Neurocomputing 163 (2015) 3–16

Contents lists available at ScienceDirect

Neurocomputing

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

Addressing imbalance in multilabel classification: Measures and random resampling algorithms

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

[a] Department of Computer Science and A.I., University of Granada, 18071 Granada, Spain
[b] Department of Computer Science, University of Jaén, 23071 Jaén, Spain
[c] Faculty of Computing and Information Technology - North Jeddah, King Abdulaziz University, 21589, Jeddah, Saudi Arabia

ARTICLE INFO

Article history:
Received 29 October 2013
Received in revised form
28 February 2014
Accepted 11 August 2014
Available online 16 April 2015

Keywords:
Multilabel classification
Imbalanced classification
Resampling algorithms
Undersampling
Oversampling

ABSTRACT

The purpose of this paper is to analyze the imbalanced learning task in the multilabel scenario, aiming to accomplish two different goals. The first one is to present specialized measures directed to assess the imbalance level in multilabel datasets (MLDs). Using these measures we will be able to conclude which MLDs are imbalanced, and therefore would need an appropriate treatment. The second objective is to propose several algorithms designed to reduce the imbalance in MLDs in a classifier-independent way, by means of resampling techniques. Two different approaches to divide the instances in minority and majority groups are studied. One of them considers each label combination as class identifier, whereas the other one performs an individual evaluation of each label imbalance level. A random undersampling and a random oversampling algorithm are proposed for each approach, giving as result four different algorithms. All of them are experimentally tested and their effectiveness is statistically evaluated. From the results obtained, a set of guidelines directed to show when these methods should be applied is also provided.

1. Introduction

Multilabel classification (MLC) [1] is receiving significant attention lately, and it is being applied in fields such as text categorization [2] and music labeling [3]. In these scenarios, each data sample is associated with several concepts (class labels) simultaneously. Therefore, MLC algorithms have to be able to give several outputs as result, instead of only one as in traditional classification.

The data used for learning a classifier is often imbalanced. Thus, the class labels assigned to each instance are not equally represented. This is a profoundly examined problem in binary datasets [4] and to a lesser extent to multiclass datasets [5]. A measure called *imbalance ratio* (IR) [4] is used to know the datasets' imbalance level. Traditionally, imbalanced classification has been faced through techniques [6] such as resampling, cost-sensitive learning, and algorithmic-specific adaptations.

That most MLDs suffer from a high level of imbalance is a commonly accepted fact in the literature [7]. However, there are not specific measures to assess the imbalance level in MLDs. Thus, the imbalanced nature of MLDs is more an assumption than an established fact. To date, there are some proposals to deal with imbalanced MLDs focused in algorithmic adaptations of MLC algorithms [7–9], so they are classifier-dependent solutions. An alternative classifier-independent way to address the imbalance in MLDs would be by means of preprocessing techniques, with resampling algorithms in particular. This approach would allow the use of any state-of-the-art MLC algorithm.

In this paper, we tackle the mentioned imbalanced problem for MLDs from a double perspective, the analysis of the imbalance level and proposals for reducing the imbalance in MLDs.[1]

There is a need for specific measures that can be used to obtain information about the imbalance level in MLDs. Three measures directed to assess the MLDs imbalance level are introduced and discussed.

Four resampling methods aimed at reducing the imbalance in MLDs are proposed. The measures will offer a convenient guide to know if an MLD suffers from imbalance or not, and therefore when it could benefit from the preprocessing. Regarding the resampling methods, undersampling and oversampling were the reasonable techniques to follow, although the difficulty on how to deal with

* Corresponding author. Tel.: +34 953 212 892; fax: +34 953 212 472.
E-mail addresses: 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 expanded version of our previous work [33] from HAIS'13, including new preprocessing proposals and a vastly extended experimental study.

http://dx.doi.org/10.1016/j.neucom.2014.08.091
0925-2312/

multiple labels has to be solved. We examine two different approaches:

- One of them is based on the Label Powerset (LP) transformation, evaluating the frequency of full labelsets. Two algorithms founded on this approach were introduced in [33], one performs random undersampling (LP-RUS) and the other one random oversampling (LP-ROS).
- The second approach evaluates the frequency of individual labels, instead of full labelsets, isolating the instances in which one or more minority labels appear. Based on this approach another two algorithms are proposed, one for random undersampling (ML-RUS) and the other one for random oversampling (ML-ROS).

