International Journal of Computational Intelligence Systems, Vol. 8, Supplement 2 (2015), 3-15

Feature Selection for Multi-label Learning: A Systematic Literature Review and Some Experimental Evaluations

Newton Spolaôr [1,2], Huei Diana Lee [2] *, Weber Shoity Resende Takaki [2], Feng Chung Wu [2]

[1] Laboratory of Computational Intelligence,
Institute of Mathematics and Computer Science, University of São Paulo,
Trabalhador São-carlense Avenue, 400, ZIP code: 13560-970, São Carlos, Brazil,
Tel.: +55 16 3373-9646. Fax: +55 16 3373-9751
E-mail: newtonspolaor@gmail.com

[2] Laboratory of Bioinformatics,
Graduate Program in Engineering and Energy Dynamic Systems, Western Paraná State University
Presidente Tancredo Neves Avenue, 6731, ZIP code 85867-900, Foz do Iguaçu, Brazil,
Tel.: +55 45 3576-8815. Fax: +55 45 3575-2733
huei@unioeste.br,{webertakaki,wufengchung}@gmail.com

Received 29 July 2015

Accepted 30 October 2015

Abstract

Feature selection can remove non-important features from the data and promote better classifiers. This task, when applied to multi-label data where each instance is associated with a set of labels, supports emerging applications. Although multi-label data usually exhibit label relations, label dependence has been little studied in feature selection. We proposed two multi-label feature selection algorithms that consider label relations. These methods were experimentally competitive with traditional approaches. Moreover, this work conducted a systematic literature review, summarizing 74 related papers.

Keywords: data mining, information gain, label construction for feature selection, multi-label ReliefF, machine learning, survey

1. Introduction

Technological progress makes it possible to generate and store more and more representative data in different domains. As analyzing such data is a complex task, intelligent processes for knowledge extraction, such as data mining, have been proposed [3]. Data mining consists of three steps: pre-processing, pattern extraction and post-processing. As a result, the process yields models (hypotheses) constructed using learning algorithms. These models represent knowledge (patterns) on a particular domain.

The pre-processing step is an important one due to the influence it holds in the learning process, among other reasons. Feature Selection (FS), a task inherent to this step, is usually applied to a dataset described by an attribute-value table.

In particular, FS can be defined as a process of searching for a subset of important features in terms of an importance measure or criterion that reflects

*Corresponding author

relevance and/or non redundancy of features. Afterwards, it removes the remaining features. This process assists in the subsequent extraction of patterns inherent to data, leading to a potential reduction of "curse of dimensionality" effects that impair learning [5]. Moreover, FS can promote time/money saving if, for example, costly features from the domain are removed because they are not necessary for learning.

Many of the importance measures proposed in the literature are applicable to single-label datasets, in which each instance (or example) is associated with a unique label (target concept). However, it is intuitive for human beings to associate their observations (instances) with two or more concepts, characterizing multi-label data. The interest in extracting knowledge from this data has increased in the last years. One can note various applications [16], such as emotion analysis, media annotation, text categorization and prediction of virus resistance to drugs. In fact, our literature review, carried out according to the Systematic literature Review method (SR), found 74 publications that use importance measures in multi-label data.

Given the increasing interest in the subject, one could raise the following question: *how to organize research on multi-label feature selection?*

An alternative to answer the question is obtained by defining dimensions to categorize publications on the subject. For example, the interaction between FS and the learning algorithm is a dimension already considered in multi-label learning surveys [18]. This dimension includes three approaches: embedded, filter and wrapper. The filter approach is the unique that is independent of the learning algorithm, *i.e.*, filter methods attempt to quantify the importance of the attributes for learning by taking into account only information extracted from data.

Besides the interaction with the learning algorithm, another dimension that could be used to organize multi-label FS methods, adapted from hierarchical classification, is the scope of the multi-label (set of labels) considered by the method. The multi-label FS scope is associated with methods that identify a feature subset for the multi-label dataset. On the other hand, the single-label FS scope includes methods that choose a different feature subset for each single label in a multi-label. Later, these methods may apply a strategy to combine the partial results. Additionally, the hybrid FS scope encompasses methods that simultaneously exhibit properties of both scopes previously mentioned.

An additional dimension to organize FS methods, adapted from multi-label learning [18], corresponds to the intensity of the label dependency exploration. As a motivation, exploring this dependency has been highlighted by the community as an issue that could support learning [2]. In this dimension, one could organize FS methods as first-order, second-order and high-order strategies. First-order strategies consider labels individually, ignoring the label dependency. The second approach considers, for example, relations between pairs of labels, while the third one takes into account relations among labels.

Figure 1 shows the three multi-label FS dimensions that are considered to answer the raised question.

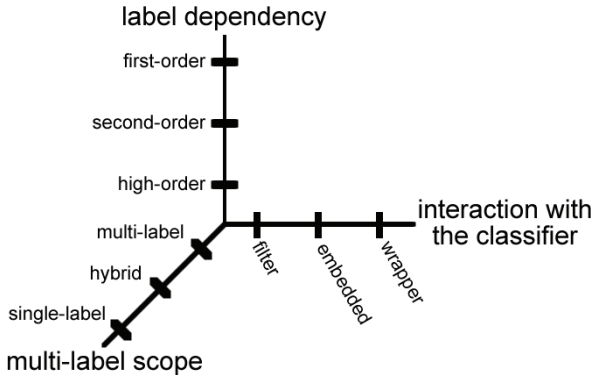

Figure 1: A taxonomy to organize publications on multi-label feature selection

It is believed that identifying label dependency may contribute to improve learning performance. Our literature review supports this assumption, as it allows one to note that considering label dependency in multi-label FS has led to good results in related work.

