Neurocomputing 326–327 (2019) 39–53

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Dealing with difficult minority labels in imbalanced mutilabel data sets

Francisco Charte[a,*], Antonio J. Rivera[b], María J. del Jesus[b], Francisco Herrera[a]

[a] Department of Computer Science and A.I., University of Granada, Granada 18071, Spain
[b] Department of Computer Science, University of Jaén, Jaén 23071, Spain

ARTICLE INFO

Article history:
Received 25 November 2015
Revised 11 June 2016
Accepted 1 August 2016
Available online 14 September 2017

Keywords:
Multilabel classification
Imbalanced classification
Preprocessing algorithms
Label concurrence

ABSTRACT

Multilabel classification is an emergent data mining task with a broad range of real world applications. Learning from imbalanced multilabel data is being deeply studied latterly, and several resampling methods have been proposed in the literature. The unequal label distribution in most multilabel datasets, with disparate imbalance levels, could be a handicap while learning new classifiers. In addition, this characteristic challenges many of the existent preprocessing algorithms. Furthermore, the concurrence between imbalanced labels can make harder the learning from certain labels. These are what we call *difficult* labels. In this work, the problem of difficult labels is deeply analyzed, its influence in multilabel classifiers is studied, and a novel way to solve this problem is proposed. Specific metrics to assess this trait in multilabel datasets, called *SCUMBLE* (*Score of ConcUrrence among iMBalanced LabEls*) and *SCUMBLELbl*, are presented along with REMEDIAL (*REsampling MultilabEl datasets by Decoupling highly ImbAlanced Labels*), a new algorithm aimed to relax label concurrence. How to deal with this problem using the R mldr package is also outlined.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Multilabel classification (MLC) [1,2] models are designed to predict the subset of labels associated to each instance in a multilabel dataset (MLD), instead of only one class as traditional classifiers do. It is a task useful in fields such as automated tag suggestion [3], protein classification [4], and object recognition in images [5], among others. Many different methods have been proposed lately to accomplish this problem.

The number of instances in which each label appears is not homogeneous. In fact, most MLDs show big differences in label frequencies. This peculiarity is known as imbalance [6], and it has been profoundly studied in traditional classification. In the context of MLC, several proposals to deal with imbalanced MLDs [7–15] have been made lately. Despite these efforts, there are still some aspects regarding imbalanced learning in MLC that would need additional analysis.

Resampling techniques are commonly used in with traditional (non-multilabel) datasets [16] to balance their class distributions, hence they are an obvious choice to face the same problem with MLDs. Notwithstanding, the nature of MLDs can be a challenge for resampling algorithms. In this paper, we will show how a specific characteristic of these datasets, the joint presence in the same instance of labels with different frequencies, could prevent the goal of these algorithms.

We hypothesized[1] that this symptom, the concurrence among imbalanced labels, would influence the resampling algorithms behavior. Specifically, the minority labels which jointly appear with majority ones would be difficult labels. In order to deal with this problem we propose to face it in two phases:

- Firstly, the concurrence problem has to be assessed. For doing so, two new metrics, named *SCUMBLE* (*Score of ConcUrrence among iMBalanced LabEls*) and *SCUMBLELbl*, designed explicitly to assess this causality, will be proposed. Its effectiveness will be experimentally demonstrated.
- Secondly, an algorithm specifically designed to preprocess MLDs affected by this problem would be needed. A such method, called REMEDIAL (*REsampling MultilabEl datasets by Decoupling highly ImbAlanced Labels*), will be introduced, and its performance will be empirically tested.

* Corresponding author.
E-mail addresses: fcharte@ujaen.es, francisco@fcharte.com (F. Charte), arivera@ujaen.es (A.J. Rivera), mjjesus@ujaen.es (M.J. del Jesus), herrera@ugr.es (F. Herrera).

[1] This paper is an extended version of our previous work [17] from HAIS'14, including additional metrics, a deeper analysis, and an algorithm aimed to solve the described problem. The proposed solutions have been implemented in R, and the software package containing them is also described.

http://dx.doi.org/10.1016/j.neucom.2016.08.158
0925-2312/© 2017 Elsevier B.V. All rights reserved.

The *SCUMBLE* measure was conceived aiming to know how difficult would be to work with a certain MLD for resampling algorithms. Its goal is to appraise the concurrence among imbalanced labels, giving as result a score easily interpretable. This score will be in the range [0,1]. A low score would denote an MLD with not much concurrence among imbalanced labels, whereas a high one would evidence the opposite case. Our hypothesis is that the lower the score obtained, the better the resampling algorithms would work. *SCUMBLELbl* complements the former metric, allowing to know which labels are more affected by this problem. The less frequent labels with a high *SCUMBLELbl* would be specially difficult cases.

Once the presence of the concurrence problem has been stated, the idea of how to deal with it naturally arises. For this reason the algorithm REMEDIAL, a specific method able to reduce concurrence among imbalanced labels, is also introduced. REMEDIAL works by decoupling imbalanced labels through an editing and oversampling approach. It is a resampling algorithm, since it produces new data samples. At the same time, it also edits existent instances. However, it does not change the number of times that each label appears in the dataset. The details about REMEDIAL and how to use this algorithm along with the proposed metrics, relying on the mldr R package [18], will also be explained.

The rest of this paper is structured as follows. Section 2 offers a brief introduction to MLC, as well as a description on how the learning from imbalanced MLDs has been faced. In Section 3, the problem of concurrence among imbalanced labels in MLDs will be defined, and how to assess this concurrence using the proposed metrics will be explained. The algorithm REMEDIAL is described in Section 4. Section 5 portraits the experimental framework used, as well as the obtained results from experimentation. Finally, Section 6 will offer the conclusions.Appendix A describes how to assess the label concurrence level and how to apply the REMEDIAL algorithm thorough a specific software package developed by the authors.

2. Preliminaries

In this section, a concise introduction to multilabel classification is offered, along with a description on how the learning from imbalanced MLDs has been faced until now.

2.1. Multilabel classification

Currently, there are many domains [4,5,19–22] in which each data pattern is not associated exclusively to one class, but to a group of them. In this context, the classes are named labels, and the set of labels that belongs to a data sample is called labelset. Let D be an MLD, D_i the ith instance, and L the full set on labels on D. The goal of a multilabel classifier is to predict a set $Z_i \subseteq L$ with the labelset for D_i.

Multilabel classification has been traditionally faced through two different approaches [23]. The first one, called data transformation, aims to produce binary or multiclass datasets from an MLD, allowing the use of non-MLC algorithms. The second, known as algorithm adaptation, has the goal of adapting established algorithms to natively work with MLDs. The two most common transformation methods are Binary Relevance (BR) [24] and Label Powerset (LP) [25]. The former produces several binary datasets from an MLD, usually one for each label or one for each label pair [26]. The latter transforms the MLD into a multiclass dataset, taking each labelset as class identifier. Regarding adapted algorithms, the number of proposals is quite high. There are multilabel KNN classifiers such as ML-kNN [27], multilabel trees based on C4.5 [28], and multilabel ANNs such as [29], as well as a profusion of algorithms based on ensembles of BR and LP classifiers. A recent review on multilabel classification algorithms can be found in [2].

Thus far, most proposed multilabel characterization metrics are focused in assessing the number of labels and labelsets. The most common ones are the total number of labels $|L|$, label cardinality (*Card*), which is the average number of labels per instance, and label density, obtained as $Card/|L|$.

2.2. Learning from imbalanced data

Imbalanced learning is a well-known problem in traditional classification [30–33], having been faced through three main approaches. First, by way of algorithmic adaptations of existent classifiers, the imbalance is taken into account in the classification process. Second, the preprocessing approach aims to balance class distributions by way of data resampling, creating (oversampling) or removing (undersampling) data samples. Third, cost sensitive classification is a combination of the two previous approaches. The data resampling approach has the advantage of being classifier independent, and its effectiveness has been proven in many scenarios.

In the MLC field, both the algorithmic adaptation and the data resampling approaches have been applied. The former is present in [7,8,11], while the latter appears in [10,12–15]. There are also proposals based on the use of ensemble of classifiers, such as [9].

When it comes to assessing the imbalance level in MLDs, the metrics in Eqs. (1) and(2) are proposed in [34]. Let D be an MLD, L the full set of labels in it, y the label being analyzed, and Y_i the labelset of ith instance in D. In Eq. (1) the symbol $[\![]\!]$ denotes de Iverson bracket, which returns 1 if the expression inside it is true or 0 otherwise. *IRLbl* is a measure calculated individually for each label. The higher is the *IRLbl* the larger would be the imbalance, allowing to know which labels are in minority or majority. *MeanIR* is the average *IRLbl* for an MLD. It is useful to estimate the global imbalance level.

$$IRLbl(y) = \frac{\max_{y' \in L} \left(\sum_{i=1}^{|D|} [\![y' \in Y_i]\!] \right)}{\sum_{i=1}^{|D|} [\![y \in Y_i]\!]}. \tag{1}$$

$$MeanIR = \frac{1}{|L|} \sum_{y \in L} IRLbl(y). \tag{2}$$

Even though the previously cited proposals for facing imbalanced learning in MLC achieve some good results, their behavior is heavily influenced by MLDs characteristics such as the imbalance levels, measured by means of the previous metrics, or the concurrence among imbalanced labels, which will be described later. In the following we will focus in this topic, specifically in regard to data resampling solutions.

2.3. Related work

In general, resampling methods aimed to work with non-MLDs can be divided into two categories, oversampling algorithms and undersampling algorithms. The former technique produces new samples with the minority class, while the latter removes instances linked to the majority class. The way in which the samples to be removed or reproduced are chosen can also be grouped into two categories, random methods and heuristic methods. Since this kind of datasets use only one class per instance, the previous techniques effectively balance the distribution of classes. However, this is not always true when dealing with MLDs. Moreover, most MLDs have more than one minority and one majority label.

The preceding approaches have been migrated to the multilabel scenario at some extent, giving as result proposals such as the following:

- *Random undersampling*: Two multilabel random undersampling algorithms are presented in [12], one of them based on the LP transformation (LP-RUS) and another one on the *IRLbl* measure (ML-RUS). The latter determines what labels are in minority, by means of their *IRLbl*, and avoids removing samples in which they appear.
- *Random oversampling*: The same paper [12] also proposes two random oversampling algorithms, called LP-ROS and ML-ROS. The former is based on the LP transformation, while the latter relies on the *IRLbl* measure. Both take into account several minority labels, and generate new instances cloning the original labelsets.
- *Heuristic undersampling*: In [14] a method to undersample MLDs following the ENN (*Edited Nearest Network*) rule was presented. The instances are not randomly chosen, as in LP-RUS or ML-RUS, but carefully selected after analyzing their *IRLbl* and the differences with their neighborhood.
- *Heuristic oversampling*: The procedure proposed in [13] is based on the original SMOTE algorithm. First, instances of an MLD are chosen using different criteria, then the selected samples are given as input to SMOTE, producing new samples with the same labelsets. In [15] a more sophisticated approach is presented, with a multilabel version of SMOTE, called MLSMOTE, able to produce synthetic samples whose labelsets are generated from those of the nearest neighbors, instead of cloning them.

A major disadvantage in some of these algorithms is that they always work over full labelsets, cloning the set of labels in existent samples or completely removing them. Although this approach can benefit some MLDs, in other cases the result can be counterproductive depending on the MLD traits.

The aforementioned multilabel resampling algorithms will not have an easy work while dealing with MLDs which have a high *SCUMBLE* level. Undersampling algorithms can produce a loss of essential information, as the samples selected for removal because majority labels appear in them can also contain minority labels. In the same way, oversampling algorithms limited to cloning the labelsets, such as the proposals in [12,13], can be also increasing the presence of majority labels.

3. Imbalanced MLDs and resampling algorithms behavior

Most traditional resampling methods do their job by removing instances with the most frequent class, or creating new samples from instances associated to the least frequent one. Since each instance can belong to one class only, these actions would effectively balance the classes frequencies. However, this is not necessarily true when working with MLDs.

3.1. Concurrence among imbalanced labels in MLDs

The instances in a MLD are usually associated simultaneously to two or more labels. It is entirely possible that one of those labels is the minority label, while other is the majority one. In the most extreme situation, all the appearances of the minority label could be jointly with the majority one, into the same instances. This will make the minority label specially difficult to classify by any MLC algorithm, as most of them tend to be biased to the majority ones. In practice the scenario would be more complicated, as commonly there are more than one minority/majority label in an MLD. Therefore, the potential existence of instances associated to minority and majority labels at once is very high. This fact is what we call concurrence among imbalanced labels.

