6	1.869	0.311	27
5-*Fapesp*	text	332	8669	discrete	66	1.774	0.027	206
6-*Genbase**	biology	662	1185	discrete	27	1.252	0.046	32
7-*Llog-f**	text	1253	1004	discrete	75	1.375	0.018	303
8-*Magtag5k*	music	5260	68	numeric	136	4.839	0.036	4163
9-*Scene*	image	2407	294	numeric	6	1.074	0.179	15
10-*Yeast*	biology	2417	103	numeric	14	4.237	0.303	198

Except for datasets *5-Fapesp* and *8-Magtag5k*, the other datasets are available in the Mulan[5] and Meka[6] repositories. In particular, *5-Fapesp* was built by members of our research laboratory[7] [11]. *8-Magtag5k*[8] is further described in [12]. Furthermore, *6-Genbase** and *7-Llog-f** are pre-processed versions of the publicly available datasets in which an identification feature and unlabeled examples, respectively, were removed.

B. Results and discussion

First, we compared the learning performance of the classifiers built from the datasets described by the features selected by (1) the standard *IG-BR* approach and by (2) *IG-BR* after applying the four *LCFS* settings to construct the new sets of labels: *LS-X*, *CS-A*, *RS-A* and *RS-X*.

For each dataset, the number of nearest neighbors k was set as the one that maximizes the *Example-based*

[3] http://mulan.sourceforge.net
[4] http://www.cs.waikato.ac.nz/ml/weka
[5] http://mulan.sourceforge.net/datasets.html
[6] http://meka.sourceforge.net/#datasets
[7] The dataset can be obtained from the authors.
[8] http://tl.di.fc.ul.pt/t/magtag5k.zip

F-measure of the *BRkNN-b* classifiers built using the original dataset. This value was found in a preliminary study, in which k was varied in the interval $[1..27]$ with step 2 and in the interval $[29..99]$ with step 10. The k values used for each dataset in Table III were $[(1,59),(2,21),(3,49),(4,15),(5,29),(6,1),(7,13),(8,17),(9,27),(10,21)]$. All the remaining parameters related to classification and FS were executed with default values. Note that this experimental setup clearly favours the classifiers built using the original datasets.

For each evaluation measure described in Section II-A2 and estimated according to the 10-fold cross-validation strategy, the results for *IG-BR* in the original datasets, as well as *IG-BR* in the datasets augmented by using the four *LCFS* settings considering 10% up to 90% of the features selected (5 FS methods $\times 9$ number of features $= 45$ cases) were tabulated. Due to lack of space, this information is available in the SM. For the sake of completeness, we also include in these tables the performance of the *BRkNN-b* classifiers built using All Features (AF), *i.e.*, without feature selection, as well as, the results of a baseline multi-label classifier named $General_B$ [13].

Although most of the classifiers built using the selected features are better than $General_B$, no FS method was significantly better in terms of each evaluation measure used in this work and number of selected features $|X'|$. In fact, when using the Friedman's statistical test [14] under the null hypothesis, which states that the performance of the classifiers built after FS are equivalent, the hypothesis is not rejected (significance level $\alpha = 0.05$). Nevertheless, the average rankings calculated by the Friedman's test give us information about the best method across the datasets — Table IV. In this table, each symbol identifies a FS method: – (*IG-BR*), $*$ (*LS-X*), o (*CS-A*), $\times$ (*RS-A*) and $+$ (*RS-X*).

Table IV
BEST FS METHOD BASED ON THE AVERAGE RANKINGS

$\|X'\|$	10%	20%	30%	40%	50%	60%	70%	80%	90%
$F\text{-}measure$	+	+	–	+	×	o	–	+	–
$Hamming\,loss$	+	+	–	×	×	o	+	–	–
$Accuracy$	+	+	–	+	×	o	–	$*$	–
F_b	+	+	×	+	×	o	–	+	–

As can be observed, *RS-X* often achieves the best average rankings, specially when $|X'| < \frac{|X|}{2}$, *i.e.*, less than half of the features are used, whereas the standard *IG-BR* comes next. Thus, we decided to focus on the comparison of both methods. We applied the Wilcoxon signed-ranks test, recommended for comparisons of two algorithms [14], with the null hypothesis that both methods are equivalent. Table V shows the p-value for each evaluation measure and number of selected features $|X'|$, as well as the best FS method when the null hypothesis is rejected ($\alpha = 0.05$ and 0.1).