The usefulness of the measures and effectiveness of the methods are proven experimentally, using different MLDs and MLC algorithms, and the results are thoroughly analyzed using statistical tests. The conducted experimentation is used as an exploratory test on how known resampling algorithms could be adapted to the multilabel scenario.

The rest of this paper is structured as follows: Section 2 briefly describes the MLC task and the learning from imbalanced data problem. Section 3 introduces the imbalance problem in MLC, and describes the proposed measures to assess the imbalance level in MLDs. The resampling methods proposal is presented in Section 4. In Section 5, the experimental framework is described, and the results obtained are analyzed. Finally, the conclusions are given in Section 6.

2. Preliminaries

MLC usually demands more complex models than traditional classification to be faced. As traditional datasets, class distribution in MLDs frequently involves some imbalance level. The imbalance level in MLDs tends to be higher indeed. This characteristic makes this task even more challenging. In this section, MLC and classification with imbalanced data problems are introduced.

2.1. Multilabel classification

In many application domains [2,3,10] each data sample is associated with a set of labels, instead of only one class label as in traditional classification. Therefore, with Y being the total set of labels in an MLD D and x_i a sample in D, a multilabel classifier h must produce as output a set $Z_i \subseteq Y$ with the predicted labels for the i-th sample. As each distinct label in Y could appear in Z_i, the total number of potential different combinations would be $2^{|Y|}$. Each one of these combinations is called a *labelset*. The same labelset can appear in several instances of D.

There are two main approaches [1] to accomplish an MLC task: data transformation and algorithm adaptation. The former aims to produce from an MLD a dataset or group of datasets that can be processed with traditional classifiers, while the objective of the latter is to adapt existent classification algorithms in order to work with MLDs. Among the transformation methods, the most popular are those based on the binarization of the MLD, such as *Binary Relevance* (BR) [11] and *Ranking by Pairwise Comparison* [12], and the LP [13] transformation, which produces a multiclass dataset from an MLD considering each labelset as class. In the algorithm adaptation approach there are proposals of multilabel C4.5 trees [14], algorithms based on nearest neighbors such as ML-kNN [15], multilabel neural networks [2,16], and multilabel SVMs [17].

In the literature there are some specific measures to characterize MLDs, such as label cardinality *Card*, defined as shown in Eq. (1), and label density *Dens*, Eq. (2). The former is the average number of active labels per sample in an MLD, while the latter is designed to obtain a dimensionless measure:

$$Card(D) = \sum_{i=1}^{|D|} \frac{|Y_i|}{|D|}. \tag{1}$$

$$Dens(D) = \frac{Card(D)}{|Y|}. \tag{2}$$

A recent review on multilabel learning algorithms can be found in [18].

2.2. Classification with imbalanced data

The learning from imbalanced data problem is founded on the different distributions of class labels in the data [19], and it has been thoroughly studied in traditional classification. In this context, the measurement of the imbalance level in a dataset is obtained as the ratio of the number of samples of the majority class and the number associated with the minority class, being known as IR [4]. The higher the IR, the larger the imbalance level. The difficulty in the learning process with this kind of data is due to the design of most classifiers, as their main goal is to reduce some global error rate [4]. This approach tends to penalize the classification of minority classes.

Generally, the imbalance problem has been faced with three different approaches [6]: data resampling, algorithmic adaptations [5], and cost sensitive classification [20]. The former is based on the rebalancing of class distributions through resampling algorithms, either deleting instances of the most frequent class (undersampling) or adding new instances of the least frequent one (oversampling). Random undersampling (RUS) [21], random oversampling (ROS) and SMOTE [22] are among the most used resampling methods to equilibrate imbalanced datasets. The advantage of this approach is in that it can be applied as a general method to solve the imbalance problem, independent of the classification algorithms used once the datasets have been preprocessed. A general overview on imbalanced learning can be found in [23].

2.3. Learning from imbalanced MLDs

Conventional resampling methods are designed to work with one output class only. Each sample in an MLD is associated with more than one class, and this is a fact to be taken into account. Furthermore, those methods usually assume that there are only one minority label and one majority label, whereas in MLDs with hundreds of labels many of them can be considered as minority/majority cases. Thus, an approach to resample MLDs, which have a set of labels as output and several of them could be considered minority/majority labels, is needed.