A multilabel oversampling algorithm that clones minority labels, such as the proposed in [12], or that generates new samples from existing ones preserving the labelsets, as is the case in [13], could be also increasing the number of instances associated to majority labels. Thus, the imbalance level would be hardly reduced if there is a high level of concurrence among imbalanced labels. In the same way, a multilabel undersampling algorithm designed to remove instances from the majority labels, such as the proposed in [12], could inadvertently cause also a loss of samples associated to the minority ones. In both cases, difficult labels (those which are in minority and have a high concurrence with majority ones) will be the most harmed by the classifier.

The ineffectiveness of these resampling methods, when they are used with certain MLDs, would be noticed once the preprocessing is applied and the classification results are evaluated. This process will need computing power and time. For that reason, it would be desirable to know in advance the level of concurrence among imbalanced labels that each MLD suffers, saving these valuable resources.

3.2. Metrics to assess the concurrence level

The concurrence of labels in an MLD can be visually explored in some cases, as shown in Figs. 1 and 2. Each arc represents a label, being the arc's length proportional to the number of instances in which this label is present. The first (Fig. 1) diagram corresponds to the genbase dataset. At the position of twelve o'clock appears a label called *P750* which is clearly a minority label. All the samples associated to this label also contains *P271*, another minority label. The same situation can be seen with label *P154*. These minority labels have not necessarily to be difficult labels. By contrast, in the yeast MLD (Fig. 2) is easy to see that the samples associated to minority labels, such as *Class14* and *Class9*, always appear together with one or more majority labels. At first sight, that the concurrence between imbalanced labels is higher in yeast than in genbase, and that the former contains some difficult labels while the latter does not, could be concluded. However, this visual exploratory technique is not useful with MLDs having more than a few dozens labels, because the diagram would be hardly legible.

Existing metrics previously described (see Section 2.2), such as *IRLBL* and *MeanIR*, assess the imbalance level of the labels, i.e., the relative frequency of each label with respect to the most common one and the average frequency. However, none of them allows to know if minority labels appear in their own or jointly with majority ones. The *SCUMBLE* metric proposed here is aimed to evaluate this casuistic, that was not considered in the literature until now.

The *SCUMBLE* metric aims to quantify the imbalance variance among the labels present in each data sample. This metric (Eq. (4)) is based on the Atkinson index [35] and the *IRLbl* measure (Eq. (1)) proposed in [34]. The former is an econometric measure directed to assess social inequalities among individuals in a population. The latter is the metric that lets us know the imbalance ratio of each label in an MLD. The Atkinson index is used to know the diversity among people's earnings, while our objective is to assess the extend to which labels with different imbalance levels appear jointly. Our first hypothesis is that the higher is the concurrence level the harder would be the work for resampling algorithms, and therefore the worse they would perform.

The Atkinson index is calculated using incomes. We used the imbalance level of each label instead, taking each instance D_i in the MLD D as a population, and the active labels in D_i (those which are relevant to D_i and therefore are set to 1) as the individuals. If the label l is present in the instance i then $IRLbl_{il} = IRLbl(l)$, otherwise $IRLbl_{il} = 0$. $\overline{IRLbl_i}$ stands for the average imbalance level

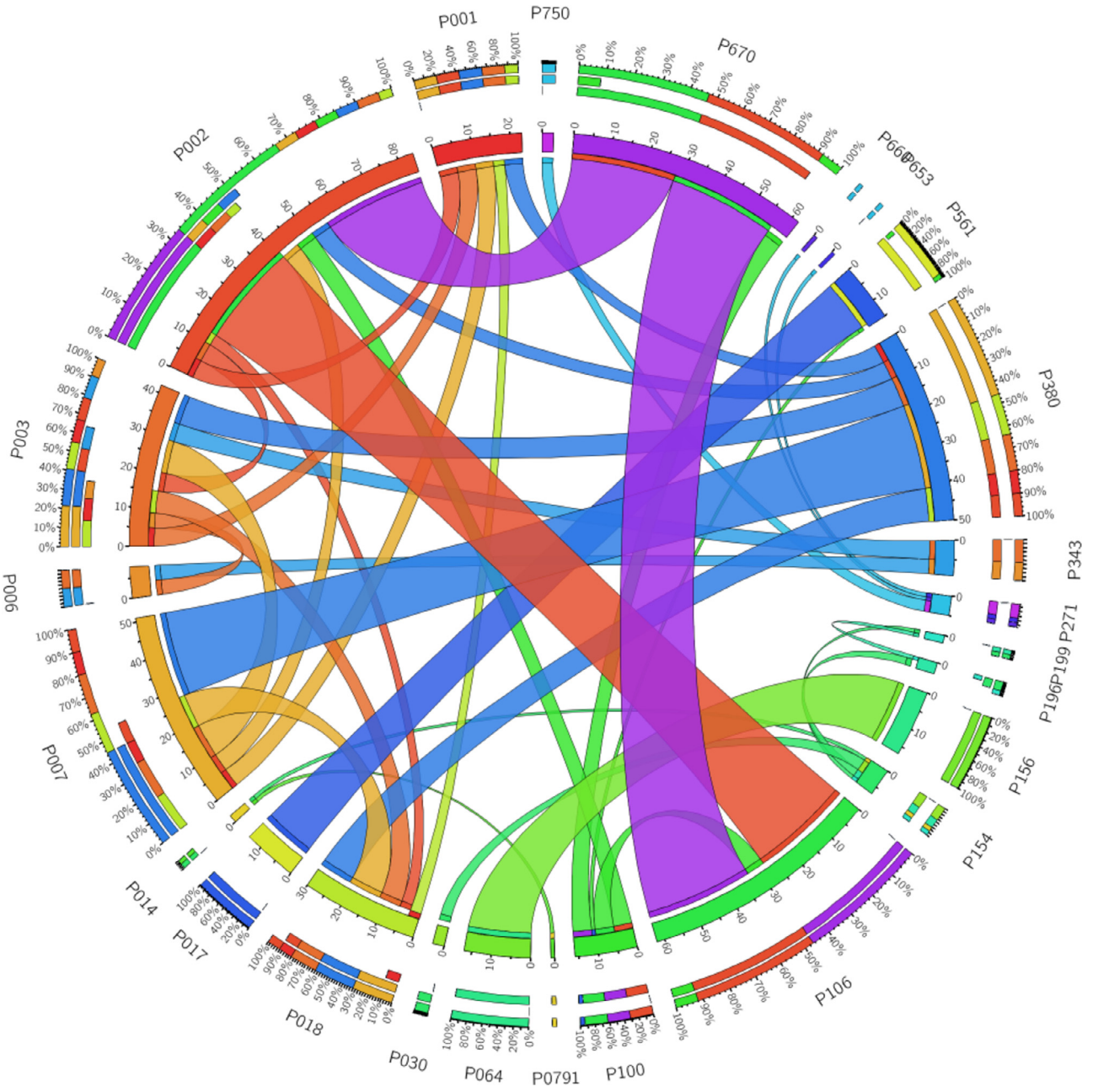


Fig. 1. Label concurrence in genbase MLD.

of the labels appearing in instance *i*. The scores for every sample are averaged, obtaining the final *SCUMBLE* value.

$$SCUMBLE_{ins}(i) = 1 - \frac{1}{IRLbl_i}\left(\prod_{l=1}^{|L|} IRLbl_{il}\right)^{(1/|L|)} \tag{3}$$

$$SCUMBLE(D) = \frac{1}{|D|}\sum_{i=1}^{|D|} SCUMBLE_{ins}(i) \tag{4}$$

Since *SCUMBLE* is computed as an average of concurrence by instance, it could be influenced by extreme values. A few instances with a very high $SCUMBLE_{ins}$ value would introduce a certain deviation into the global *SCUMBLE* measure. To estimate the importance of this deviation, the *SCUMBLE.CV* metric (see Eq. (5)) provides the corresponding coefficient of variation. The higher is the *SCUMBLE.CV*, the larger would be the differences in concurrence among instances.

$$SCUMBLE.CV = \frac{SCUMBLE\sigma}{SCUMBLE},$$

$$SCUMBLE\sigma = \sqrt{\sum_{i=1}^{|D|} \frac{(SCUMBLE_{ins}(i) - SCUMBLE)^2}{|D| - 1}} \tag{5}$$

The *SCUMBLE* measure for an MLD would provide a glimpse at how much concurrence between imbalanced labels there is in it. It also would be interesting to know which labels are more affected by this problem. This is the aim of the *SCUMBLELbl* metric (Eq. (6)). The *SCUMBLELbl.CV* metric can also be obtained, following the same procedure described above for *SCUMBLE.CV*. Since the number of instances in which the assessed label appears is used as denominator, dividing the sum of *SCUMBLE*, that *SCUMBLELbl* will be lower for majority labels is something intuitively deductible. Majority labels usually will interact with minority ones only in a

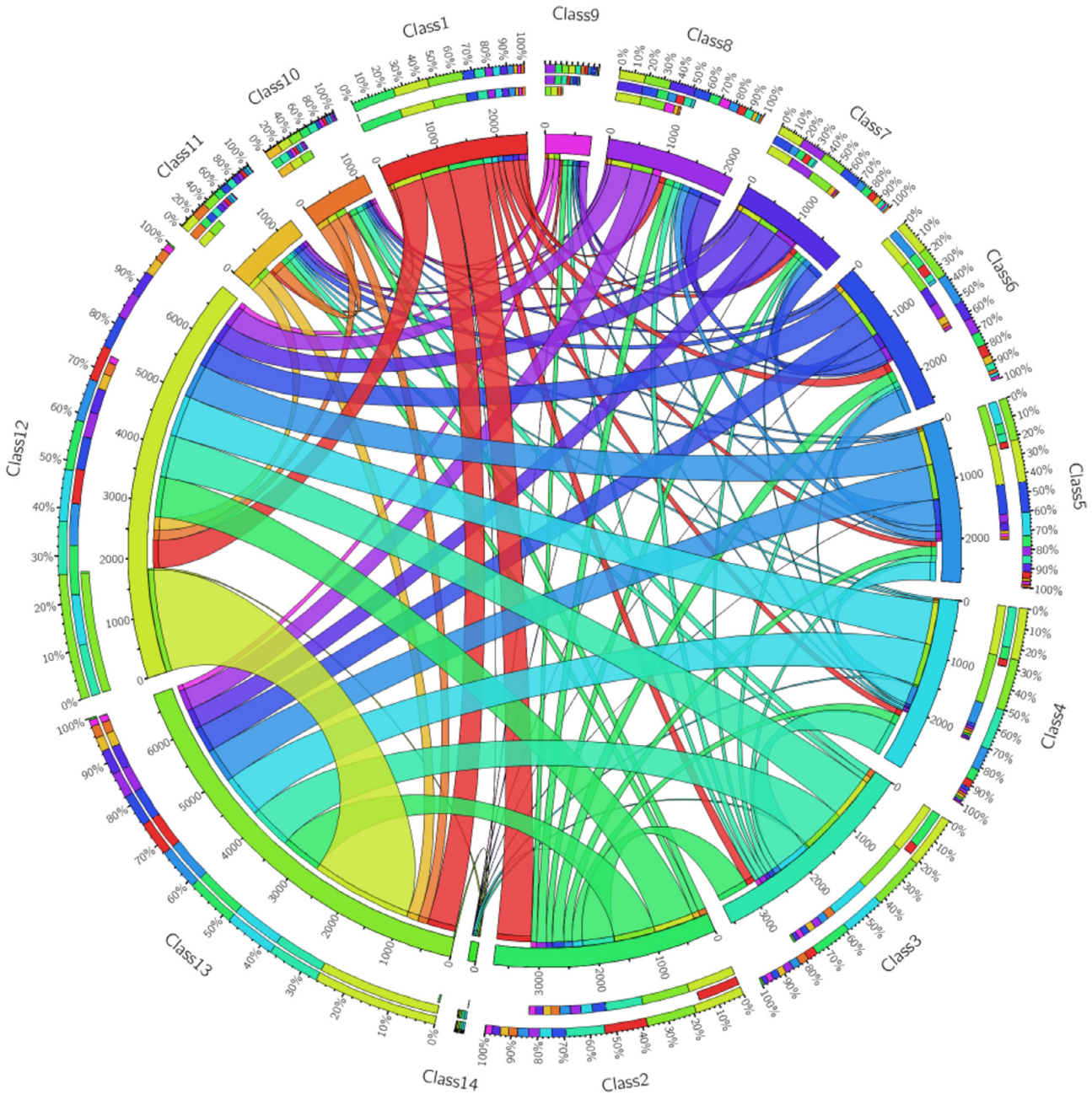


Fig. 2. Label concurrence in yeast MLD.

few instances, those containing the minority label. Therefore, this metric would allow comparisons between labels with a similar frequency in the MLD. Our second hypothesis is that this information would be useful to know which of the minority labels are in fact heavily related to majority ones. In other words, which of them are difficult labels.

$$SCUMBLELbl(y) = \frac{\sum_{i=1}^{|D|} [\![y \in Y_i]\!].SCUMBLE_{ins}(i)}{\sum_{i=1}^{|D|} [\![y \in Y_i]\!]} \quad (6)$$

Whether our initial hypothesis are correct or wrong, and therefore these metrics are able to predict the difficulty that an MLD implies for resampling algorithms or not, is something to be proven experimentally.