We hypothesize that feature selection algorithms for multi-label data that consider label dependency will perform better than those that ignore this information. To verify this hypothesis, we propose

the design and implementation of filter multi-label feature selection algorithms that consider label relations. These algorithms are evaluated, for example, according to the performance of the classifiers generated using the features selected by each algorithm.

The rest of this work is organized as follows. Section 2 describes multi-label learning concepts. Sections 3, 4 and 5 present the three main contributions of this work. Section 6 concludes this paper.

2. Multi-label learning

Besides defining multi-label learning, this section describes learning methods and multi-label evaluation measures considered in this work.

2.1. Definitions

Let D be a dataset composed of N instances $E_i = (x_i, Y_i)$, $i = 1 \ldots N$. Each instance E_i is associated with a feature vector $x_i = (x_{i1}, x_{i2}, ..., x_{iM})$ described by M features 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 1 shows this representation. In this scenario, the multi-label classification task consists in generating a classifier H that, given a new instance $E = (x, ?)$, is capable of accurately predicting its multi-label Y, *i.e.*, $H(E) \rightarrow Y$.

Table 1: 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

The main difference between multi-label and single-label learning is that the former deals with a set of labels often correlated, whereas the latter considers possible values of the class (labels) that are mutually exclusive.

In single-label classification, each instance E_i is associated with only one class value, which in turn is the label y_i contained in the set of labels L, *i.e.*, $y_i \in L$, with $|L| > 1$. If there are two or more possible class values ($|L| > 2$), the problem is named *multi-class classification*. If the class value is *yes* or *no*, the problem is named *binary classification*.

2.2. Multi-label learning methods

Multi-label learning methods can be organized into two main categories [18,16]:

- Problem Transformation: transforms the multi-label dataset into one or more single-label datasets. After processing, traditional classification algorithms are used to solve the single-label problem(s) separately. Two simple methods in this category are Binary Relevance (BR) and Label Powerset (LP);
- Algorithm Adaptation: learns from multi-label datasets directly, *i.e.*, without transforming them, after adapting specific learning algorithms. Two methods that exemplify this category consists in Binary Relevance k Nearest Neighbor (*BRkNN*) [14] and Multi-label Mutual k Nearest Neighbor (*MLMUT*) [1].

The frequently used *BR* approach decomposes the multi-label problem into many binary classification problems, one per label. In each single-label problem, instances associated with the label are regarded as positive and the other ones as negative. Any single-label learning algorithm can be used as the basis for *BR* to learn from the binary problems. Later, a labeling criterion to predict the multi-label for a new instance according to the results of binary classifiers is used. Besides ignoring label dependency, *BR* has to build a large number of binary classifiers when the number of labels q is large, which can lead to the imbalance problem in binary data. *LP*, in turn, transforms a multi-label dataset into a multiclass one by mapping each distinct multi-label Y into a single class value. Although the label dependency is partially considered, *LP* can also lead to the imbalance problem in multi-class data.

The algorithm *BRkNN* is $|L|$ times faster than the application of the *BR* approach using the lazy k Nearest Neighbor (*kNN*) as base algorithm because the former searches for the k nearest neighbors only

once. Adaptations to deal with the multi-label problem directly yielded the Binary Relevance k Nearest Neighbor extension A (*BRkNN-a*) and Binary Relevance k Nearest Neighbor extension B (*BRkNN-b*) [14]. Both extensions are based on a label confidence value, which is estimated for each label according to the percentage of the k neighbors that contains this label. In particular, *BRkNN-b* uses a more sophisticated strategy to specify this percentage that considers the average size of the multi-labels of the k neighbors.

It should be emphasized that lazy algorithms are susceptible to irrelevant features. Thus, they are useful for the evaluation of feature selection methods. Based on the mentioned properties, the *BRkNN-b* extension was used in this work to evaluate multi-label FS methods.

MLMUT [1] uses the k nearest neighbors strategy to identify the set of instances B_k that will be considered to predict the multi-label of a new instance E, *i.e.*, $|B_k| = k$. Let $B_k(E)$ be the set of k nearest neighbors of a new instance E. The set $B_{mut}(E)$ consists of the instances E_i that are contained in $B_k(E)$ when E is contained in $B_k(E_i)$. This leads to $|B_{mut}(E)| \leqslant |B_k(E)|$.

This work proposes the extensions Multi-label ReliefF (*RF-ML*) and Mutual Multi-label ReliefF (*RFM-ML*). Although both search for nearest neighbors during the feature evaluation process, *RFM-ML* considers the mutual neighborhood idea inherent to *MLMUT*.

2.3. Multi-label classification evaluation

Unlike single-label classification, which generates correct or incorrect prediction, multi-label classification should also consider partially correct prediction. The multi-label evaluation measures used in this work are (example-based) F-measure, Hamming Loss, Accuracy and Micro-averaged F-measure. These frequently used measures are described in Ref. [16] and range in the interval [0,1]. For Hamming Loss, the smaller the value, the better the multi-label classifier is, whereas higher values for the other measures indicate better classifiers.