Most of the published algorithms aim to deal with the imbalance problem by means of algorithmic adaptations of MLC classifiers, or the use of ensembles of classifiers. Furthermore all of them are classifier-dependent, instead of general application methods able to work with another MLC learning algorithms. Some of the existent proposals are the following:

- *Ensemble Multilabel Learning* [7] is a method based on the use of heterogeneous algorithms to build an ensemble of MLC classifiers. The authors aim to face two problems simultaneously, learning from imbalanced data and capturing correlation information among labels. The ensemble is composed of five well-known MLC algorithms, being able to improve classification results in some configurations.

- The algorithm proposed in [8], called *Inverse Random Undersampling*, was originally designed for traditional classification, but the authors also did some experimentation with MLDs. The basic idea is to train several classifiers using all the instances associated with the minority class, while taking only a small random subset of the majority class instances for each classifier. The adaptation to MLC is made leaning on the BR transformation method.
- In [9] the problem faced is the prediction of subcellular localizations of human proteins, a highly imbalanced MLC task. The algorithm proposed is based on the use of Gaussian Process, a Bayesian method used to build non-parametric probabilistic models. By means of a covariance matrix the correlations among labels are obtained, and the imbalance is fixed associating a weight coefficient to each sample.

These methods are tied to one or more MLC algorithms by design, and have proven that classifier-dependent algorithms are a convenient path to deal with the imbalance problem. Notwithstanding, there are other interesting ways for facing it, in particular the data preprocessing approach. There are several advantages in this approach. Being independent of the classification process, it can be applied without interfering with working MLC systems. Moreover, the separation of tasks allows each algorithm to be focused in their specific work. Since this is an unexplored approach, we believe it is worth examining its possibilities. In Section 4 we propose different algorithms aimed to reduce the imbalance level in MLDs based on resampling techniques.

3. How to assess the imbalance level in MLDs

In this section the specific characteristics of imbalanced MLDs are discussed, and the proposed measures to assess the MLDs imbalance level are described.

3.1. Imbalance characteristics in MLDs

Most MLDs [24] have hundreds of labels, with each instance being associated with a subset of them. Intuitively, it is easy to see that the more different labels exist, the more possibilities there are that some of them have a very low/high presence.

In Fig. 1, which represents the sample distribution per label of bibtex and enron datasets, this fact can be verified. The leftmost bar on each subfigure corresponds to the most frequent label, whereas the rightmost bar represents the least frequent one. This would be the most extreme imbalance ratio in the MLD, but similar differences exist among any of the other pairs. However, it is not straightforward to infer the imbalance level from measures such as *Card* and *Dens*, which are the most widely used in the literature in order to characterize MLDs. The 3.378 *Card* value in enron indicates that each sample of this MLD is associated with slightly more than 3 labels, on average. Therefore, any of its 46 labels could appear together with other two or three labels in the same instance. This includes combinations in which the most frequent and least frequent labels appear jointly, but neither *Card* nor *Dens* is designed to consider this casuistry.

Many of the proposals made in the literature [7–9] for dealing with imbalanced datasets in MLC claim the imbalanced nature of MLDs. However, none of them give a measurement of the imbalance level, nor offer a procedure to measure it. Comparing Fig. 1(a) and (b), it is easy to see that in the latter the imbalance problem is much more prominent than in the former. Although a few very frequent labels exist in bibtex, 3 stand out of 159, the remainder ones appear in a quite similar number of instances. By contrast, the differences in enron are much more remarkable. Nonetheless, assessing the imbalance level only by means of a graphical representation is not an accurate procedure. Therefore, there is a need for specific measures that can be used to obtain information about the imbalance level in MLDs.

3.2. Measures to assess the imbalance level in MLDs

In binary classification, the imbalance level is measured taking into account only two classes: the majority class and the minority class. Many MLDs have hundreds of labels, and several of them may have a very low/high presence. For that reason, it is important to define the level of imbalance in MLC considering not only two labels, but all of them. In this scenario, we propose the use of the following measures, which were introduced in [33].