4. The algorithm REMEDIAL

In this section the algorithm REMEDIAL, firstly introduced in [36] as a specific method for MLDs with concurrence of highly imbalanced labels, is described. How REMEDIAL has been implemented into the mldr package, and how to use it, is also explained inAppendix A.

As its name suggests, REMEDIAL (*REsampling MultilabEl datasets by Decoupling highly ImbAlanced Labels*) is a method specifically designed for MLDs that suffer from concurrence between imbalanced labels. In this context, *highly imbalanced labels* has to be understood as labels with large differences in their *IRLbls*. This is a fact assessed with the *SCUMBLE* measure, thus REMEDIAL is directed to MLDs with a high *SCUMBLE* level.

When the few samples in which a minority label is present also contain one or more majority labels, whose frequency in the

Table 1
Main characteristics of the datasets.

Dataset	Instances	Attributes	Labels	Labelsets	Card	Dens	MeanIR	MaxIR	SCUMBLE	Ref.
Corel5k	5000	499	374	3175	3.522	0.009	189.568	1 120.000	0.394	[37]
Mediamill	43,907	120	101	6555	4.376	0.043	256.405	1 092.548	0.355	[38]
Cal500	502	68	174	502	26.044	0.150	20.578	88.800	0.337	[19]
Enron	1702	1001	53	753	3.378	0.064	73.953	913.000	0.303	[20]
Corel16k	13,618	500	144	4692	2.815	0.020	32.998	116.407	0.279	[39]
Cs	9270	635	274	4749	2.556	0.009	85.002	226.700	0.272	[3]
Tmc2007	28,596	49,060	22	1341	2.158	0.098	15.157	41.980	0.175	[40]
Yeast	2417	103	14	198	4.237	0.303	7.197	53.412	0.104	[21]
Bibtex	7395	1836	159	2856	2.402	0.015	12.498	20.431	0.094	[41]
Medical	978	1449	45	94	1.245	0.028	89.501	266.000	0.047	[22]
Genbase	662	1186	27	32	1.252	0.046	37.315	171.000	0.029	[4]

MLD is much higher, the power of the input features to predict the labels might be biased to the majority ones. Our hypothesis is that, in a certain way, majority labels are masking the minority ones when they appear together, a problem that could be solved to some extent by decoupling the labels in these instances.

REMEDIAL is kind of a resampling algorithm. It could be seen as an oversampling method, since it produces new instances in some cases. At the same time it also modifies existent samples. However, REMEDIAL never changes the number of samples associated to each label, i.e., the absolute frequency of the labels in the MLD. In short, REMEDIAL is an editing plus oversampling algorithm, and it is an approach which has synergies with traditional resampling techniques. The method pseudo-code is shown in Algorithm 1.

Algorithm 1 REMEDIAL algorithm.

```
1:  function REMEDIAL(MLD D, Labels L)
2:      ▷ Calculate imbalance levels
3:      IRLbl_l ← calculateIRLbl(l in L)
4:      IRMean ← mean(IRLbl)
5:      ▷ Calculate SCUMBLE
6:      SCUMBLEIns_i ← calculateSCUMBLE(D_i in D)
7:      SCUMBLE ← mean(SCUMBLEIns)
8:      for each instance i in D do
9:          if SCUMBLEIns_i > SCUMBLE then
10:             D'_i ← D_i                    ▷ Clone the affected instance
11:             ▷ Maintain minority labels
12:             D_i[labels_{IRLbl<=IRMean}] ← 0
13:             ▷ Maintain majority labels
14:             D'_i[labels_{IRLbl>IRMean}] ← 0
15:             D ← D + D'_i
16:         end if
17:     end for
18: end function
```

The *IRLbl*, *IRMean* and *SCUMBLE* measures are computed in lines 2–7. $SCUMBLE_{Ins_i}$ is the concurrence level of the instance D_i. The mean *SCUMBLE* for the MLD is obtained by averaging the individual *SCUMBLE* for each sample.

Taking the mean *SCUMBLE* as reference, only the samples with a $SCUMBLEIns > SCUMBLE$ are processed. Those instances, which contain minority and majority labels, are decoupled into two instances, one containing only the majority labels and another one with the minority labels. In line 10 D_i, a sample affected by the problem at glance, is cloned in D'_i. The formula in line 12 edits the original D_i instance by removing the majority labels from it. Majority labels are considered as those whose *IRLbl* is equal or below to *IRMean*. Line 14 does the opposite, removing from the cloned D'_i the minority labels. D_i belongs to the D MLD, but D'_i has to be added to it (line 15).

What differentiates REMEDIAL from existing resampling methods, such as the ones enumerated in Section 2.3, is that it does not change the label frequencies in the MLD. All existent proposals increase the number of instances associated to minority labels or decrease the amount of samples linked to majority ones. On the other hand, the goal of REMEDIAL is to look for instances where minority and majority labels appear together, splitting them if is it necessary, but without deleting or adding labels. As far as we are concern, there is not a comparable method to REMEDIAL proposed in the literature.

5. Experimentation and analysis

The conducted experimentation has been structured into two phases. First, the interest is in checking how the *SCUMBLE* level impacts the performance of some resampling methods. Second, how the proposed REMEDIAL algorithm influences the MLDs, and the classification behavior, is analyzed. The test bed framework is described in the next subsection, the obtained results and corresponding analysis of the two aforementioned phases are provided in the following ones.

5.1. Experimental framework

In the first phase of the experimentation, to determine the usefulness of the *SCUMBLE* metric, six of the MLDs shown in Table 1, corel5k, cal500, enron, yeast, medical and genbase, were used. They have been chosen as representatives of different *SCUMBLE* values, including the extreme levels, corel5k (highest) and genbase (lowest), and four values which are in between. The rightmost column indicates each dataset's origin. All of them are imbalanced, so theoretically they could benefit from applying a resampling algorithm. Aside from the *SCUMBLE* measure, the *MaxIR* and *MeanIR* values are also shown. These measurements correspond to whole datasets. The values taken as reference point to the posterior analysis will be average values from training partitions[2] using a 2×5 folds scheme. The datasets appear in Table 1 sorted by *SCUMBLE* value, from higher to lower. According to this measure, corel5k and cal500 would be among the most difficult MLDs in the first group, since they have a high level of concurrence among labels with different imbalance levels. On the other hand, medical and genbase would be the most benefited from resampling , as most of the majority/minority labels in them do not appear together.

Regarding the resampling algorithms, the two proposed in [34] have been applied. Both are based on the LP transformation. LP-ROS does oversampling by cloning instances with minority labelsets, whereas LP-RUS performs undersampling removing samples associated to majority labelsets. All the dataset partitions

[2] The dataset partitions used in this experimentation, as well as color version of all figures, are available to download at http://simidat.ujaen.es/SCUMBLE.

were preprocessed, and the imbalance measures were calculated for each algorithm.

In the second phase of the experimentation, to check the influence of REMEDIAL in classification results, the eleven MLDs shown in Table 1 have been given as input, before and after preprocessing them with REMEDIAL, to six different MLC algorithms:

- BR (*Binary Relevance*) [25]. Ensemble of binary classifiers. It is a transformation based method. A binary classifier is generated for each label and the individual predictions are joined to obtain the final prediction.
- HOMER (*Hierarchy of Multilabel Classifiers*) [42]. Ensemble of multiclass classifiers. It is a transformation method based on the label powerset approach, thus each label combination is interpreted as a label class.
- IBLR (*Instance-Based Learning by Logistic Regression*) [43]. Instance based classifier. IBLR is an improved version of ML-kNN [27], the best-known instance based multilabel classifier.
- CLR (*Calibrated Label Ranking*) [44]. Ensemble of binary classifiers based on pair-wise comparisons. A binary classifier is generated for each label pair, instead of each label as in BR. The classifier produces a ranking of labels, from which the predicted labelset is obtained after applying a threshold.
- ECC (*Ensemble of Classifier Chains*) [45]. Ensemble of binary classifiers based on chaining each model with the next one. The ensemble generates several chains setting the classifiers for each label at random locations in their respective chain.
- EPS (*Ensemble of Pruned Sets*) [46]. Ensemble of multiclass classifiers with pruned labelsets. Each classifier relies on the PS [47] method to prune infrequent labelsets, easing the work of the underlying multiclass classifier.

The C4.5 tree induction algorithm has been used as base classifier where an underlying binary or multiclass classifier is needed. Default parameters were used in all cases.

As stated in [48], the performance of a multilabel classifier should be always assessed by means of several evaluation metrics. In this case, classification results are evaluated using five usual multilabel measures: Hamming Loss (HL), Precision, MacroFMeasure (MacroFM), One Error (OE), and Ranking Loss (RL). HL (see Eq. (7)) is a global sample-based measure. It assesses differences between Z_i, the predicted labelset, and Y_i, the real one, without distinction among labels. The Δ operator returns the symmetric difference between both labelsets. The lower the HL the better the predictions are. Precision (8) is also example-based, and it is among the most usual performance metrics when it comes to evaluate a classifier. MacroFM is the label-based version of the usual F-Measure (see Eqs. (8)–(10)). As can be seen in Eq. (11), in MacroFM F-Measure is evaluated independently for each label and then it is averaged. In the latter equation *TP* stands for *True Positives, FP* for *False Positives, TN* for *True Negatives*, and *FN* for *False Negatives*. OE (12) and RL (13) are ranking-based evaluation metrics. In these equations, $rk(x_i, l)$ is a function that returns the confidence degree for the label l in the prediction Z_i provided by the classifier for the instance x_i. Additional information about all these metrics can be found in [1].

$$HL = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i \Delta Z_i|}{|L|}. \tag{7}$$

$$Precision = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i \cap Z_i|}{|Z_i|} \tag{8}$$

$$Recall = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i \cap Z_i|}{|Y_i|} \tag{9}$$

$$F\text{-}Measure = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{10}$$

$$MacroFM = \frac{1}{|L|}\sum_{i=1}^{|L|} F\text{-}Measure(TP_i, FP_i, TN_i, FN_i) \tag{11}$$

$$OE = \frac{1}{n}\sum_{i=1}^{n}[\![\operatorname{argmax}_{y \in Z_i}\langle rk(x_i, y)\rangle \notin Y_i]\!]. \tag{12}$$

$$RL = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|Y_i|.|\overline{Y_i}|}|y_a, y_b : rk(x_i, y_a) > rk(x_i, y_b)|, \quad (y_a, y_b) \in Y_i \times \overline{Y_i} \tag{13}$$

5.2. SCUMBLE influence in preprocessing and classification algorithms

Once the LP-ROS and LP-RUS resampling algorithms were applied, the imbalance levels on the preprocessed MLDs were reevaluated. Table 2 shows the new *MaxIR* and *MeanIR* values for each dataset. Comparing these values with the original ones, it can be verified that a general improvement in the imbalance levels has been achieved. Although there are some exceptions, in most cases both *MaxIR* and *MeanIR* are lower after applying the resampling algorithms.