Regardless of the evaluation measure, the classifiers built using the features selected by *IG-BR* in the datasets augmented by using *RS-X* are significantly better when the number of features is small (13 cases: 4 for $\alpha = 0.05$ and 9 for $\alpha = 0.1$). However, the classifiers built using the features selected directly by *IG-BR* achieve significant improvement in only 3 cases for $\alpha = 0.1$. These results show a clear advantage of *RS-X* when the number of features selected is small, which is the aim of FS. Furthermore, comparing to *IG-BR*, the increase in complexity of *RS-X* is only related to the cost of using XOR or XNOR to generate the set of new labels, as the selection of the label pairs is random.

Table V
WILCOXON STATISTICAL TEST RESULTS: *IG-BR* VS *RS-X*

$\|X'\|$	10%	20%	30%	40%	50%	60%	70%	80%	90%
					F-measure				
p-value	0.03	0.08	0.43	0.05	0.10	0.50	0.91	0.91	0.57
$\alpha = 0.05$	+								
$\alpha = 0.1$	+	+		+	–				
					Hamming Loss				
p-value	0.06	0.02	0.08	0.46	0.47	0.84	0.95	0.58	0.16
$\alpha = 0.05$		+							
$\alpha = 0.1$	+	+	–						
					Accuracy				
p-value	0.04	0.13	0.55	0.16	0.25	0.50	0.57	1.00	0.31
$\alpha = 0.05$	+								
$\alpha = 0.1$	+								
					F_b				
p-value	0.07	0.01	0.43	0.10	0.10	0.73	0.65	1.00	0.55
$\alpha = 0.05$		+							
$\alpha = 0.1$	+	+		+	–				

By outperforming *IG-BR*, which also aggregates the *IG* scores of each feature by averaging them across all labels, in small subsets of selected features, *RS-X* could be related to previous work which compares different aggregation strategies in the original set of labels [10]. This work suggests that the averaging strategy is a good choice when the number of selected features is small.

Up to now, we have compared the performance of the classifiers built by *BRkNN-b* using the features selected by *IG-BR* in the original datasets and *IG-BR* in the datasets augmented by using the four settings of *LCFS*. In this comparison, *RS-X* shows better behavior when fewer features are selected. However, the quality of the classifiers have not been taken into account. To this end, we compare the performance of the classifiers built by *BRkNN-b*, using up to 40% of the features selected by *IG-BR* in the datasets augmented by using *RS-X*, with the performance achieved by the *BRkNN-b* classifiers using All Features (AF), *i.e.*, the original dataset. Table VI shows, for each dataset, and for each one of the four evaluation measures, *i.e.*, $F\text{-}measure$/ $Hamming\,loss$/ $Accuracy$/ F_b, whenever the classifiers built using the features selected by *IG-BR* in the datasets augmented by *RS-X* have evaluation measure values better than or equal to (indicated by ★), or at most 5% worse than the ones of the classifiers using AF (indicated by ☆). The symbol 0 indicates the other cases.

Table VI
CLASSIFIERS BUILT USING THE FEATURES SELECTED WITH THE AID OF *RS-X vs* THE CLASSIFIERS BUILT USING ALL FEATURES

	10%	20%	30%	40%
1	☆/☆/☆/☆	☆/☆/☆/☆	☆/☆/☆/☆	☆/★/☆/☆
2	★/★/★/★	★/★/★/★	★/★/★/★	★/★/★/★
3	★/★/★/★	★/★/★/★	★/★/★/★	★/★/★/★
4	0 / 0 / 0 / 0	0 / 0 / 0 / 0	☆/ 0 / 0 /☆	☆/ 0 /☆/☆
5	★/★/★/★	★/★/★/★	★/★/★/★	★/★/★/★
6	★/★/★/★	★/★/★/★	★/★/★/★	★/★/★/★
7	★/★/★/★	★/★/★/★	★/★/★/★	★/☆/★/★
8	0 / 0 / 0 / 0	0 / 0 / 0 / 0	0 / 0 / 0 / 0	0 /☆/ 0 / 0
9	0 / 0 / 0 / 0	0 / 0 / 0 / 0	0 / 0 / 0 / 0	0 / 0 / 0 / 0
10	☆/ 0 / 0 /☆	☆/☆/☆/☆	☆/☆/☆/☆	☆/☆/☆/☆

