Label construction for multi-label feature selection

Brazilian Conference on Intelligent Systems, 3th, 2014, São Carlos.
http://www.producao.usp.br/handle/BDPI/48632

Downloaded from: Biblioteca Digital da Produção Intelectual - BDPI, Universidade de São Paulo
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 $E_i = (x_i, Y_i), i = 1 \ldots N$. Each example (instance) $E_i$ is associated with a feature vector $x_i = (x_{i1}, x_{i2}, \ldots, x_{iM})$ 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 $E = (x, ?)$, is capable of accurately predicting its multi-label $Y$, i.e., $H(E) \rightarrow Y$.

<table>
<thead>
<tr>
<th>Table I</th>
<th>MULTI-LABEL DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$X_1$</td>
</tr>
<tr>
<td>$E_1$</td>
<td>$x_{11}$</td>
</tr>
<tr>
<td>$E_2$</td>
<td>$x_{21}$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$E_N$</td>
<td>$x_{N1}$</td>
</tr>
</tbody>
</table>

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 evaluation measures, which can optimize different aspects, is out of the scope of this work and can be found in [2]. In what follows, we describe the four measures used in this work.

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

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

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

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

In addition, Micro-averaged F-measure ($F_{\mu}$), defined by Equation 4, is a label-based measure, where $TP_{y_j}$, $FP_{y_j}$, $TN_{y_j}$ and $FN_{y_j}$ represent, respectively, the number of true/false positives/negatives for a label $y_j \in L$.

$$ F_{\mu}(H, D) = \frac{2 \sum_{j=1}^{q} TP_{y_j}}{2 \sum_{j=1}^{q} TP_{y_j} + \sum_{j=1}^{q} FP_{y_j} + \sum_{j=1}^{q} FN_{y_j}} \tag{4} $$

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 types of interaction: wrapper, embedded and filter [1]. In particular, the first two approaches involve strong interaction. On the other hand, filters use general properties of the dataset to remove unimportant features from it, regardless of the learning algorithm. Thus, although the features chosen may not be the best ones for a specific learning algorithm, filter FS is performed only once for all learning algorithms. 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.
of feature has times, the L 44 (61.11%) D with the set of single D j D ELATED WORK ON MULTI:

consists in the set of examples where (5) y LCFS X 50 (69.44%) 9 (12.50%) datasets. LCFS #publications (%)

y L (10 (13.89%), X 6 (8.33%) new is L j has the value i entropy D q 1 and 72 y approach from summarizes the 72 publica-
y- y entropy and the weighted sum of the entropy of each subset D new single labels by combining the original
LCFS CS should be chooses illustrates these steps for LCFS y different pairs. and i 1 c y q LCFS different pairs of labels method consists of two steps, each one con-
q of the j LCFS i.e. 1

In other words, the IG of feature X j, j = 1 . . . M, calculates the difference between the entropy of the dataset D and the weighted sum of the entropy of each subset D e where D e consists in the set of examples where X j has the value v. Therefore, if X j has 10 distinct values1 in D, the sum would be applied to 10 different D e 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

<table>
<thead>
<tr>
<th>categorization</th>
<th>approach</th>
<th>#publications (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>order of label dependence</td>
<td>first-order</td>
<td>44 (61.11%)</td>
</tr>
<tr>
<td></td>
<td>second-order</td>
<td>10 (13.89%)</td>
</tr>
<tr>
<td></td>
<td>high-order</td>
<td>9 (12.50%)</td>
</tr>
<tr>
<td></td>
<td>hybrid</td>
<td>3 (4.17%)</td>
</tr>
<tr>
<td></td>
<td>unrecognized</td>
<td>6 (8.33%)</td>
</tr>
<tr>
<td>interaction with the learning</td>
<td>filter</td>
<td>30 (69.44%)</td>
</tr>
<tr>
<td>algorithm</td>
<td>embedded</td>
<td>10 (13.89%)</td>
</tr>
<tr>
<td></td>
<td>wrapper</td>
<td>7 (9.72%)</td>
</tr>
<tr>
<td></td>
<td>unrecognized</td>
<td>5 (6.94%)</td>
</tr>
</tbody>
</table>

As can be observed, filters and first-order strategies have been the most usual choices in multi-label FS. Differently from the 10 second-order approach publications, the proposed method, described next, pioneers label construction.

IV. THE PROPOSED METHOD: LCFS

Given a multi-label dataset D with the set of single labels L = {y1, y2, y3, . . . , y q }, the main idea of LCFS is to construct q’ new single labels by combining the original labels within pairs (yi, y j), i  j, yi L and y j L. In each iteration, LCFS selects a pair of labels (yi, 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 (yi, 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](http://example.com/figure1.png)

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 = {y1, y2, y3, . . . , y q } of the dataset D, LCFS chooses q’ different pairs of labels2 (yi, y j), i  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 i e, 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 (yi, y j), i  j, to construct the new

1Discretization is applied to numerical features before using IG.