2.3.1. Baseline classifiers

As reference, we used $General_B$ [6], a simple baseline learning algorithm that learns by looking only at the multi-labels of the dataset. As $General_B$ do not necessarily focuses on optimizing specific loss functions, it can be used as a global baseline for the difficult task of evaluating multi-label predictions.

$General_B$ ranks the q simple labels in L according to their relative frequencies in data multi-labels to include only the σ most frequent labels in the predicted multi-label Z. To obtain a representative Z, $General_B$ defines σ as the nearest integer value of the average number of labels associated with each instance. In case of ties (same frequency), the baseline chooses labels that maximize their co-occurrence with better ranked labels.

3. Multi-label feature selection

Although research on feature selection for multi-label data is relatively recent, it is already possible to note in the literature the application of several importance measures, as well as instances of all the multi-label feature selection dimensions previously mentioned. To obtain a wide and reproducible review of the literature, we conducted the systematic review method. Basic information of 74 publications gathered by the systematic review method is presented in the supplementary material available in `https://db.tt/0yO8zXcR`. This innovative systematic review on multi-label feature selection consists in one of the main contributions of this work.

By categorizing these publications according to the taxonomy previously mentioned, it is possible to observe, for example, that first-order and filter are prevalent approaches. This is partially explained because both have lower computational cost than other approaches. Furthermore, the filter approach is independent of multi-label learning algorithms, which can be useful in some cases. Regarding the multi-label scope dimension, there is a more balanced distribution among the hybrid, single-label and multi-label approaches.

Figure 2 shows the number of publications found by SR that were published from 1997 (total: 74 pa-

pers). As the last SR update was made in June 2014, few 2014 publications were found.

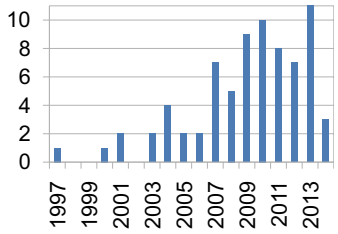

Figure 2: Number of related papers by publication year

Some related papers are highlighted in Ref. [10]. Among them, there are two alternative extensions to the ReliefF algorithm, which were published simultaneously with the *RF-ML* extension proposed by us. Although *RF-ML* has properties in common with the two literature extensions, its main differential consists in the unique combination of these properties for filter FS to support non-hierarchical ("flat") multi-label learning.

4. Problem transformation-based feature selection

The standard approach for multi-label FS is implementable within the *BR* approach. However, as mentioned, this approach has a disadvantage that may subsequently impair learning: the label dependency is often ignored. An alternative to reduce this problem would be to build labels from relations between the original labels and include the new labels during the FS task. Thus, even with the independent handling of all labels by *BR*, the new labels would be indirectly representing associations between labels.

This section describes the Label Construction for Feature Selection method (*LCFS*), which builds binary variables (new labels) from relations inherent to two or more labels [10]. The *LCFS* settings currently implemented consider relations between labels, featuring the method as a second-order strategy [18], which aims to reduce the *BR* disadvantage previously mentioned.

4.1. Feature selection via traditional problem transformation approaches

Two frequently used approaches for multi-label FS consist in *BR* and *LP* — Section 2.2. The basic idea is to transform the multi-label data into single-label data with one of these approaches before using a single-label feature selection algorithm. Regarding *BR*, partial results are combined in this work according to the Averaging strategy, which stood out in an experimental evaluation with 20 multi-label textual datasets [10]. In this strategy, the final importance value v_j of each attribute X_j is the average across the q values estimated for X_j in the q single-label problems.

4.2. Label construction for feature selection

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

The *LCFS* method consists of two steps, each one aiming to answer a different question:

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

Thus, setting *LCFS* involves choosing a strategy to select pairs of labels and a strategy to combine the labels within each pair. An additional parameter is the number of new labels q' to be built. The two method steps, described below, are illustrated in Ref. [10].

4.2.1. Selection

Given a set of labels $L = y_1, y_2, \ldots, y_q$ of the dataset D, *LCFS* chooses q' different pairs of labels (y_i, y_j),

$i \neq j$, according to a specific strategy. In this work, two label pairs are considered different if they do not have a common label. The idea is that the selected pairs capture some relation between labels that could be considered for FS.

LCFS supports different selection strategies, such as the Random Selection (*RS*) as well as heuristic strategies based on the number of instances labeled by each original single label (label frequency). In particular, two strategies ground on 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 labeled by both labels within a pair, allowing *LCFS* to select the q' first 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 the greatest value is selected and removed from the correspondent list. This procedure is repeated until selecting q' different pairs.

4.2.2. Generation

LCFS combines in this step both labels from all previously selected pairs (y_i, y_j), $i \neq j$, to construct the new labels y_{ij}. The idea is that the y_{ij} values represent a relation between y_i and y_j. In the end, all instances in D are labeled by the q original labels and by the q' new labels. *LCFS* supports different combination strategies between binary variables (labels). In this work, we used three simple logic operators to generate the values of the new labels of each instance in D:

- 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-occurrent labels. XNOR or coincidence function assigns the value 1 (one) for y_{ij} iff the labels y_i and y_j agree, whereas XOR does the opposite.

After generating q' new labels, datasets labeled by $q + q'$ labels can be submitted to the traditional *BR* approach for feature selection. Note that, by combining *BR* with *LCFS*, any single-label FS algorithm can be applied in a dataset augmented with second-order label information [18].

4.3. Experimental setting

In this section, the experimental evaluation compares multi-label feature selection methods based on problem transformation. These methods are organized into two groups:

- Group 1: FS algorithms based on the traditional *BR* and *LP* approaches;
- Group 2: settings of the proposed method *LCFS*.