It would be interesting to know if the imbalance reduction is proportionally coherent with the values obtained from the *SCUMBLE* measure. The graphs in Figs. 3 and4 are aimed to visually illustrate the connection between *SCUMBLE* values and the relative variations in imbalance levels. For each MLD, their *SCUMBLE* value is represented along with the percentage change in *MaxIR* and *MeanIR* after applying the LP-ROS/LP-RUS resampling methods. The tendency for the three values among the six MLDs is depicted by three logarithmic lines. As can be seen, a clear parallelism exists between the continuous line, which corresponds to *SCUMBLE*, and the dashed lines. This affinity is specially remarkable with the LP-RUS algorithm (Fig. 4).

Although the previous figures allow to infer that an important correlation between the *SCUMBLE* measure and the success of the resampling algorithms exists, this relationship must be formally

Table 2
Imbalance levels after applying resampling algorithms (average values on training partitions).

	LP-ROS				LP-RUS			
	MaxIR		MeanIR		MaxIR		MeanIR	
Dataset	Before	After	Before	After	Before	After	Before	After
Corel5k	896.000	969.400	166.057	140.743	896.000	817.100	166.057	155.032
Cal500	133.192	179.358	21.274	25.468	133.192	133.192	21.274	21.274
Enron	657.050	710.967	72.552	53.255	657.050	620.050	72.552	68.672
Yeast	53.689	15.418	7.218	2.612	53.689	83.800	7.218	19.884
Medical	212.800	39.963	68.388	10.556	212.800	46.570	68.388	6.371
Genbase	136.800	13.703	31.665	4.500	136.800	150.800	31.665	51.157

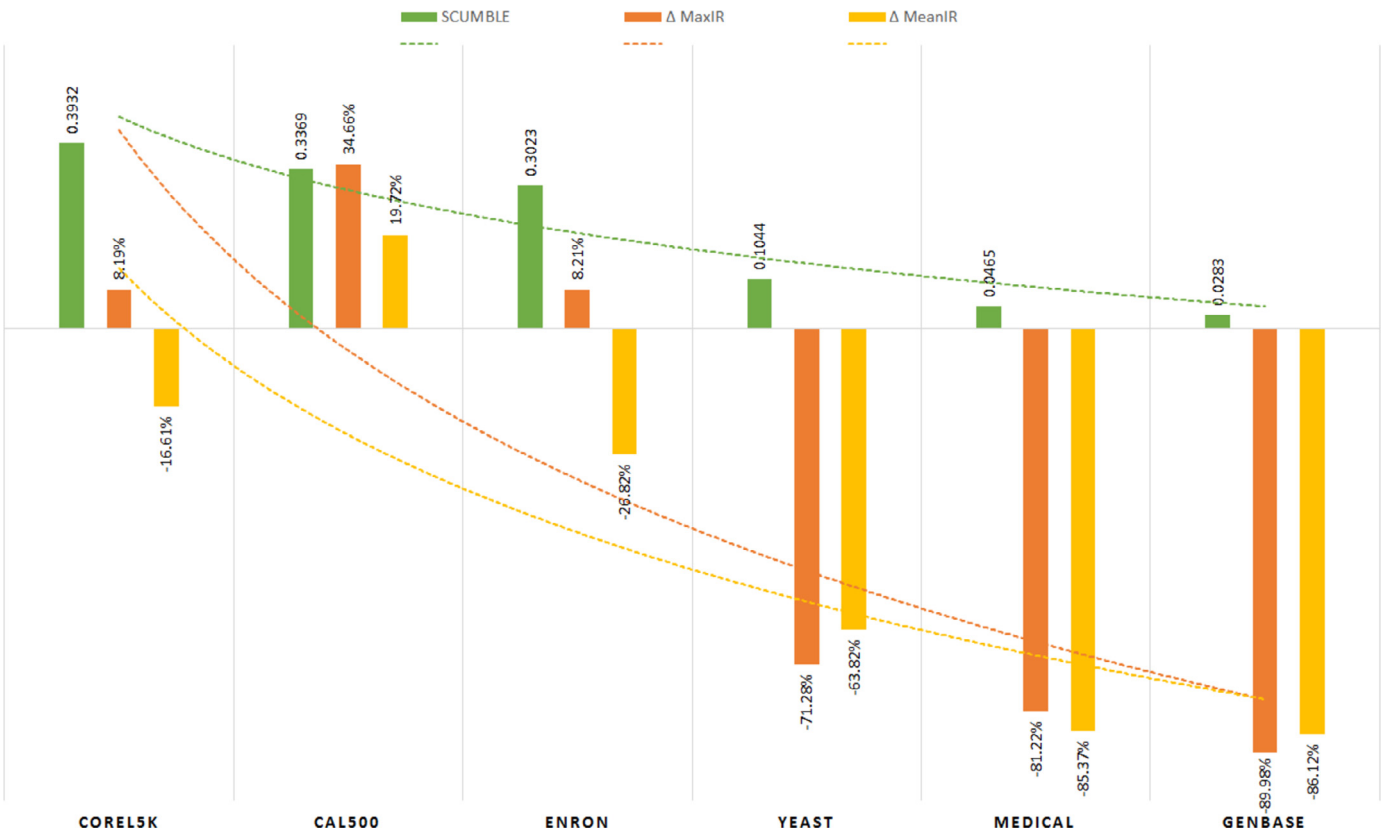


Fig. 3. SCUMBLE vs. changes in imbalance level after applying LP-ROS.

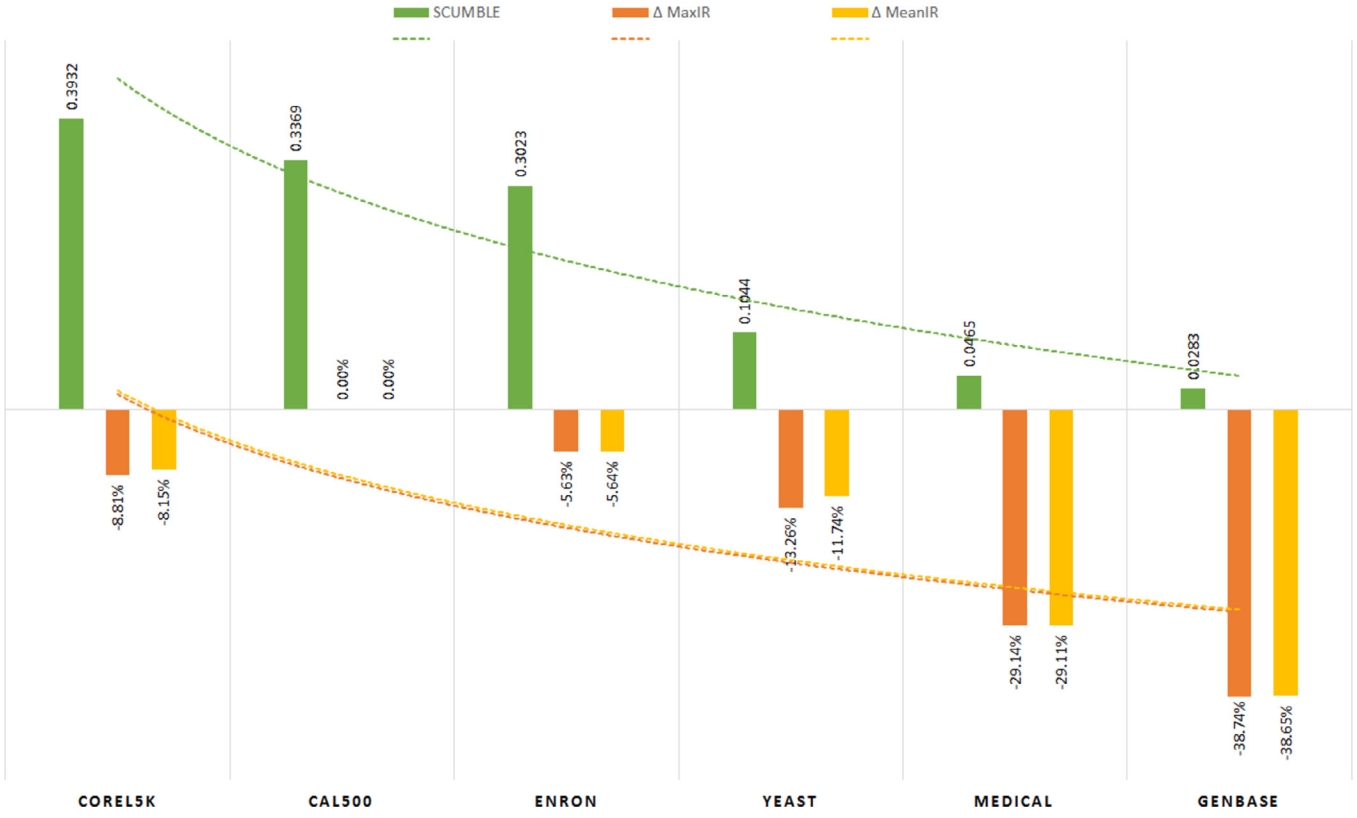


Fig. 4. SCUMBLE vs. changes in imbalance level after applying LP-RUS.

Table 3
Results from the Pearson correlation tests.

	SCUMBLE vs. ΔMaxIR		SCUMBLE vs. ΔMeanIR	
Algorithm	Cor	*p*-value	Cor	*p*-value
LP-ROS	0.8120	0.0497	0.9189	0.0096
LP-RUS	0.8607	0.0278	0.8517	0.0314

Table 4
F-measure values obtained by HOMER MLC algorithm (average values over test partitions).

Dataset	Base	LP-RUS	LP-ROS	ΔRUS	ΔROS
Corel5k	0.3857	0.2828	0.2920	−26.6788	−24.2935
Cal500	0.3944	0.3127	0.3134	−20.7150	−20.5375
Enron	0.5992	0.5761	0.5874	−3.8551	−1.9693
Yeast	0.6071	0.6950	0.6966	14.4787	14.7422
Medical	0.9238	0.9158	0.9162	−0.8660	−0.8227
Genbase	0.9896	0.9818	0.9912	−0.7882	0.1617

analyzed. To this end, a Pearson correlation test was applied over the *SCUMBLE* values and the relative changes in imbalance levels for each resampling algorithm. The resulting correlation coefficients and *p-values* are shown in Table 3. It can be seen that all the coefficients are above 80%, and all the *p-values* are under 0.05. Therefore, a statistical correlation between the *SCUMBLE* measure and the behavior of the tested resampling algorithms can be concluded.

Following this analysis, it seems reasonable to avoid resampling algorithms when the *SCUMBLE* measure for an MLD is well above 0.1, such as is the case with corel5k, cal500 and enron. In this situation the benefits obtained from resampling, if any, are very small. The result can even be a worsening of the imbalance level. In average, the *MeanIR* for the three MLDs with *SCUMBLE* > 0.3 has been reduced only by 6%, while the *MaxIR* is actually increasing in the same percentage. By contrast, the average *MeanIR* reduction for the other three MLDs, with *SCUMBLE* $\lesssim 0.1$, reaches 52% and the *MaxIR* reduction 54%.

Aiming to know how these changes in the imbalance levels would influence classification results, and if a correlation with *SCUMBLE* values exists, the HOMER [42] algorithm was used. It must be highlighted that the interest here is not in the raw performance values, but in how they change after a resampling algorithm has been applied and how this change correlates with *SCUMBLE* values. Therefore, the HOMER algorithm is used only as a tool to obtain classification results before and after applying the resampling. Any other MLC algorithm could be used for this task. Additionally, the proposed *SCUMBLE* measure is not used in the experimentation to influence the behavior of LP-ROS, LP-RUS or HOMER by any means. The goal is to explore the correlation between changes in classification results and *SCUMBLE* values.

Table 4 shows these results assessed with F-measure, the harmonic mean of precision and recall measures. It can be seen that with the three MLDs which show high *SCUMBLE* values, the preprocessing has produced a remarkable deterioration in classification results. Among the other three MLDs the resampling has improved them in some cases, while producing a slight worsening (less than 1%) in others. Therefore, even though the MLC algorithm behavior would be also affected by other dataset characteristics, that the *SCUMBLE* metric would offer valuable information to determine the convenience of applying a resampling method can be concluded.

5.3. REMEDIAL experimental results

Once the usefulness of the *SCUMBLE* metric has been demonstrated, the next experimental phase has been applying the algorithm REMEDIAL to the eleven datasets previously shown in Table 1, then learning from them using six multilabel classifiers. The results obtained from each one of them over the datasets, before and after preprocessing, are provided in Tables 5–9. Each table correspond to one evaluation metric. Best results are highlighted in bold. EPS was not able to process a couple of datasets.