As can be observed, very good results were obtained in 5 out of the 10 datasets, where the four evaluation measure values of the classifiers based on our proposal are better than or equal to the ones of the classifiers using AF (except for dataset 7-*Llog-f**, where *Hamming loss* is at most 5% worse when 40% of the features are considered). Good results were obtained in all cases in dataset 1-*Cal500*, as the results are at most 5% worse than the AF ones, whereas there is a very good Hamming loss result when 40% of the features are considered. Similar results are obtained in dataset 10-*Yeast* when 20% up to 40% of the features are considered. Good results were obtained in dataset 4-*Emotions* only when 40% of the features are considered (except for *Hamming loss*). On the other hand, poor results are obtained in datasets 8-*Magtag5k* and 9-*Scene* even when 40% of the features are considered. In fact, it is necessary to consider 60% and 70% of the features selected in datasets 8-*Magtag5k* and 9-*Scene* respectively in order to obtain (☆/ ☆/ ☆/ ☆).

VI. CONCLUSION

This work proposes *LCFS*, a method to construct labels to support multi-label feature selection.

Four different *LCFS* settings are compared with the standard approach for FS, which only considers the original label set, in 10 benchmark datasets. The best setting, *RS-X*, which uses the XOR and XNOR operators to combine labels within pairs randomly selected, gives rise to better classifiers when a small number of selected features (up to 40%) is considered. This shows that constructing labels to support multi-label feature selection is a promising research topic and deserves further attention from the community.

As future work, we plan to evaluate more sophisticated settings, such as strategies based on label weighting [15].

ACKNOWLEDGMENT

This research was supported by the São Paulo Research Foundation (FAPESP), grant 2011/02393-4.

REFERENCES

[1] H. Liu and H. Motoda, *Computational Methods of Feature Selection*. Chapman & Hall/CRC, 2008.

[2] G. Tsoumakas, I. Katakis, and I. P. Vlahavas, "Mining multi-label data," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. Springer, 2010, pp. 667–685.

[3] K. Lillywhite, D.-J. Lee, B. Tippetts, and J. Archibald, "A feature construction method for general object recognition," *Pattern Recognition*, vol. 46, no. 12, pp. 3300–3314, 2013.

[4] R. Prati and F. Olivetti de Franca, "Extending features for multilabel classification with swarm biclustering," in *IEEE Congress on Evolutionary Computation*, 2013, pp. 2964–2971.

[5] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 99, pp. 1–59, 2013.

[6] B. A. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," EBSE-2007-01 Technical Report. 65 pg., 2007, Evidence-based Software Engineering.

[7] N. Spolaôr, M. C. Monard, and H. D. Lee, "A systematic review to identify feature selection publications in multi-labeled data," ICMC Technical Report No 374. 31 pg., 2012, University of São Paulo.

[8] N. Spolaôr, E. A. Cherman, M. C. Monard, and H. D. Lee, "ReliefF for multi-label feature selection," in *Brazilian Conference on Intelligent Systems*, 2013, pp. 6–11.

[9] E. Spyromitros, G. Tsoumakas, and I. Vlahavas, "An empirical study of lazy multilabel classification algorithms," in *Hellenic conference on Artificial Intelligence*. Springer-Verlag, 2008, pp. 401–406.

[10] N. Spolaôr and G. Tsoumakas, "Evaluating feature selection methods for multi-label text classification," in *BioASQ workshop*, 2013, pp. 1–12.

[11] R. G. Rossi and S. O. Rezende, "Building a topic hierarchy using the bag-of-related-words representation," in *Symposium on Document Engineering*, 2011, pp. 195–204.

[12] G. Marques, M. A. Domingues, T. Langlois, and F. Gouyon, "Three current issues in music autotagging," in *Conference of the International Society for Music Information Retrieval*, 2011, pp. 795–800.

[13] J. Metz, L. F. Abreu, E. A. Cherman, and M. C. Monard, "On the estimation of predictive evaluation measure baselines for multi-label learning," in *Advances in Artificial Intelligence - IBERAMIA 2012*, ser. Lecture Notes in Computer Science, J. Pavón, N. D. Duque-Méndez, and R. Fuentes-Fernández, Eds. Springer, 2012, vol. 7637, pp. 189–198.

[14] J. Demšar, "Statistical comparison of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, no. 1, pp. 1–30, 2006.

[15] S. Jungjit, A. A. Freitas, M. Michaelis, and J. Cinatl, "Two extensions to multi-label correlation-based feature selection: A case study in bioinformatics," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2013, pp. 1519–1524.