3.2.1. IRLbl: imbalance ratio per label

With D being an MLD with a set of labels Y, and Y_i the i-th label, it is calculated for the label y as the ratio between the majority label and the label y, as shown in Eq. (3). This value will be 1 for the most frequent label and a greater value for the rest. The larger the *IRLbl* is, the higher would be the imbalance level for the considered label:

$$IRLbl(y) = \frac{\operatorname{argmax}_{y'=Y_1}^{Y_{|Y|}}\left(\sum_{i=1}^{|D|} h(y', Y_i)\right)}{\sum_{i=1}^{|D|} h(y, Y_i)}, \quad h(y, Y_i) = \begin{cases} 1, & y \in Y_i \\ 0, & y \notin Y_i. \end{cases} \tag{3}$$

3.2.2. MeanIR: mean imbalance ratio

This measure will offer a value that represents the average level of imbalance in an MLD, obtained as shown in Eq. (4). It must be taken into account that different label distributions can produce the same *MeanIR* value; hence this measure should always be used jointly with the following:

$$MeanIR = \frac{1}{|Y|} \sum_{y=Y_1}^{Y_{|Y|}} (IRLbl(y)). \tag{4}$$

3.2.3. CVIR: coefficient of variation of IRLbl

This is the coefficient of variation of *IRLbl*, and is calculated as shown in Eq. (5). It will indicate if all labels suffer from a similar level of imbalance or, on the contrary, there are big differences in them. The larger the *CVIR* value, the higher would be this difference:

$$CVIR = \frac{IRLbl\sigma}{MeanIR}, \quad IRLbl\sigma = \sqrt{\sum_{y=Y_1}^{Y_{|Y|}} \frac{(IRLbl(y) - MeanIR)^2}{|Y| - 1}} \tag{5}$$

Table 1 shows these three measures for 13 well-known MLDs. The high *MeanIR* and *CVIR* values for corel5k and mediamill suggest that these MLDs are the most imbalanced, and therefore they could be the more benefited from the resampling. Several MLDs measurements also indicate that they have different levels of imbalance. On the contrary, the values associated with emotions and scene denote their nature of well-balanced MLDs, despite the fact that they have two of the highest *Dens* values. As can be seen, the differences highlighted before between bibtex and enron, based only on the observation of Fig. 1, are confirmed by their *MeanIR* and *CVIR* values. The extreme frequency differences between the most and least represented labels in enron are denoted by $MaxIR = 913$, whereas for bibtex the value is barely above 20. The average imbalance level of enron, with $MeanIR = 83.9528$ and $CVIR = 1.9596$, is also much higher than in bibtex, with $MeanIR = 12.4983$ and $CVIR = 0.4051$. Although in this case the *Card* and *Dens* values in enron are slightly higher than in bibtex, in general there are not a correlation between these two measures and the imbalance level in each MLD.

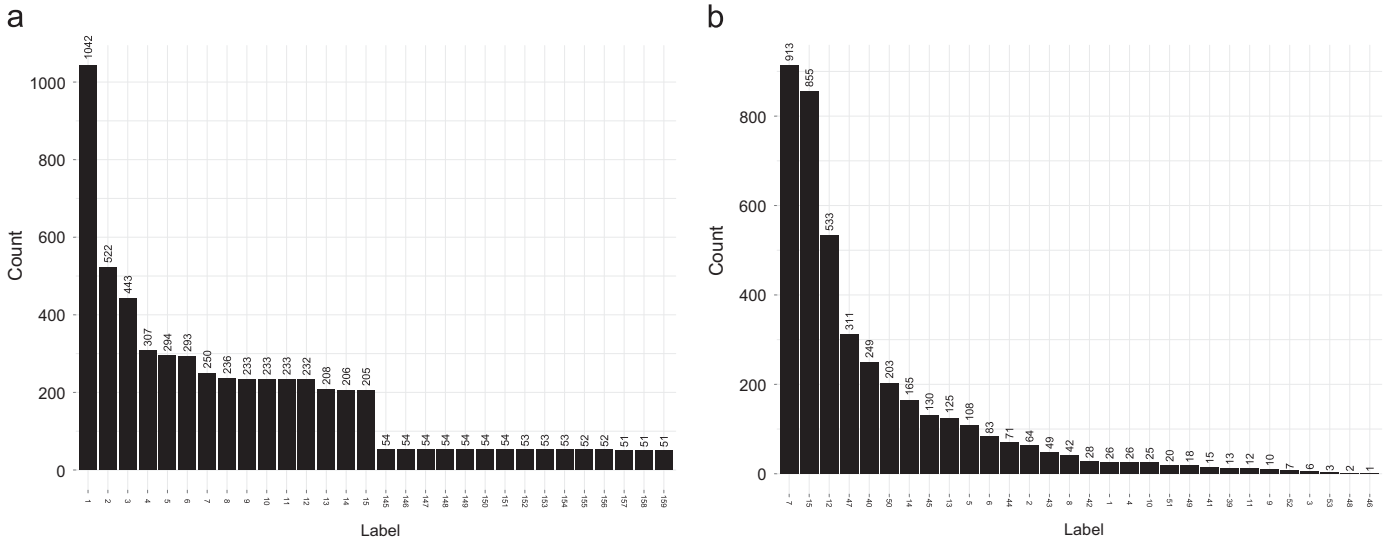

Fig. 1. Number of instances in which appear the 15 most common labels (left side of each picture) and the 15 rarest ones (right): (a) bibtex dataset. *Card*=2.402, *Dens*=0.015 and (b) enron dataset. *Card*=3.378, *Dens*=0.064.