The analysis of these results can be structured into three parts depending on where we put the focus, the classifiers, the datasets or the evaluation metrics.

- Going through the results by classifier, that REMEDIAL works better with BR and HOMER than with IBLR and CLR can be easily observed. The results for ECC and EPS are not conclusive, with almost as many cases with improvements and worsenings. Binary relevance based algorithms train a classifier for each label, taking as positive the instances containing it and as negative the remainder samples. When a majority label is being processed, all the instances in which it appears jointly with a minority label are processed as positive, disregarding the fact that they contain other labels. The decoupling of these labels tends to balance the bias of each classifier, something that also influences the behavior of ECC. LP based algorithms, such as HOMER, surely are favored by REMEDIAL, since the decoupling produces simpler labelsets. Moreover, the number of distinct labelsets is reduced after the resampling. The influence of REMEDIAL on instance based classifiers, such as IBLR, is easy to devise. The attributes of the decoupled samples do not change, so they will occupy exactly the same position with respect to the instance which is taken as reference for searching nearest neighbors. Therefore, the classifier will get two samples at the same distance but with disjoint labelsets, something that can

Table 5
Results before and after applying REMEDIAL assessed with Hamming Loss (↓).

	BR		CLR		ECC		EPS		HOMER		IBLR	
Dataset	Before	After	Before	After	Before	After	Before	After	Before	After	Before	After
Bibtex	0.0147	**0.0132**	**0.0127**	0.0130	**0.0126**	0.0135	–	–	0.0185	**0.0166**	0.0165	**0.0155**
Cal500	0.1630	**0.1497**	**0.1381**	0.1442	**0.1422**	0.1490	–	–	0.1875	**0.1815**	0.2341	**0.2125**
Corel16k	0.0206	**0.0196**	0.0198	**0.0195**	0.0387	**0.0195**	**0.0196**	0.0268	0.0271	**0.0228**	0.0199	**0.0198**
Corel5k	0.0098	**0.0094**	0.0095	**0.0094**	0.0094	**0.0089**	0.0173	**0.0101**	0.0132	**0.0118**	0.0242	**0.0148**
Cs	0.0094	**0.0089**	0.0088	0.0088	**0.0086**	0.0578	0.0133	**0.0112**	0.0117	**0.0104**	0.0182	**0.0143**
Enron	**0.0522**	0.0540	**0.0476**	0.0517	0.0484	**0.0086**	0.0733	**0.0601**	0.0574	**0.0555**	**0.0571**	0.0593
Genbase	**0.0012**	0.0084	**0.0014**	0.0080	**0.0014**	0.0386	**0.0028**	0.0040	**0.0016**	0.0064	**0.0022**	0.0092
Mediamill	0.0343	**0.0331**	**0.0291**	0.0321	0.0288	**0.0118**	0.0524	**0.0377**	0.0384	**0.0355**	**0.0291**	0.0338
Medical	**0.0107**	0.0131	**0.0109**	0.0132	**0.0100**	0.0717	**0.0141**	0.0143	**0.0109**	0.0118	0.0198	0.0198
Tmc2007	**0.0568**	0.0684	**0.0538**	0.0658	**0.0507**	0.2316	0.0872	**0.0693**	**0.0607**	0.0647	**0.0646**	0.0775
Yeast	0.2505	**0.2347**	**0.2202**	0.2228	0.3594	**0.0094**	**0.2042**	0.2853	0.2601	**0.2476**	**0.1941**	0.2264

Table 6
Results before and after applying REMEDIAL assessed with Precision (↑).

	BR		CLR		ECC		EPS		HOMER		IBLR	
Dataset	Before	After	Before	After	Before	After	Before	After	Before	After	Before	After
Bibtex	0.5770	**0.7451**	0.8267	**0.9173**	0.7385	**0.8023**	–	–	0.4701	**0.5356**	0.4920	**0.5004**
Cal500	0.4397	**0.5326**	0.6363	**0.8327**	0.5636	**0.6848**	–	–	0.3842	**0.3983**	**0.2859**	0.2743
Corel16k	0.3610	**0.4682**	0.4455	**0.6056**	0.1944	**0.6913**	**0.4697**	0.2010	0.2475	**0.2921**	**0.3623**	0.3079
Corel5k	0.3643	**0.4781**	0.4621	**0.5983**	0.5465	**0.6868**	0.1938	**0.2906**	0.2232	**0.2438**	0.0598	**0.0602**
Cs	0.5174	**0.6239**	0.6297	**0.7247**	0.6211	**0.7611**	0.3366	**0.3632**	0.3884	**0.4394**	**0.1076**	0.0919
Enron	0.6391	**0.7063**	0.7047	**0.7813**	0.6681	**0.9960**	0.4873	**0.5592**	0.5893	**0.6269**	0.6151	**0.6519**
Genbase	0.9947	**0.9977**	0.9946	**0.9977**	**0.9950**	0.8877	**0.9950**	0.9942	0.9932	**0.9961**	**0.9899**	0.9890
Mediamill	0.6683	**0.8091**	0.7959	**0.8707**	0.7986	**0.8740**	0.4827	**0.6271**	0.6177	**0.6805**	0.7758	**0.8388**
Medical	0.8633	**0.8680**	0.8699	**0.8725**	0.8636	**0.8701**	0.7813	**0.7837**	0.8639	**0.8648**	0.7272	**0.7552**
Tmc2007	0.7675	**0.8334**	0.7855	**0.8539**	**0.8056**	0.7370	0.6079	**0.6903**	0.7389	**0.7829**	0.7309	**0.8031**
Yeast	0.6020	**0.6647**	0.6768	**0.7323**	0.4777	**0.6899**	**0.6960**	0.5422	0.5876	**0.6169**	0.7110	**0.7442**

Table 7
Results before and after applying REMEDIAL assessed with Macro F-Measure (↑).

	BR		CLR		ECC		EPS		HOMER		IBLR	
Dataset	Before	After	Before	After	Before	After	Before	After	Before	After	Before	After
Bibtex	0.3368	**0.3604**	0.3342	**0.3518**	0.3750	**0.3763**	–	–	0.2984	**0.2985**	**0.2140**	0.1950
Cal500	**0.2933**	0.2286	**0.3323**	0.2436	**0.3058**	0.0670	–	–	0.3301	**0.3372**	**0.2772**	0.2527
Corel16k	**0.1550**	0.1266	**0.1084**	0.0707	**0.1477**	0.1056	0.1223	**0.1401**	**0.1510**	0.1377	**0.1146**	0.0956
Corel5k	0.1774	**0.1827**	**0.1330**	0.1073	0.1666	**0.2922**	**0.1860**	0.1767	**0.1963**	0.1860	0.1060	**0.1432**
Cs	0.3457	**0.3795**	**0.2801**	0.2606	**0.3617**	0.2760	**0.3044**	0.2992	**0.2999**	0.2922	**0.1355**	0.1341
Enron	0.4029	**0.4189**	**0.4199**	0.3755	0.4324	**0.8970**	0.3828	**0.3933**	**0.3836**	0.3828	**0.3458**	0.2755
Genbase	0.9890	**0.9923**	**0.9848**	0.9415	**0.9906**	0.1741	**0.9775**	0.9527	**0.9806**	0.9662	**0.9655**	0.8449
Mediamill	0.2836	**0.2959**	**0.2307**	0.2011	0.2445	**0.8085**	**0.3382**	0.2657	**0.2492**	0.2147	**0.2818**	0.1820
Medical	**0.8165**	0.8013	**0.7942**	0.7864	**0.8179**	0.3318	0.7283	**0.7292**	**0.7981**	0.7855	**0.6404**	0.6189
Tmc2007	**0.6015**	0.4243	**0.6073**	0.3578	**0.5966**	0.4063	0.5802	**0.5951**	**0.5981**	0.4551	**0.4667**	0.2786
Yeast	0.4341	**0.5204**	**0.4480**	0.4075	**0.4782**	0.1356	**0.4629**	0.4428	0.4363	**0.4400**	**0.4945**	0.3901

Table 8
Results before and after applying REMEDIAL assessed with One Error (↓).

	BR		CLR		ECC		EPS		HOMER		IBLR	
Dataset	Before	After	Before	After	Before	After	Before	After	Before	After	Before	After
Bibtex	0.5060	**0.4648**	**0.4110**	0.4120	**0.3886**	0.4018	–	–	0.6040	0.6040	**0.6043**	0.6651
Cal500	0.7202	**0.6963**	0.1254	**0.1245**	**0.1504**	0.3735	–	–	0.8167	**0.7660**	**0.8756**	0.8885
Corel16k	**0.6964**	0.7289	**0.6650**	0.6696	0.7138	**0.7036**	**0.6809**	0.7906	**0.7712**	0.8065	**0.7106**	0.7359
Corel5k	**0.7067**	0.7134	**0.6716**	0.6753	0.6828	**0.5181**	**0.7853**	0.9070	**0.7994**	0.8165	0.9401	**0.9066**
Cs	0.5690	**0.5679**	**0.5112**	0.5130	0.4833	**0.3144**	**0.5596**	0.6629	**0.6680**	0.7042	0.9041	**0.8711**
Enron	0.3922	**0.3554**	0.2350	**0.2311**	0.2700	**0.0037**	**0.3044**	0.4168	0.4456	**0.4360**	**0.3805**	0.3734
Genbase	**0.0052**	0.0060	**0.0022**	0.0030	**0.0022**	0.1769	**0.0037**	0.0068	0.0114	**0.0106**	**0.0098**	0.0384
Mediamill	0.3943	**0.2093**	**0.1125**	0.1155	**0.1153**	0.1616	**0.1155**	0.1651	0.3839	**0.3479**	**0.1215**	0.1351
Medical	**0.1906**	0.1984	0.1559	**0.1544**	**0.1534**	0.1716	**0.1830**	0.1917	**0.2107**	0.2224	0.3190	**0.3175**
Tmc2007	0.2374	**0.2137**	0.1575	**0.1544**	**0.1603**	0.2685	**0.1855**	0.2122	**0.2788**	0.2836	**0.2298**	0.2345
Yeast	0.4181	**0.3856**	0.2399	**0.2366**	**0.2574**	0.7249	**0.2520**	0.2873	0.4268	**0.3885**	**0.2255**	0.2441

be confusing depending on how the algorithm predicts the labelset of the reference sample.

- Analyzing the results by dataset, two thirds of the best values for enron, corel5k and cal500 are obtained after applying REMEDIAL. As can be checked, these are the datasets with highest *SCUMBLE* levels. On the other hand, the results that correspond to the genbase, medical and tmc2007 have not improvements. As shown in Table 1, these are three datasets with low *SCUMBLE* values. Although some differences are quite small, in general the decoupling of labels has worsened classification performance. The remainder five MLDs get mixed results, although this trend (the higher the *SCUMBLE* level the more the result is improved) is similar. As a consequence a clear guideline follows from the analysis of these results, REMEDIAL should not be used with MLDs with low *SCUMBLE* levels, since it is an algorithm specifically designed to face the opposite casuistic.
- Lastly, focusing on the evaluation metrics, that Precision is higher after applying REMEDIAL for most of the datasets and classifiers, with only 9 out of 64 (14%) cases without improvements, can be observed. According to the other four evaluation metrics, HL, MacroFM, OE and RL, there is almost a tie between cases whose results have been improved and those which have not achieved this goal. The view changes drastically depending on each classifier/metric combination. For instance, the HL and Precision values for HOMER state that REMEDIAL improves results in 19 out of 22 cases (86%), but MacroFM and RL indicates the same only for 5 out of 22 (23%).

The statistical significance of the differences in the results just pointed out has been assessed by means of a paired Wilcoxon statistical test. The exact *p*-values for each metric/classifier are the shown in Table 10. Those preceded with a * symbol can be considered as significant from a statistical point of view, applying the usual 0.05 threshold. Most of the differences are not statistically significant. However, Precision and MacroFM show important differences in half or more of the cases. The former metric reveals statistically significant improvements with BR, CLR, ECC and HOMER. On the contrary, MacroFM indicates that the worsening of results is remarkable for CLR, HOMER and IBLR.

Table 9
Results before and after applying REMEDIAL assessed with Ranking Loss (↓).