Afterwards, an additional comparison between the best member of each group is conducted.

Regardless of the group, the considered FS methods estimate the importance of the features according to the measures Information Gain (*IG*) and ReliefF. These measures are frequently used in related work and are described in Ref. [10].

In Group 1, the combination between the *BR* and *LP* approaches with the importance measures *IG* and ReliefF yields four feature selection methods:

1. *IG* based on *LP* (*IG-LP*);

2. ReliefF based on *LP* (*RF-LP*);

3. *IG* based on *BR* (*IG-BR*);

4. ReliefF based on *BR* (*RF-BR*).

In Group 2, four *LCFS* settings that combine different Selection (**S**) and Generation (**G**) strategies are considered. All are configured to generate $q' = \frac{q}{2}$, *i.e.*, every single label is selected once if q is even, or a label is ignored if q is odd.

1. *LS-X*: **S**: LS; **G**: XOR or XNOR is chosen from the lists related to c_e and c_d;

2. *CS-A*: **S**: CS; **G**: AND;

3. *RS-A*: **S**: RS; **G**: AND;

4. *RS-X*: **S**: RS, **G**: XOR or XNOR is randomly selected.

From the final feature ranking found by a FS method, the subsets of the best features $X' \subset X$, $|X| = M$, $|X'| = 10\%M, 20\%M, \ldots, 90\%M$ are specified. These nine subsets are used to describe reduced versions of the original multi-label dataset, which are submitted to a learning algorithm. The idea is to evaluate the quality of the selected attributes in terms of classification performance.

As mentioned, the *BRkNN-b* learning algorithm is used in this work to evaluate FS methods. The main *BRkNN-b* and ReliefF parameter is the number of nearest neighbors k. We decided to set $k = 10$ (ten) for both, as adopted by default in Weka framework for ReliefF [17]. It should be emphasized, however, that other k values have been evaluated [12,13]. In Ref. [10], the values of the other parameters are described. The frameworks Weka and Mulan implement *BR*, *LP*, *BRkNN-b*, *IG* and ReliefF. Moreover, the *LCFS* method proposed by us is based on both frameworks.

Table 2 describes the 10 (ten) benchmark datasets used in the experiments. For each dataset, the table shows: the dataset name; number of instances (N); number of features (M); number of labels ($|L|$); Label Cardinality (LC), which is the average number of labels associated with each instance; Label Density (LD), which is the cardinality normalized by $|L|$; number of distinct multi-labels ($\sharp Diff$).

Table 2: Dataset description

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

In all experiments, the classifiers performance was estimated according to the evaluation measures described in Section 2.3. In particular, the estimates were obtained by the 10-fold cross-validation strategy with paired folds. All results are presented in detail in Ref. [10].

Besides $General_B$, we included random feature selection (*SAR*) as a reference in some experiments.

4.4. Results and discussion

In general, the performance of the classifiers built using the features chosen by each of the four FS algorithms from Group 1 was similar. A noticeable exception was verified in the 8-Magtag5k dataset. Figure 3 exemplifies this fact by showing the F-Measure performance of *BRkNN-b* classifiers (y axis) for each feature subset size (x-axis) in this data set. Note that the results related to *SAR* are marked with the magenta color.

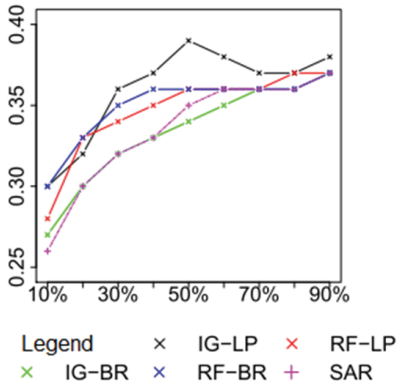

Figure 3: F-Measure performance of *BRkNN-b* classifiers built using the features selected by FS algorithms and *SAR* from the 8-Magtag5k dataset

In fact, there was no significant difference among these methods according to the Friedman's statistical test. This scenario held when random feature selection (*SAR*) was included as a reference, although this method was usually worse than the other ones. However, the *IG-LP* algorithm was noticeably better in most average rankings of the corresponding classifiers calculated by the test, while *IG-BR* reached similar achievement with Hamming Loss.

In order to improve the *IG-BR* competitiveness against *IG-LP*, we investigated the four *LCFS* settings previously mentioned in Group 2. *LCFS* considers relations between labels during the construction of new labels, which were then used for FS based on *IG-BR*. We also included *IG-BR* in this comparison as a reference. Once more, no significant difference among the methods was found according to the Friedman's test. Figure 4 gives one an idea of how *IG-BR* and the four *LCFS* settings led to similar results in terms of F-Measure.