2In 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:

\[
\text{AND : } y_{ij} = 1 \text{ if } y_i = 1 \text{ and } y_j = 1; \quad y_{ij} = 0 \text{ otherwise.}
\]

\[
\text{OR : } y_{ij} = 1 \text{ if } y_i \neq y_j; \quad y_{ij} = 0 \text{ otherwise.}
\]

\[
\text{XNOR: } y_{ij} = 1 \text{ if } y_i \neq y_j; \quad y_{ij} = 0 \text{ otherwise.}
\]

The AND operator clearly highlights co-occurring labels. XNOR, also known as the coincidence function, assigns the value 1 to \( y_{ij} \) if 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 \( BRENN-b \) to evaluate the quality of the features selected, as lazy algorithms are sensitive to irrelevant features. \( BRENN-b \), 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 \( X' \subset X \), \( |X'| = 10\%, 20\%, \ldots, 90\% \) of ranked by each FS method are used to describe the dataset, which is submitted to \( BRENN-b \).

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

\( \text{RS-X} \): 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 \( q' = \left\lfloor \frac{q}{2} \right\rfloor \), 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 (\( T \)); 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>
<thead>
<tr>
<th>Dataset</th>
<th>Domain</th>
<th>N</th>
<th>M</th>
<th>Type</th>
<th>L</th>
<th>LC</th>
<th>LD</th>
<th>#Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-Cat509</td>
<td>music</td>
<td>502</td>
<td>68</td>
<td>numeric</td>
<td>174</td>
<td>26.084</td>
<td>0.150</td>
<td>502</td>
</tr>
<tr>
<td>2-Corel5k</td>
<td>image</td>
<td>5000</td>
<td>499</td>
<td>discrete</td>
<td>374</td>
<td>3.522</td>
<td>0.009</td>
<td>3175</td>
</tr>
<tr>
<td>3-Corel16001</td>
<td>image</td>
<td>13766</td>
<td>500</td>
<td>discrete</td>
<td>153</td>
<td>2.859</td>
<td>0.019</td>
<td>4803</td>
</tr>
<tr>
<td>4-Emotions</td>
<td>music</td>
<td>593</td>
<td>72</td>
<td>numeric</td>
<td>6</td>
<td>1.869</td>
<td>0.311</td>
<td>27</td>
</tr>
<tr>
<td>5-Fapesp</td>
<td>text</td>
<td>322</td>
<td>153</td>
<td>numeric</td>
<td>677</td>
<td>1.774</td>
<td>0.027</td>
<td>206</td>
</tr>
<tr>
<td>6-Genbase*</td>
<td>biology</td>
<td>57</td>
<td>57</td>
<td>numeric</td>
<td>30</td>
<td>1.252</td>
<td>0.046</td>
<td>32</td>
</tr>
<tr>
<td>7-Llog-f*</td>
<td>text</td>
<td>1253</td>
<td>1004</td>
<td>discrete</td>
<td>75</td>
<td>1.375</td>
<td>0.018</td>
<td>303</td>
</tr>
<tr>
<td>8-Magtag5k*</td>
<td>music</td>
<td>5260</td>
<td>68</td>
<td>numeric</td>
<td>136</td>
<td>4.839</td>
<td>0.038</td>
<td>4163</td>
</tr>
<tr>
<td>9-Scene</td>
<td>image</td>
<td>2407</td>
<td>294</td>
<td>numeric</td>
<td>6</td>
<td>1.074</td>
<td>0.179</td>
<td>15</td>
</tr>
<tr>
<td>10-Text</td>
<td>biology</td>
<td>4217</td>
<td>103</td>
<td>numeric</td>
<td>14</td>
<td>4.237</td>
<td>0.303</td>
<td>198</td>
</tr>
</tbody>
</table>

Excerpt for datasets 5-Fapesp and 8-Magtag5k, the other datasets are available in the Mulan\(^3\) 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: \( \text{LS-X}, \text{RS-A}, \text{RS-S} \) and \( \text{RS-X} \).

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

\(3\)http://mulan.sourceforge.net/datasets.html
\(4\)http://meka.sourceforge.net/datasets
\(5\)The dataset can be obtained from the authors.
\(6\)http://ti.dl.fcc.ule.pt/~magtag5k.zip

250
The $F$-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 datasets augmented by using the four LCFS settings considering 10% up to 90% of the features selected (5 FS methods × 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**

<table>
<thead>
<tr>
<th>$X'$</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
</tr>
</thead>
<tbody>
<tr>
<td>F-measure</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>x</td>
<td>o</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Hamming loss</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>x</td>
<td>x</td>
<td>o</td>
<td>+</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Accuracy</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>x</td>
<td>x</td>
<td>o</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>$F_b$</td>
<td>*</td>
<td>*</td>
<td>x</td>
<td>*</td>
<td>x</td>
<td>o</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

As can be observed, RS-X often achieves the best average rankings, specially when $|X'| < \frac{N}{2}$, 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.

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., $F-measure/\text{Hamming loss}/\text{Accuracy}/F_b$, 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 $\star$), or at most 5% worse than the ones of the classifiers using AF (indicated by $\star\star$). The symbol 0 indicates the other cases.
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-log-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 (\(\chi/\chi/\chi/\chi\)).

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 LCFS strategies based on label weighting [15] and apply Exploratory Data Analysis to understand better the results from specific datasets.

Acknowledgment

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

References