	BR		CLR		ECC		EPS		HOMER		IBLR	
Dataset	Before	After	Before	After	Before	After	Before	After	Before	After	Before	After
Bibtex	**0.1635**	0.1986	**0.0620**	0.0622	**0.0950**	0.1206	–	–	**0.3295**	0.3382	**0.1742**	0.1824
Cal500	0.3159	**0.2086**	0.1809	**0.1805**	**0.1975**	0.2343	–	–	0.3911	**0.3775**	0.3790	**0.3226**
Corel16k	0.1857	**0.1848**	0.1335	0.1335	0.2907	**0.1840**	**0.1772**	0.3367	**0.3973**	0.4228	**0.1687**	0.1733
Corel5k	0.1474	**0.1416**	0.1176	0.1176	0.1414	**0.1295**	**0.4807**	0.6748	**0.4387**	0.4690	0.2754	**0.2620**
Cs	0.1996	**0.1703**	0.0672	0.0672	0.1118	**0.0911**	**0.2604**	0.3409	**0.3532**	0.3770	**0.2234**	0.2271
Enron	0.1746	**0.1409**	0.0737	**0.0736**	0.0852	**0.0062**	**0.1667**	0.2333	**0.2502**	0.2759	0.1066	0.1066
Genbase	**0.0030**	0.0137	**0.0088**	0.0089	**0.0036**	0.0534	**0.0078**	0.0096	**0.0060**	0.0240	**0.0040**	0.0326
Mediamill	0.1742	**0.0761**	**0.0336**	0.0338	0.0439	**0.0394**	**0.0738**	0.1154	**0.2162**	0.2236	**0.0391**	0.0404
Medical	**0.0703**	0.0785	0.0297	0.0297	**0.0357**	0.0547	0.0686	**0.0664**	**0.0999**	0.1045	0.0653	**0.0640**
Tmc2007	**0.1139**	0.1271	0.0347	**0.0338**	**0.0444**	0.1895	**0.0631**	0.0956	**0.1547**	0.1852	**0.0558**	0.0589
Yeast	0.3156	**0.2536**	0.1799	**0.1785**	**0.2021**	0.1442	**0.1854**	0.2241	0.3407	**0.3213**	**0.1643**	0.1768

Table 10
Exact *p*-values produced by the Wilcoxon statistical test for each classifier/metric.

Classifier	Hamming Loss	Precision	Macro F-Measure	One Error	Ranking Loss
BR	0.824098	*0.003857	0.893904	0.142368	0.266402
CLR	*0.016316	*0.003857	*0.006692	0.893904	0.824098
ECC	0.689084	*0.029383	0.398305	0.449804	0.964541
EPS	0.150786	0.352542	0.932647	0.150786	0.150786
HOMER	0.168167	*0.003857	*0.029383	0.893904	*0.045447
IBLR	0.683481	0.266402	*0.018408	0.398305	0.414823

Overall, REMEDIAL would be a recommended resampling for MLDs with high *SCUMBLE* levels and when BR or LP based classifiers are going to be used. In these cases the prediction of minority labels would be improved, and the global performance of the classifiers would be better. MLDs such as genbase, medical and tmc2007, as their intrinsic traits have demonstrated, should not be processed with REMEDIAL. The same would be applicable to classifiers such as IBLR, as putting two data samples at the same location but having disjoint labelsets tend to confuse this kind of algorithms. Excluding these cases, the global evaluation of the results produced by REMEDIAL would be much more positive.

The already described before are the benefits brought by REMEDIAL on their own, but this algorithm could be used as a first step aimed to ease the work of traditional resampling techniques. Mixing REMEDIAL with standard oversampling and undersampling techniques would be an interesting further study.

6. Conclusions

From the conducted experimentation and further analysis it can be inferred that, while working with imbalanced MLDs, standard resampling methods should be avoided when the *SCUMBLE* level is well above 0.1. In this situation the benefits from resampling are almost negligible, or even detrimental.

In the described scenario, with MLDs suffering from high concurrence among imbalance labels, the proposed REMEDIAL algorithm has proven to be effective. The algorithm looks for instances with a high *SCUMBLE* level and decouples minority and majority labels, producing new instances. The conducted experimentation has proven that REMEDIAL is able to improve classification results when applied to MLDs with a high *SCUMBLE*.

How to assess the concurrence problem, and how to deal with it in practice, has been explained by means of the mldr R package (seeAppendix A). This software has been extended by the authors to include the metrics and algorithms described in this paper. The goal is to help anyone interested in this topic to conduct their personal analysis.

It could be concluded that basic resampling algorithms, which clone the labelsets in new instances or remove samples, are not a general solution in the multilabel field. More sophisticated approaches, which take into account the concurrence among imbalanced labels, would be needed. A potential way for designing these new algorithms would be joining REMEDIAL with some of the existent resampling methods. Once the labels have been decoupled, traditional oversampling and undersampling algorithms would find less obstacles to do their work.

Acknowledgments

This work is partially supported by the Spanish Ministry of Economy, Industry and Competitiveness under projects TIN2014-57251-P and TIN2015-68454-R, and the Andalusian regional project P11-TIC-7765.

Appendix A. SCUMBLE and REMEDIAL implementations in the mldr package

This appendix describes how to obtain the *SCUMBLE* measurement for any MLD, as well as how to apply the REMEDIAL algorithm proposed in this study to any MLD, by means of a software package developed by the same authors.

A1. Assessing label concurrence with the mldr package

The mldr package [18] provides an easy way to make exploratory analysis over MLDs from R, one of the best known tools for machine learning tasks. A quick tutorial describing how to install and use this package can be found in [49]. The capabilities of the mldr package have been extended to include functions aimed to ease the concurrence analysis in MLDs. These new capabilities, developed ad hoc for the present work, are described below.

Once the package has been loaded into R, the first step will be reading the MLD to analyze. MULAN [50] and MEKA [51] file formats are supported. In order to load an MLD, the `mldr` function has to be called providing the file name. The returned result is an S3 R object containing the data (instances with attribute values) and also a plethora of characterization metrics.

The group of metrics that are general to the whole MLD can be retrieved with the `summary` function, as shown in the upper part of Fig. A.5. In this example the measures belonging to the genbase

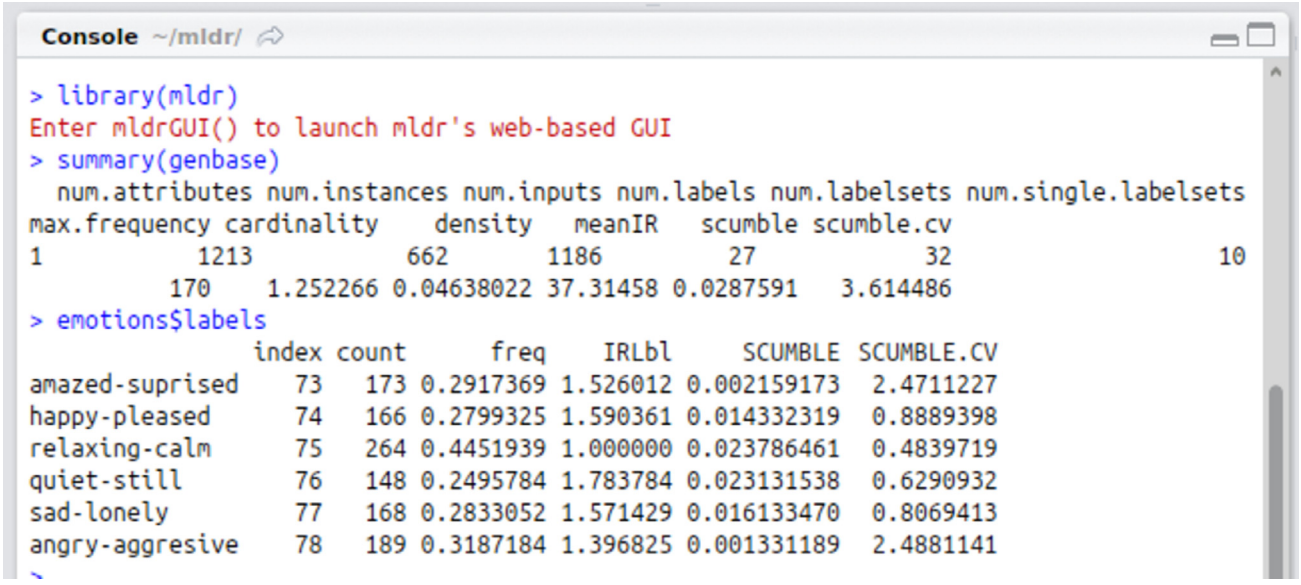


Fig. A5. Obtaining basic concurrence metrics using the mldr R package.

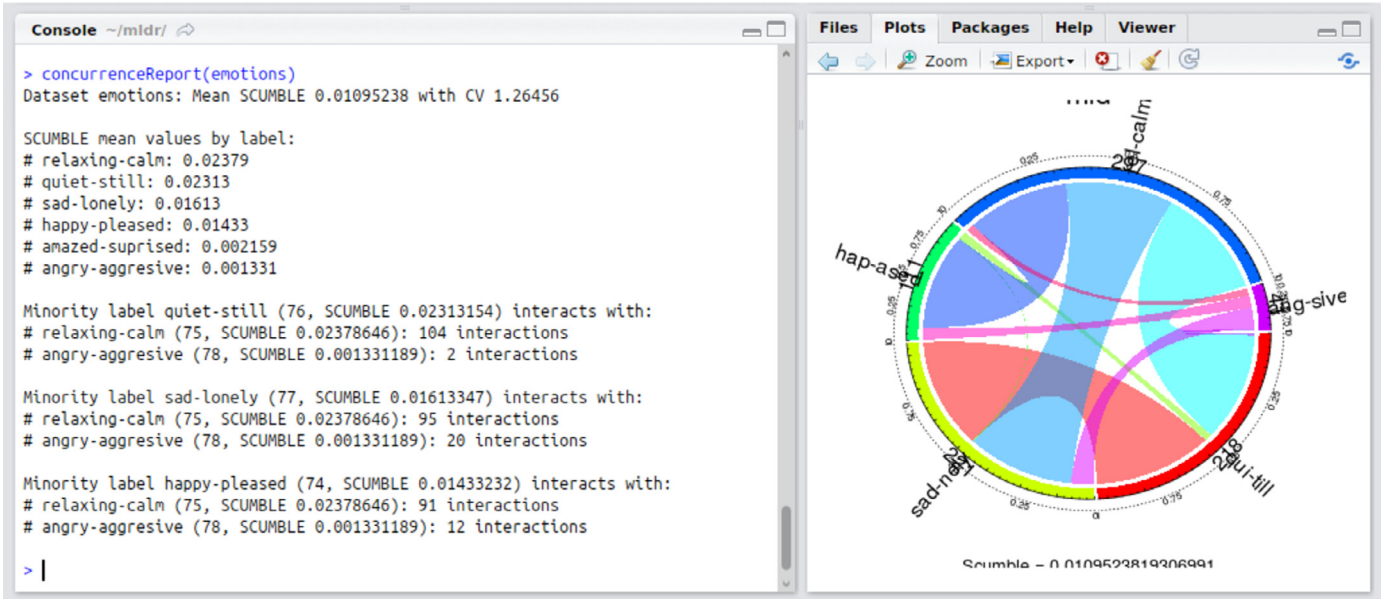


Fig. A6. The concurrence report provides information about label interactions, both textually and visually.

MLD have been obtained. By querying the `labels` member of the object the information relative to each label is retrieved, including the *SCUMBLELbl* and its corresponding coefficient of variation as shown in the bottom part of the same Fig. A.5.

Relying on the measures obtained with the previous methods, essentially the *IRLbl*, *SCUMBLELbl* and name of each label, it is possible to infer which are the minority labels and which of those are more affected by the concurrence problem. However, it would not be easy to know what majority labels are interacting with each minority one. This information can be visually explored using the specific `plot` function provided by the mldr package, able to generate interaction plots similar to the ones shown in Figs. 1 and2.

Another alternative would be calling the mldr's function `concurrenceReport`. It generates a full report stating what are the *SCUMBLE* levels for the MLD and each of its labels, as well as a summary of label interactions and a plot of them. This report is sent to the console by default (see Fig. A.6), but it can also be saved as a PDF document by providing the `pdfOutput` parameter with the `TRUE` value. The report will include the minority labels most affected by the concurrence problem, sorted by their *SCUMBLELbl* value. It will be, in fact, a list of difficult labels, along with the majority labels each one of them interacts with.