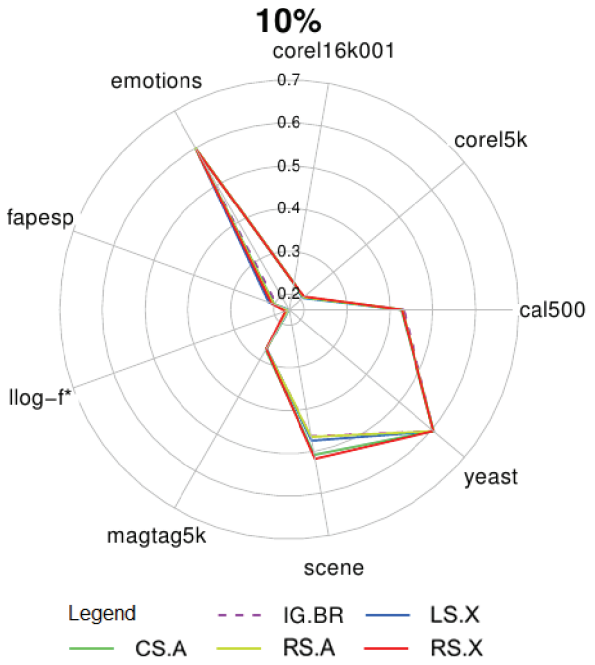

Figure 4: F-Measure performance of the *LCFS* settings and *IG-BR* in terms of the quality of *BRkNN-b* classifiers generated using $|X'| = 10\%M$ of the selected features

However, *RS-X* stood out because it was the unique *LCFS* setting that led to the improvement of some classifiers originally worse than the $General_B$ baseline. It also obtained the highest amount of classifiers with maximum average ranking calculated by the Friedman's test. Thus, we decided to compare specifically *IG-BR* and *RS-X*. By applying the Wilcoxon's test to compare them, we found significant differences with $\alpha = 0.05$, even when the k parameter from *BRkNN-b* was varied [13]. In all these cases, *RS-X* was statistically superior.

These achievements led us to compare *RS-X* with *IG-LP*. No significant difference was found between classifiers built using the features selected by each method. However, the average rankings of the classifiers related to the comparisons *RS-X* vs *IG-BR* and *IG-LP* vs *IG-BR*, indicate, for example, that *RS-X* highlighted when the number of features used to build the classifier was lower, which is considered a good result for the FS task [10]. The most remarkable improvement involves the feature subset size $|X'| = 20\%M$, in which *RS-X* supported the building of classifiers good for all evaluation measures used. This suggests that *RS-X* improves the *IG-BR* competitiveness in comparison with *IG-LP*. In fact, *IG-BR* only reached similar result against *IG-LP* with three times more attributes: $|X'| = 60\%M$. It should be emphasized that using *LCFS* before *IG-BR* considers some label relations, as *IG-LP* does, but avoids disadvantages of the *LP* approach, such as the restriction of the multi-labels considered to those that are contained in the training set. The good *RS-X* results were achieved, basically, by adding to *IG-BR* the computational cost corresponding to the application of the operator XOR or XNOR to generate new labels, as the selection of label pairs is random.

Due to these and other reasons, we conclude that *LCFS*, using the *RS-X* setting, makes *IG-BR* more competitive against the best representative elected from five multi-label FS algorithms based on *BR* and *LP*. Furthermore, the relative superiority showed by *IG-LP* and *RS-X* is an evidence that to consider label relations is important for multi-label FS based on problem transformation approaches.

5. ReliefF algorithm adaptation for multi-label feature selection

The main advantage of ReliefF, a traditional family of single-label FS algorithms, over strictly univariate importance measures such as *IG* is that it considers the effect of interacting features [8]. The pioneer algorithm in this family, Relief, is only able to evaluate features in binary data with no missing values or noise. ReliefF reduces these limitations and supports feature selection in multiclass data, while

RReliefF tackles data with numerical labels. These algorithms are best described in the seminal references and in Ref. [10].

Initially, Relief family algorithms had been applied in multi-label data by problem transformation approaches. However, ReliefF and RReliefF extensions able to perform multi-label FS directly, without any previous data transformation, have been recently published [7,9,11]. Two of these extensions were proposed by us: *RF-ML* and *RFM-ML*. Both extensions preserve ReliefF and RReliefF properties, such as the consideration of the effect of interacting features. They also originally combine characteristics of other contemporary extensions to conduct filter FS to support flat multi-label learning.

This section experimentally evaluates the extensions *RF-ML* and *RFM-ML* and compares them with algorithms that apply ReliefF and *IG* according to the *BR* and *LP* problem transformation approaches.

5.1. Proposed ReliefF-based extensions

We proposed two ReliefF-based extensions [12,10], which are briefly described in what follows.

5.1.1. RF-ML

Although *RF-ML* is similar to RReliefF, the extension proposed by us differs from the original algorithm by: (1) introducing a dissimilarity function $mld(.,.)$, which considers multiple labels simultaneously, and (2) searching for k multi-label nearest instances (neighbors). Thus, *RF-ML* forgoes the transformation of the multi-label problem into the single-label one. By doing so, it demands lower computational cost than the frequently used combination between the single-label ReliefF and the *BR* and *LP* problem transformation approaches.

It should be emphasized that *RF-ML* also preserves some ReliefF and RReliefF properties, such as the consideration of the effect of interacting features, by analyzing the dissimilarity between instances in the feature space regardless of the number of data labels.

As mentioned, the introduction of the dissimilarity function between multi-labels $mld(.,.)$ differentiates *RF-ML* from its predecessor RReliefF. Any dissimilarity measure between two sets can implement $mld(.,.)$, suggesting flexibility for multi-label FS. In this work, two measures were used for such purpose: normalized Hamming distance and Jaccard dissimilarity.

The Hamming Distance (HD) between two multi-labels Y_i and Y_j is defined as $|Y_i \cup Y_j| - |Y_i \cap Y_j|$, which is equivalent to the amount of distinct single labels between these multi-labels. Thus, the absence and the presence of labels are treated equally. The HD used in this work is normalized by q, the number of single labels in a data set.