Table 1
Basic characteristics and imbalance measures of datasets.

Dataset	Instances	Features	Labels	Card	Dens	MaxIR	MeanIR	CVIR	Ref
bibtex	7395	1836	159	2.402	0.015	20.4314	12.4983	0.4051	[25]
cal500	502	68	174	26.044	0.150	88.8000	20.5778	1.0871	[26]
corel5k	5000	499	374	3.522	0.009	1120.0000	189.5676	1.5266	[27]
corel16k	13,766	500	161	2.867	0.018	126.8000	34.1552	0.8088	[28]
emotions	593	72	6	1.868	0.311	1.7838	1.4781	0.1796	[3]
enron	1702	753	53	3.378	0.064	913.0000	73.9528	1.9596	[29]
genbase	662	1186	27	1.252	0.046	171.0000	37.3146	1.4494	[10]
llog	1460	1004	75	1.180	0.016	171.0000	39.2669	1.3106	[30]
mediamill	43,907	120	101	4.376	0.043	1092.5484	256.4047	1.1749	[31]
scene	2407	294	6	1.074	0.179	1.4643	1.2538	0.1222	[13]
slashdot	3782	1079	22	1.181	0.054	194.6667	19.4624	2.2878	[30]
tmc2007	28,596	49,060[a]	22	2.158	0.098	41.9802	17.1343	0.8137	[32]
yeast	2417	198	14	4.237	0.303	53.4118	7.1968	1.8838	[17]

[a] The 500 most relevant features were selected.

Unlike *Card* and *Dens*, which only evidence the mean number of labels per instance, the joint use of the *MeanIR* and *CVIR* measures would denote if an MLD is imbalanced or not, whereas *IRLbl* would be useful in individual label evaluation. As a general rule, any MLD with a *MeanIR* value higher than 1.5 (50% more of samples with majority label vs minority label, in average) and *CVIR* value above 0.2 (20% of variance in the *IRLbl* values) should be considered as imbalanced.

Table 2
Average rankings for undersampling algorithms.

Algorithm	Accuracy		Micro-FM		Macro-FM	
LP-RUS 10	3.114	↓	2.795	↔	2.523	⋆
LP-RUS 20	4.227	⇊	3.523	⇊	3.443	↔
LP-RUS 25	5.034	⇊	4.432	⇊	3.841	⇊
ML-RUS 10	2.136	⋆	2.193	⋆	3.034	↔
ML-RUS 20	3.023	↓	2.716	↔	3.988	⇊
ML-RUS 25	3.466	⇊	5.341	⇊	4.170	⇊

4. Resampling algorithms for reducing imbalance in MLDs

In order to design any resampling algorithm for MLDs, the first point to consider is how the specific nature of this kind of datasets will be addressed, as the output variable is not a class but also a group of them. Two approaches are followed in the present study for deciding what cases can be considered minority or majority. The following subsections discuss these approaches and describe each one of the proposed algorithms.

4.1. LP based resampling algorithms

The first approach relies on the LP transformation method [13]. This basic method transforms the MLD in a multiclass dataset, processing each different combination of labels (*labelset*) as class identifier. A maximum of $2^{|L|}$ distinct labelsets could exist in an MLD. It is an approach that has been successfully used as base transformation in classification algorithms such as RAkEL [34] and HOMER [35]. Although BR transformation is maybe more popular when it comes to design new MLC algorithms, LP has been also used to complete other kinds of tasks such as the stratified partitioning of MLDs [36]. Moreover, facing multilabel resampling through BR would imply a pair-wise imbalance treatment, instead of the joint treatment that could be achieved through LP. Therefore, it is a technique that deserves to be tested in the context of the problem at hand. LP-RUS and LP-ROS interpret each labelset as class identifier while resampling an MLD, deleting majority labelsets and cloning minority labelsets.