In addition to the aforementioned functions, which are only a small sample of the set provided by the mldr package, a web GUI is also available. This can be launched from the R command line with the `mldrGUI`[3] function. It is structured into several pages, accessible by the tags located at the top. In the Concurrence page the same information provided by the `concurrenceReport` can be found, along with a customizable plot showing label interac-

[3] An online version of the mldr's web interface, accessible from any browser without needing to install R or the mldr package, is available at https://fdavidcl.shinyapps.io/mldr. Although the bandwidth provided by shinyapps.io is limited, the application can be used to test the functionality described in this section.

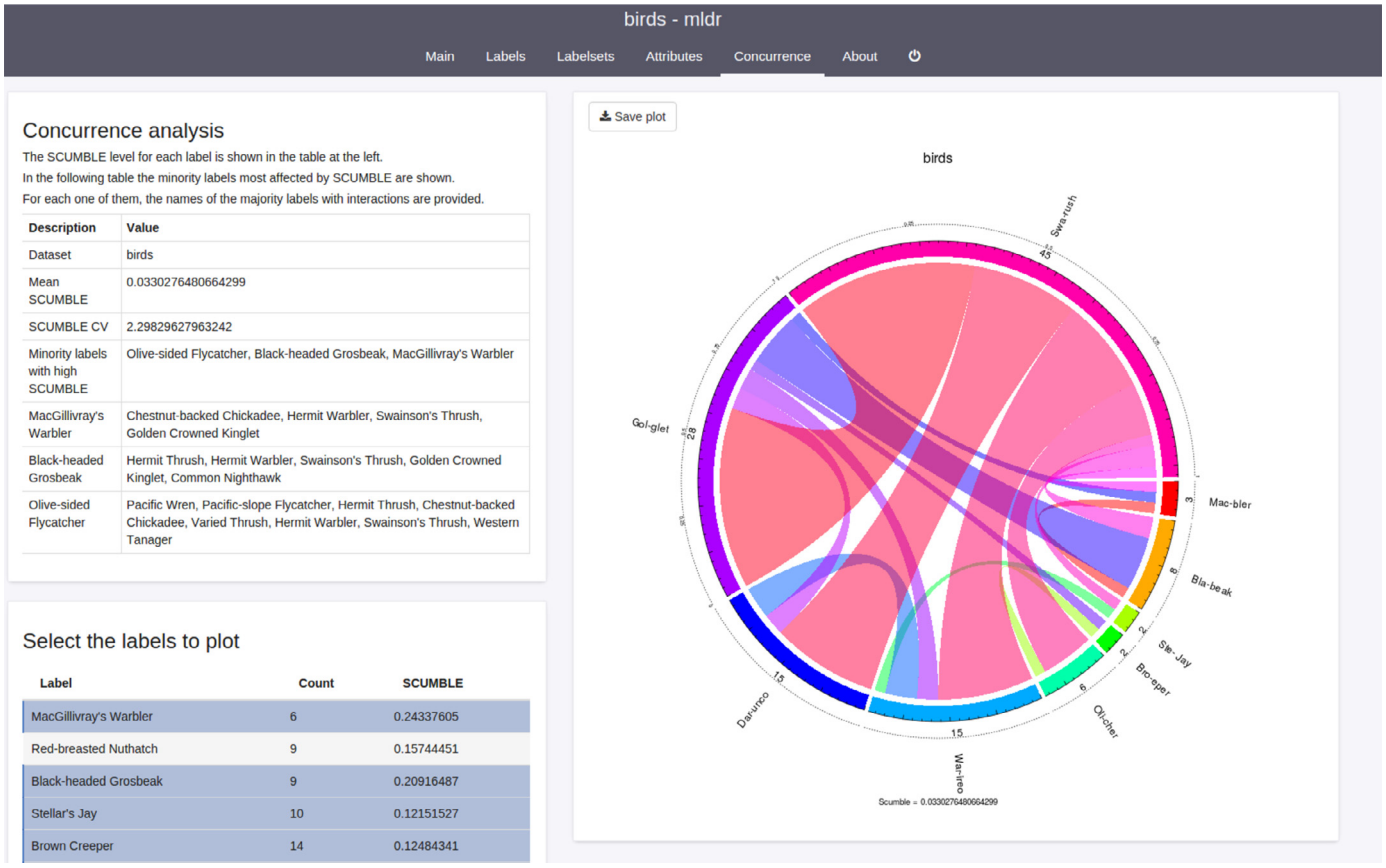


Fig. A7. The mldr GUI eases the process of obtaining customized concurrence information.

```
Console ~/mldr/
> library(mldr)
Enter mldrGUI() to launch mldr's web-based GUI
>
> genbase.decoupled <- remedial(genbase)
> summary(genbase)
  num.attributes num.instances num.inputs num.labels num.labelsets num.single.labelsets max.frequency cardinality    density   meanIR   scumble scumble.cv
1           1213          662       1186         27            32                  10           170    1.252266 0.04638022 37.31458 0.0287591   3.614486
> summary(genbase.decoupled)
  num.attributes num.instances num.inputs num.labels num.labelsets num.single.labelsets max.frequency cardinality    density   meanIR    scumble scumble.cv
1           1213          752       1186         27            32                   7           171    1.102394 0.04082939 37.31458 0.01395751   3.483609
```

Fig. A8. MLD basic traits before and after applying the REMEDIAL preprocessing algorithm.

tions. This page is partially visible in Fig. A.7. The list below the report allows the interactive selection of labels to be shown in the plot. The result can be saved to a file.

Overall, the exploratory tools implemented into the mldr package will provide all the information needed to analyze how the concurrence among imbalanced labels affects a certain MLD, as well as which of the labels could be considered difficult labels.

A2. The mldr package's REMEDIAL implementation

Along with the exploratory functionality previously described, the mldr package has been also extended by including a reference implementation of the algorithm REMEDIAL. The function containing this implementation is called `remedial`. To use it an mldr object has to be given as input, obtaining as output the preprocessed version of the same object. In Fig. A.8 how to use this function is shown. The algorithm is applied to the genbase MLD, storing the result into the `genbase.decoupled` variable.

From the information provided by the `summary` function, corresponding to the MLD before and after the preprocessing, the following facts can be observed:

- The number of instances grows, as REMEDIAL produces new data samples.
- Since the number of active labels in the MLD does not change, neither do the number of labels and the imbalance related metrics, such as *MeanIR*.
- Because the same number of active labels are split into a larger number of instances, label cardinality and density decrease.
- In general, the decoupling of labels tend to produce simpler and more frequent labelsets.
- The global *SCUMBLE* and the *SCUMBLELbl* are reduced.

As the algorithm REMEDIAL takes as reference the mean *SCUMBLE* to determine which samples are going to be decoupled, and this measure is reduced as a result of applying REMEDIAL, it can be run several times over the same data to progressively reduce the concurrence problem. In Fig. A.9 the emotions MLD is used to show a simple example. The main metrics of the MLD and its labels are displayed after calling the `remedial` function once and twice. The differences are remarkable as can be seen.

```
Console ~/mldr/
>
> summary(emotions)
  num.attributes num.instances num.inputs num.labels num.labelsets num.single.labelsets max.frequency cardinality   density   meanIR    scumble scumble.cv
1             78           593         72          6            27                    4            81    1.868465 0.3114109 1.478068 0.01095238    1.26456
> summary(remedial(emotions))
  num.attributes num.instances num.inputs num.labels num.labelsets num.single.labelsets max.frequency cardinality   density   meanIR      scumble scumble.cv
1             78           815         72          6            19                    2           257    1.359509 0.2265849 1.478068 0.0005623093   2.566903
> summary(remedial(remedial(emotions)))
  num.attributes num.instances num.inputs num.labels num.labelsets num.single.labelsets max.frequency cardinality   density   meanIR      scumble scumble.cv
1             78          1043         72          6            13                    1           257     1.06232 0.1770534 1.478068 0.0003179895    3.98286
> emotions$labels
                index count      freq    IRLbl     SCUMBLE  SCUMBLE.CV
amazed-suprised    73   173 0.2917369 1.526012 0.002159173   2.4711227
happy-pleased      74   166 0.2799325 1.590361 0.014332319   0.8889398
relaxing-calm      75   264 0.4451939 1.000000 0.023786461   0.4839719
quiet-still        76   148 0.2495784 1.783784 0.023131538   0.6290932
sad-lonely         77   168 0.2833052 1.571429 0.016133470   0.8069413
angry-aggresive    78   189 0.3187184 1.396825 0.001331189   2.4881141
> remedial(remedial(emotions))$labels
                index count      freq    IRLbl      SCUMBLE SCUMBLE.CV
amazed-suprised    73   173 0.1658677 1.526012 7.523092e-05  1.3107425
happy-pleased      74   166 0.1591563 1.590361 1.413782e-04  2.3392930
relaxing-calm      75   264 0.2531160 1.000000 3.659295e-04  6.0592314
quiet-still        76   148 0.1418984 1.783784 1.500168e-03  0.5692001
sad-lonely         77   168 0.1610738 1.571429 1.259564e-03  0.7641657
angry-aggresive    78   189 0.1812081 1.396825 5.111396e-04  5.0990195
>
```

Fig. A9. The algorithm can be applied several times to progressively reduce the concurrence problem.

References

[1] F. Herrera, F. Charte, A.J. Rivera, M.J. del Jesus, Multilabel Classification. Problem analysis, metrics and techniques, Springer, 2016, doi:10.1007/978-3-319-41111-8.

[2] E. Gibaja, S. Ventura, A tutorial on multilabel learning, ACM Comput. Surv. 47 (3) (2015) 52:1–52:38, doi:10.1145/2716262.

[3] F. Charte, A.J. Rivera, M.J. del Jesus, F. Herrera, QUINTA: a question tagging assistant to improve the answering ratio in electronic forums, in: Proceedings of the EUROCON International Conference on Computer as a Tool (EUROCON), IEEE, 2015, pp. 1–6, doi:10.1109/EUROCON.2015.7313677.

[4] S. Diplaris, G. Tsoumakas, P. Mitkas, I. Vlahavas, Protein classification with multiple algorithms, in: Proceedings of the Tenth Panhellenic Conference on Informatics, Volos, Greece, 2005, pp. 448–456, doi:10.1007/11573036_42.

[5] Z. Wei, H. Wang, R. Zhao, Semi-supervised multi-label image classification based on nearest neighbor editing, Neurocomputing 119 (2013) 462–468, doi:10.1016/j.neucom.2013.03.011.

[6] N.V. Chawla, N. Japkowicz, A. Kotcz, Editorial: special issue on learning from imbalanced data sets, SIGKDD Explor. Newsl. 6 (1) (2004) 1–6, doi:10.1145/1007730.1007733.

[7] G. Tepvorachai, C. Papachristou, Multi-label imbalanced data enrichment process in neural net classifier training, Proceedings of the IEEE International Joint Conference on Neural Networks, 2008, pp. 1301–1307, doi:10.1109/IJCNN.2008.4633966.

[8] J. He, H. Gu, W. Liu, Imbalanced multi-modal multi-label learning for subcellular localization prediction of human proteins with both single and multiple sites, PloS One 7 (6) (2012) 7155, doi:10.1371/journal.pone.0037155.

[9] M.A. Tahir, J. Kittler, A. Bouridane, Multilabel classification using heterogeneous ensemble of multi-label classifiers, Pattern Recognit. Lett. 33 (5) (2012) 513–523, doi:10.1016/j.patrec.2011.10.019.

[10] M.A. Tahir, J. Kittler, F. Yan, Inverse random under sampling for class imbalance problem and its application to multi-label classification, Pattern Recognit. 45 (10) (2012) 3738–3750, doi:10.1016/j.patcog.2012.03.014.

[11] C. Li, G. Shi, Improvement of learning algorithm for the multi-instance multi-label RBF neural networks trained with imbalanced samples, J. Inf. Sci. Eng. 29 (4) (2013) 765–776.

[12] F. Charte, A.J. Rivera, M.J. del Jesus, F. Herrera, Addressing imbalance in multilabel classification: Measures and random resampling algorithms, Neurocomputing 163 (0) (2015) 3–16, doi:10.1016/j.neucom.2014.08.091.

[13] A.F. Giraldo-Forero, J.A. Jaramillo-Garzón, J.F.R.-M. noz, C.G. Castellanos-Domínguez, Managing imbalanced data sets in multi-label problems: a case study with the SMOTE algorithm, in: Proceedings of the Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, in: LNCS, Vol. 8258, Springer, 2013, pp. 334–342, doi:10.1007/978-3-642-41822-8_42.