On the other hand, the Jaccard Dissimilarity $JD(Y_i, Y_j)$ is useful, for example, for cases in which Y_i and Y_j are composed of few single labels, *i.e.*, when $|Y_i \cup Y_j| << q$. The idea is that the dissimilarity measure forgoes single labels absent in both multi-labels, *i.e.*, the measure takes into account only labels present in at least one of the multi-labels. Note that, as is the case with the normalized Hamming distance, the JD values range in the interval [0,1]. Both measures are described in Ref. [10].

5.1.2. RFM-ML

The main difference between *RF-ML* and *RFM-ML* is that the latter establishes a neighborhood between the instances E_i e E_j iff E_j is the nearest neighbor of E_i and vice-versa. Despite the more severe restrictions, as the number of mutual nearest neighbors $k' \leqslant k$, the mutual neighborhood has contributed to obtain good results in multi-label classification [1].

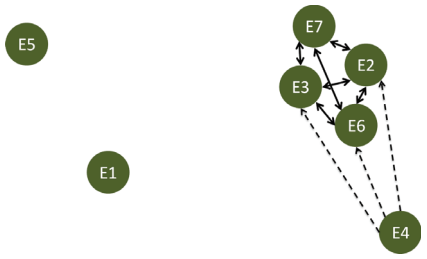

Figure 5: Illustrative dataset for *RF-ML*. If one considers $k = 3$ nearest neighbors, the instance E_4 has 3 (three) nearest neighbors, but no mutual nearest neighbor ($k' = 0$)

Sometimes, the number of mutual neighbors k' can be equal to zero — Figure 5. To deal with this issue, *RFM-ML* ignores instances with $k' = 0$, such that they do not affect the feature importance estimate.

5.2. Experimental setting

In the experimental evaluation performed in Ref. [10], we used the 10 (ten) benchmark datasets described in Section 4.3 and 45 synthetic datasets generated by a publicly available framework developed by collaborators [15]. In this paper, only the benchmark sets were considered. As Figure 6 shows, the experimental evaluations conducted consider a specific order to compare the following methods.

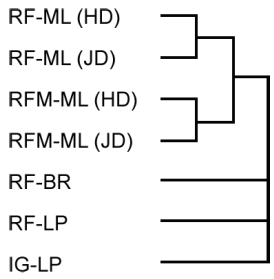

Figure 6: Illustration of the experimental setting regarding this section

- *RF-ML* using the Hamming distance measure, *i.e.*, *RF-ML* (HD);
- *RF-ML* using the Jaccard dissimilarity measure, *i.e.*, *RF-ML* (JD);
- *RFM-ML* using the Hamming distance measure, *i.e.*, *RFM-ML* (HD);
- *RFM-ML* using the Jaccard dissimilarity measure, *i.e.*, *RFM-ML* (JD);
- *RF-BR*, *RF-LP* e *IG-LP*.

In particular, this order enables us to choose the best *RF-ML* and *RFM-ML* setting for direct comparison with FS algorithms based on *IG* and ReliefF described in Section 4.3. It should be noted that *IG-LP* is included due to its prominence in the experimental evaluation regarding feature selection based on problem transformation — Section 4.4.

Most of the experimental setting regarding each of these methods is similar to the one used in Section 4. More information about both settings is described in Ref. [10].

5.2.1. Evaluation measure specific for FS

Motivated by the need to measure the stability of feature subsets returned by multiple executions of an algorithm, Kuncheva [4] proposed a similarity index. This measure satisfies some properties for the subsets S_1 and S_2, which have the same size, *i.e.*, $|S_1| = |S_2| = g$, $0 < g < M$. The monotonicity property, for example, indicates that the larger the intersection between S_1 and S_2, the greater the similarity value is.

5.3. Results and discussion

Regarding the comparison of the *RF-ML* and *RFM-ML* settings, the experimental evaluations indicated that settings based on Hamming distance led to significantly better results in terms of Hamming Loss, although the values achieved by this evaluation measure changed little. They also indicate that the use of the Jaccard dissimilarity measure obtained similar achievements in the remaining evaluation measures [10]. As *BRkNN-b* classifiers had more difficulty to perform well in terms of Hamming Loss and *RF-ML* always led to classifiers competitive or better than the ones derived from *RFM-ML*, the *RF-ML* (HD) setting was chosen for the remaining experimental evaluations.

Different from the findings from experimental evaluations in synthetic datasets [10], in which *RF-ML* was statistically superior in many cases, no significant difference was found ($\alpha = 0.05$) among *RF-ML* (HD), *RF-BR*, *RF-LP* and *IG-LP* in terms of the evaluation measures calculated in benchmark datasets. By taking into account the average rankings of the classifiers calculated by the Friedman's test, *IG-LP* was noticeably the best option in most of the cases, followed by *RF-LP* and *RF-ML* (HD). Nevertheless, there was larger equilibrium among them when the smallest feature subsets were used — $|X'| = 10\%$ and $|X'| = 20\%$. Although the evaluation measures considered in each scenario were different,

it must be emphasized that the prominence of *IG-LP* and *RF-LP* in benchmark datasets contrasts with the highlight of *RF-ML* and *RF-BR* in synthetic datasets. Thus, further studies were carried out to better evaluate these FS algorithms.

As mentioned, the Hamming Loss values changed little. To observe this, one can analyze the number of ties regarding each evaluation measure. In particular, a tie is verified in a FS experiment conducted in this work if there are at least two *BRkNN-b* classifiers with identical evaluation measure values. In this context, we built classifiers using a specific number of features selected from a particular dataset, although each classifier considers the output of a distinct FS algorithm. Altogether, 90 experiments were conducted — 9 (nine) feature subset sizes $\times$ 10 (ten) datasets. Table 3 highlights Hamming Loss as the measure with the highest number of ties.

Table 3: Percentage of ties regarding classifiers built from features selected by *IG-LP*, *RF-BR*, *RF-LP* and *RF-ML*

F-measure	*Hamming Loss*	*Accuracy*	F_b
81%	**99%**	86%	77%

However, if the precision of the Hamming Loss values were increased by two significant digits, the correspondent number of ties would reduce down to 2% and the proposed *RF-ML* (HD) would stand out for the smallest feature subsets. Furthermore, this extension would become the one with the highest number of *BRkNN-b* classifiers with the best average ranking according to the evaluation measure. This suggests a possible relation between Hamming Loss and the Hamming distance measure, as *RF-ML* did not achieve such prominence when using the Jaccard dissimilarity measure.

An additional study found that, except for $|X'| = 60\%M$, both *IG-LP* and *RF-ML* (HD) showed good ability to reduce the dimensionality of the benchmark datasets without significant loss of predictive performance. In the specific case of $|X'| = 60\%M$, only *IG-LP* preserved this ability.

In addition to the previous analyzes on predictive performance, another experimental evaluation was conducted to investigate the support that *RF-ML* (HD) and FS algorithms based on problem transformation can provide for *BRkNN-b* and some non lazy classification algorithms [10]. The results indicate that *RF-ML* led to significantly worse results in only 13% of the cases, mainly related to *BRkNN-b*. However, it should be emphasized that *RF-ML* was outperformed by an algorithm that requires previous data discretization and shows relatively high computational complexity.

The latest complementary study evaluated FS algorithms by using the classifier independent similarity index for feature subsets described in Section 5.2. Table 4 shows the index values averaged across the 10 benchmark datasets for the feature subset size $|X'| = 10\%M$, while the remaining results are presented in the supplementary material available in `https://db.tt/0yO8zXcR`. The smallest average similarities are marked in bold.

Table 4: Average similarity of the feature subsets X', $|X'| = 10\%M$, found by *IG-LP*, *RF-BR*, *RF-LP* and *RF-ML*

	IG-LP	*RF-BR*	*RF-LP*	*RF-ML*
IG-LP		0.40	0.51	**0.22**
RF-BR	0.40		0.69	**0.28**
RF-LP	0.51	0.69		**0.30**
RF-ML	**0.22**	**0.28**	**0.30**	

Regardless of the number of selected features, *RF-ML* achieved lower average similarity than the other algorithms in the benchmark datasets. The *RF-ML* prominence in this measure could motivate its inclusion in ensembles of multi-label FS algorithms.

As mentioned, it is important to note that *IG-LP* and *RF-LP*, highlighted in some cases, have disadvantages inherent in the *LP* approach, such as restricting the multi-labels considered to those that are contained in the training set only. This can hinder the quality of the selected features and learning in practice. On the other hand, *RF-ML* avoids these disadvantages, is faster than *RF-LP* and *RF-BR* and forgoes data discretization. In addition, *RF-ML* considers label relations by directly dealing with the multi-label problem without any data transformation.

6. Conclusion

This work proposes a few multi-label feature selection algorithms that follow the filter approach and consider label relations. Three contributions were considered in the previous sections.

The systematic review allowed us to widely explore the literature on multi-label FS. A positive aspect was the synthesis of 74 related papers, which provides a contemporary panorama of the research area. Regarding the objective, the SR showed that considering label dependency for FS is still incipient.

Proposing *LCFS* as an alternative to include label relations information into the frequently used *BR* problem transformation approach for FS was an innovative way to achieve the work objective. This method can be applied when one is interested in taking into account second-order label relations for feature selection, regardless of the importance measure used. Regarding the objective, the corresponding experimental evaluation contributed to obtain evidence that considering label relations is important for multi-label FS based on problem transformation approaches. A *LCFS* limitation consists in the temporary increase in the number of single labels in a multi-label data during the feature selection task, which influences the complexity of *BR*-based algorithms. As future work, we plan to use label selection procedures to remove unimportant labels generated by *LCFS*.

Another contribution consists in the proposal of the *RF-ML* and *RFM-ML* extensions for ReliefF and RReliefF. These extensions perform FS directly, without any problem transformation. Thus, the multi-labels contained in the dataset and the relations among the corresponding single labels are preserved. Unlike *LCFS*, this alternative is specific to the importance measure inherent to the Relief family of algorithms, which considers the effect of interacting features and avoids the discretization of numerical data. The main change inherent to both extensions is the introduction of the dissimilarity between label sets (multi-labels) to model the probability that two instances have different labeling. Although *RF-ML* avoids disadvantages of the *LP* approach, such as restricting the multi-labels considered to those that are contained in the training set, this did not lead to significant superiority in terms of predictive performance in the benchmark datasets. Alternatives that could be further studied include to investigate the relation between *RF-ML* dissimilarity measures and multi-label evaluation measures, as well as to empirically evaluate other algorithm parameters. In addition, a further study of data properties by using, for example, exploratory data analysis, would also be useful to better understand the results.