[14] F. Charte, A.J. Rivera, M.J. del Jesus, F. Herrera, MLeNN: a first approach to heuristic multilabel undersampling, in: Proceedings of the Fifteenth International Conference Intelligent Data Engineering and Automated Learning, Vol. 8669 of LNCS, Salamanca, Spain, 2014, pp. 1–9, doi:10.1007/978-3-319-10840-7_1.

[15] F. Charte, A.J. Rivera, M.J. del Jesus, F. Herrera, MLSMOTE: approaching imbalanced multilabel learning through synthetic instance generation, Knowl. Based Syst. 89 (2015) 385–397, doi:10.1016/j.knosys.2015.07.019.

[16] V. García, J. Sánchez, R. Mollineda, On the effectiveness of preprocessing methods when dealing with different levels of class imbalance, Knowl. Based Syst. 25 (1) (2012) 13–21, doi:10.1016/j.knosys.2011.06.013.

[17] F. Charte, A. Rivera, M.J. Jesus, F. Herrera, Concurrence among imbalanced labels and its influence on multilabel resampling algorithms, in: Proceedings of the Ninth International Conference on Hybrid Artificial Intelligent Systems, Vol. 8480 of LNCS, Salamanca, Spain, 2014.

[18] F. Charte, D. Charte, Working with multilabel datasets in R: the mldr package, R J. 7 (2) (2015) 149–162. http://journal.r-project.org/archive/2015-2/charte-charte.pdf

[19] D. Turnbull, L. Barrington, D. Torres, G. Lanckriet, Semantic annotation and retrieval of music and sound effects, IEEE Audio Speech Lang. Process 16 (2) (2008) 467–476, doi:10.1109/TASL.2007.913750.

[20] B. Klimt, Y. Yang, The enron corpus: a new dataset for email classification research, in: Proceedings of the European Conference on Machine Learning, Pisa, Italy, 2004, pp. 217–226, doi:10.1007/978-3-540-30115-8_22.

[21] A. Elisseeff, J. Weston, A kernel method for multi-labelled classification, in: Proceedings of the Advances in Neural Information Processing Systems, vol. 14, MIT Press, 2001, pp. 681–687.

[22] K. Crammer, M. Dredze, K. Ganchev, P.P. Talukdar, S. Carroll, Automatic code assignment to medical text, in: Proceedings of the Workshop on Biological, Translational, and Clinical Language Processing, Prague, Czech Republic, 2007, pp. 129–136.

[23] G. Tsoumakas, I. Katakis, I. Vlahavas, Mining multi-label data, in: O. Maimon, L. Rokach (Eds.), Data Mining and Knowledge Discovery Handbook, vol. 34, Springer, Boston, MA, USA, 2010, pp. 667–685, doi:10.1007/978-0-387-09823-4_34.

[24] S. Godbole, S. Sarawagi, Discriminative methods for multi-labeled classification, in: Proceedings of the Advances in Knowledge Discovery and Data Mining, vol. 3056, 2004, pp. 22–30, doi:10.1007/978-3-540-24775-3_5.

[25] M. Boutell, J. Luo, X. Shen, C. Brown, Learning multi-label scene classification, Pattern Recognit. 37 (9) (2004) 1757–1771, doi:10.1016/j.patcog.2004.03.009.

[26] E.L. Mencía, S.H. Park, J. Fürnkranz, Efficient voting prediction for pairwise multilabel classification, Neurocomputing 73 (7–9) (2010) 1164–1176, doi:10.1016/j.neucom.2009.11.024.

[27] M. Zhang, Z. Zhou, ML-KNN: a lazy learning approach to multi-label learning, Pattern Recognit. 40 (7) (2007) 2038–2048, doi:10.1016/j.patcog.2006.12.019.

[28] A. Clare, R.D. King, Knowledge discovery in multi-label phenotype data, in: Proceedings of the Fifth European Conference on Principles on Data Mining and Knowledge Discovery, vol. 2168, Freiburg, Germany, 2001, pp. 42–53, doi:10.1007/3-540-44794-6_4.

[29] M.-L. Zhang, Z.-J. Wang, MIMLRBF: RBF neural networks for multi-instance multi-label learning, Neurocomputing 72 (16–18) (2009) 3951–3956, doi:10.1016/j.neucom.2009.07.008.

[30] A. Fernández, V. López, M. Galar, M.J. del Jesus, F. Herrera, Analysing the classification of imbalanced data-sets with multiple classes: binarization techniques and ad-hoc approaches, Knowl. Based Syst. 42 (2013) 97–110, doi:10.1016/j.knosys.2013.01.018.

[31] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284, doi:10.1109/TKDE.2008.239.

[32] V. López, A. Fernández, S. García, V. Palade, F. Herrera, An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics, Inf. Sci. 250 (2013) 113–141, doi:10.1016/j.ins.2013.07.007.

[33] R.C. Prati, G.E.A.P.A. Batista, D.F. Silva, Class imbalance revisited: a new experimental setup to assess the performance of treatment methods, Knowl. Inf. Syst. 45 (1) (2014) 247–270, doi:10.1007/s10115-014-0794-3.
[34] F. Charte, A. Rivera, M.J. Jesus, F. Herrera, A first approach to deal with imbalance in multi-label datasets, in: Proceedings of the Eight International Conference Hybrid Artificial Intelligent Systems, Vol. 8073 of LNCS, Salamanca, Spain, 2013, pp. 150–160, doi:10.1007/978-3-642-40846-5_16.
[35] A.B. Atkinson, On the measurement of inequality, J. Econ. Theory 2 (3) (1970) 244–263.
[36] F. Charte, A. Rivera, M.J. del Jesus, F. Herrera, Resampling multilabel datasets by decoupling highly imbalanced labels, in: Proceedings of the Hybrid Artificial Intelligent Systems, Lecture Notes in Computer Science, vol. 9121, Springer International Publishing, 2015, pp. 489–501, doi:10.1007/978-3-319-19644-2_41.
[37] P. Duygulu, K. Barnard, J. de Freitas, D. Forsyth, Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary, in: Proceedings of the Seventh European Conference on Computer Vision-Part IV, Copenhagen, Denmark, 2002, pp. 97–112, doi:10.1007/3-540-47979-1_7.
[38] C.G.M. Snoek, M. Worring, J.C. van Gemert, J.M. Geusebroek, A.W.M. Smeulders, The challenge problem for automated detection of 101 semantic concepts in multimedia, in: Proceedings of the Fourteenth Annual ACM International Conference on Multimedia, Santa Barbara, CA, USA, 2006, pp. 421–430, doi:10.1145/1180639.1180727.
[39] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D.M. Blei, M.I. Jordan, Matching words and pictures, J. Mach. Learn. Res. 3 (2003) 1107–1135.
[40] A.N. Srivastava, B. Zane-Ulman, Discovering recurring anomalies in text reports regarding complex space systems, in: Proceedings of the Aerospace Conference, IEEE, 2005, pp. 3853–3862, doi:10.1109/AERO.2005.1559692.
[41] I. Katakis, G. Tsoumakas, I. Vlahavas, Multilabel text classification for automated tag suggestion, in: Proceedings of the ECML PKDD'08 Discovery Challenge, Antwerp, Belgium, 2008, pp. 75–83.
[42] G. Tsoumakas, I. Katakis, I. Vlahavas, Effective and efficient multilabel classification in domains with large number of labels, in: Proceedings of the ECML/PKDD Workshop on Mining Multidimensional Data, Antwerp, Belgium, 2008, pp. 30–44.
[43] W. Cheng, E. Hüllermeier, Combining instance-based learning and logistic regression for multilabel classification, Mach. Learn. 76 (2–3) (2009) 211–225, doi:10.1007/s10994-009-5127-5.
[44] J. Fürnkranz, E. Hüllermeier, E. Loza Mencía, K. Brinker, Multilabel classification via calibrated label ranking, Mach. Learn. 73 (2008) 133–153, doi:10.1007/s10994-008-5064-8.
[45] F. Fabris, A.A. Freitas, J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, Mach. Learn. 85 (2014) 333–359.
[46] J. Read, B. Pfahringer, G. Holmes, Multi-label classification using ensembles of pruned sets, in: Proceedings of the Eight IEEE International Conference on Data Mining, Pisa, Italy, 2008, pp. 995–1000.
[47] J. Read, A pruned problem transformation method for multi-label classification, in: Proceedings of the New Zealand Computer Science Research Student Conference (NZCSRS 2008), 2008, pp. 143–150.
[48] G. Madjarov, D. Kocev, D. Gjorgjevikj, S. Džeroski, An extensive experimental comparison of methods for multi-label learning, Pattern Recognit. 45 (9) (2012) 3084–3104, doi:10.1016/j.patcog.2012.03.004.
[49] F. Charte, F.D. Charte, How to work with multilabel datasets in R using the mldr package, doi:10.6084/m9.figshare.1356035.
[50] G. Tsoumakas, E.S. Xioufis, J. Vilcek, I. Vlahavas, MULAN: a java library for multi-label learning, J. Mach. Learn. Res. 12 (2011) 2411–2414.
[51] J. Read, P. Reutemann, MEKA multi-label dataset repository, http://meka.sourceforge.net/#datasets.



Francisco Charte received his B.Eng. degree in Computer Science from the University of Jaén in 2010 and his M.Sc. and Ph.D. in Computer Science from the University of Granada in 2011 and 2015. He is currently a post-doctoral researcher at the University of Granada (Spain). His main research interests include machine learning with applications to multilabel classification, high dimensionality and imbalance problems, and association rule mining, as well as CPU/GPU algorithm parallelization techniques.



Antonio J. Rivera received his B.Sc. degree and his Ph.D. in Computer Science from the University of Granada in 1995 and 2003, respectively. He is a lecturer of Computer Architecture and Computer Technology with the Computer Science Department at the University of Jaén (Spain). His research interests include areas such as multilabel classification, imbalance problems, evolutionary computation, neural network design, time series prediction and regression tasks.



María J. Del Jesus received the M.Sc. and Ph.D. degrees in Computer Science from the University of Granada, Granada, Spain, in 1994 and 1999, respectively. She is an Associate Professor with the Department of Computer Science, University of Jaén, Spain. Her current research interests include fuzzy rule-based systems, genetic fuzzy systems, subgroup discovery, data preparation, feature selection, evolutionary radial basis neural networks, knowledge extraction based on evolutionary algorithms, and data mining.



Francisco Herrera received his M.Sc. in Mathematics in 1988 and Ph.D. in Mathematics in 1991, both from the University of Granada, Spain. He is currently a Professor in the Department of Computer Science and Artificial Intelligence at the University of Granada. He has been the supervisor of 38 Ph.D. students. He has published more than 300 papers in international journals. He is coauthor of the books "Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases" (World Scientific, 2001), "Data Preprocessing in Data Mining" (Springer, 2015), "The 2-tuple Linguistic Model. Computing with Words in Decision Making" (Springer, 2015), "Multilabel Classification. Problem analysis, metrics and techniques" (Springer, 2016). He currently acts as Editor in Chief of the international journals "Information Fusion" (Elsevier) and "Progress in Artificial Intelligence" (Springer). He acts as editorial board member or associate editor of a dozen of journals. He received the following honors and awards: ECCAI Fellow 2009, IFSA Fellow 2013, 2010 Spanish National Award on Computer Science ARITMEL to the "Spanish Engineer on Computer Science", International Cajastur "Mamdani' Prize for Soft Computing (Fourth Edition, 2010), IEEE Transactions on Fuzzy System Outstanding 2008 and 2012 Paper Award (bestowed in 2011 and 2015 respectively), 2011 Lotfi A. Zadeh Prize Best paper Award of the International Fuzzy Systems Association, 2013 AEPIA Award to a scientific career in Artificial Intelligence (September 2013), and 2014 XV Andalucía Research Prize Maimónides (by the regional government of Andalucía). He belongs to the list of the Highly Cited Researchers in the areas of Engineering and Computer Sciences: http://highlycited.com/ (Thomson Reuters, 2015). His h-index is 109 in Scholar Google, receiving more than 44,000 citations. His current research interests include among others, soft computing (including fuzzy modeling and evolutionary algorithms), information fusion, decision making, bibliometrics, biometric, data preprocessing, data science and big data.