An inherent limitation of the experimental evaluations conducted in this work is the difficulty to assess multi-label classifiers. Different from single-label learning evaluation, the multi-label learning assessment also needs to take into account partially correct predictions. Multiple evaluation measures should be used, such as exemplified by this work, as each one considers different classification issues. In this scenario, tools that support the analysis of multiple multi-label measures, such as $General_B$, should be better investigated in future work.

The implementations developed in this work are publicly available in `http://goo.gl/sPMHmm` and in `http://goo.gl/pSwzgp`.

Acknowledgments

This research was supported by the São Paulo Research Foundation FAPESP, grant 2011/02393-4. We would like to thank M. C. Monard for her primordial support in this work.

References

1. Cherman, E.A., Spolaôr, N., Valverde-Rebaza, J.C., Monard, M.C.: Lazy multi-label learning algorithms based on mutuality strategies. Journal of Intelligent & Robotic Systems pp. 1–16 (2014). DOI 10.1007/s10846-014-0144-4
2. Dembczynski, K., Waegeman, W., Cheng, W., Hüllermeier, E.: On label dependence and loss minimization in multi-label classification. Machine Learning **88**, 5–45 (2012). DOI 10.1007/s10994-012-5285-8
3. Han, J., Kamber, M.: Data mining: concepts and techniques. Morgan Kaufmann (2011)
4. Kuncheva, L.I.: A stability index for feature selection. In: IASTED International Multi-Conference: Artificial Intelligence and Ap-

plications, pp. 390–395 (2007). Available in: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.101.6458&rank=1
5. Liu, H., Motoda, H.: Computational Methods of Feature Selection. Chapman & Hall/CRC (2008)
6. Metz, J., Abreu, L.F., Cherman, E.A., Monard, M.C.: On the estimation of predictive evaluation measure baselines for multi-label learning. In: J. Pavón, N.D. Duque-Méndez, R. Fuentes-Fernández (eds.) Advances in Artificial Intelligence - IBERAMIA 2012, *Lecture Notes in Computer Science*, vol. 7637, pp. 189–198. Springer (2012). DOI 10.1007/978-3-642-34654-5_20
7. Pupo, O.G.R., Morell, C., Soto, S.V.: ReliefF-ML: An extension of relieff algorithm to multi-label learning. In: J. Ruiz-Shulcloper, G. Sanniti di Baja (eds.) Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, *Lecture Notes in Computer Science*, vol. 8259, pp. 528–535. Springer Berlin Heidelberg (2013). DOI 10.1007/978-3-642-41827-3_66
8. Robnik-Šikonja, M., Kononenko, I.: Theoretical and empirical analysis of ReliefF and RReliefF. Machine Learning **53**(1–2), 23–69 (2003). DOI 10.1023/A:1025667309714
9. Slavkov, I., Karcheska, J., Kocev, D., Kalajdziski, S., Džeroski, S.: Extending ReliefF for hierarchical multi-label classification. In: Workshop on New Frontiers in Mining Complex Patterns - European Conference on Machine Learning/Principles and Practice of Knowledge Discovery in Databases, pp. 156–167 (2013). Available in: http://www.di.uniba.it/ ceci/micFiles/NFMCP2013
10. Spolaôr, N.: Feature selection for multi-label learning (in Portuguese). Phd thesis, University of São Paulo - Brazil (2014). Available in: http://www.teses.usp.br/teses/disponiveis/55/55134/tde-25032015-160505/pt-br.php
11. Spolaôr, N., Cherman, E.A., Monard, M.C., Lee, H.D.: ReliefF for multi-label feature selection. In: Brazilian Conference on Intelligent Systems, pp. 6–11 (2013). DOI 10.1109/BRACIS.2013.10
12. Spolaôr, N., Monard, M.C.: Evaluating ReliefF-based multi-label feature selection algorithm. In: A.L.C. Bazzan, K. Pichara (eds.) Advances in Artificial Intelligence – IBERAMIA 2014, *Lecture Notes in Computer Science*, vol. 8864, pp. 194–205. Springer International Publishing (2014). DOI 10.1007/978-3-319-12027-0_16
13. Spolaôr, N., Monard, M.C., Tsoumakas, G., Lee, H.D.: Label construction for multi-label feature selection. In: Brazilian Conference on Intelligent Systems, pp. 1–6. IEEE (2014). DOI 10.1109/BRACIS.2014.52
14. Spyromitros, E., Tsoumakas, G., Vlahavas, I.: An empirical study of lazy multilabel classification algorithms. In: Hellenic conference on Artificial Intelligence, pp. 401–406. Springer-Verlag (2008). DOI 10.1007/978-3-540-87881-0_40
15. Tomás, J.T., Spolaôr, N., Cherman, E.A., Monard, M.C.: A framework to generate synthetic multi-label datasets. Electronic Notes in Theoretical Computer Science **302**(0), 155–176 (2014). DOI 10.1016/j.entcs.2014.01.025
16. Tsoumakas, G., Katakis, I., Vlahavas, I.P.: Mining multi-label data. In: O. Maimon, L. Rokach (eds.) Data Mining and Knowledge Discovery Handbook, pp. 667–685. Springer US (2010). DOI 10.1007/978-0-387-09823-4_34
17. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann (2011)
18. Zhang, M.L., Zhou, Z.H.: A review on multi-label learning algorithms. IEEE Transactions on Knowledge and Data Engineering **26**(8), 1819–1837 (2014). DOI 10.1109/TKDE.2013.39