LP-based resampling methods do not rely on the measures proposed in Section 3 to do their work. Hence a previous measurement of the MLD imbalance level has to be performed, then deciding if the resampling could benefit or not classification results. As the experimentation in [33] showed, those MLDs with a *MeanIR* < 1.5 and a low *CVIR* value, below 0.5, could hardly get any benefit from the resampling.

Table 3
Average rankings for oversampling algorithms.

Algorithm	Accuracy		Micro-FM		Macro-FM	
LP-ROS 10	4.830	⇊	5.193	⇊	5.170	⇊
LP-ROS 20	4.489	⇊	4.943	⇊	4.807	⇊
LP-ROS 25	4.307	⇊	4.773	⇊	4.591	⇊
ML-ROS 10	2.284	⋆	1.409	⋆	2.250	↔
ML-ROS 20	2.636	↔	2.080	↔	2.160	↔
ML-ROS 25	2.455	↔	2.602	⇊	2.023	⋆

Table 4
Final stage – average rankings.

Algorithm	Accuracy		Micro-FM		Macro-FM	
Base	1.852	↔	2.636	⇊	1.898	↔
ML-RUS 10	2.364	⇊	1.568	⋆	2.386	⇊
ML-ROS 10	1.784	⋆	1.795	↔	1.716	⋆

4.1.1. LP-RUS: LP based random undersampling

LP-RUS is a multilabel undersampling algorithm that works deleting random samples of majority labelsets. The process will stop when the MLD *D* is reduced by the indicated percentage. The method accomplishes this task as shown in Algorithm 1.

This procedure aims to achieve a labelset representation in the MLD as close as possible to a uniform distribution. However, since a limit on the minimum dataset size has been established with the *P* parameter, a certain degree of imbalance among the labelsets could remain in the MLD. In any case, the imbalance level should always be lower than in the original dataset.

Algorithm 1. LP-RUS algorithm's pseudo-code.

Inputs: ⟨Dataset⟩ *D*, ⟨Percentage⟩ *P*
Outputs: Preprocessed dataset
samplesToDelete ← |*D*| /100∗*P* ▷ *P%* size reduction
▷ Group samples according to their labelsets

Table A1
Undersampling algorithms results – accuracy.

Algorithm	Dataset	LP-RUS 10	LP-RUS 20	LP-RUS 25	ML-RUS 10	ML-RUS 20	ML-RUS 25
CLR	bibtex	0.2043	0.1958	0.1880	**0.2292**	0.2286	0.2261
HOMER	bibtex	0.2409	0.2216	0.2193	**0.2618**	0.2603	0.2580
IBLR	bibtex	0.1424	0.1276	0.1221	**0.1684**	0.1654	0.1652
RAkEL	bibtex	0.2776	0.2646	0.2657	**0.2966**	0.2937	0.2933
CLR	cal500	0.1787	0.1787	0.1787	0.1708	0.1771	**0.1796**
HOMER	cal500	0.2410	**0.2500**	0.2455	0.2346	0.2324	0.2348
IBLR	cal500	**0.1926**	**0.1926**	**0.1926**	0.1898	0.1843	0.1897
RAkEL	cal500	0.2135	0.2135	0.2135	0.2101	0.2140	**0.2154**
CLR	corel16k	0.0434	0.0409	0.0401	0.0453	0.0454	**0.0466**
HOMER	corel16k	0.1080	0.1087	0.1023	0.1118	0.1112	**0.1132**
IBLR	corel16k	0.0224	0.0208	0.0200	**0.0256**	0.0254	0.0250
RAkEL	corel16k	0.0618	0.0592	0.0592	0.0633	0.0628	**0.0638**
CLR	corel5k	0.0328	0.0319	0.0292	0.0355	**0.0368**	0.0363
HOMER	corel5k	0.0956	0.0870	0.0907	**0.1016**	0.0977	0.1010
IBLR	corel5k	0.0266	0.0253	0.0235	0.0296	0.0323	**0.0324**
RAkEL	corel5k	0.0534	0.0518	0.0480	0.0589	0.0592	**0.0594**
CLR	enron	0.3912	0.3456	0.3422	**0.4184**	0.4156	0.4156
HOMER	enron	0.3822	0.3353	0.3349	**0.4085**	0.4026	0.3976
IBLR	enron	0.2834	0.2756	0.2752	0.3005	0.2946	**0.3147**
RAkEL	enron	0.3812	0.3324	0.3292	**0.4034**	0.3966	0.4004
CLR	genbase	**0.9822**	0.9816	0.9812	0.9716	0.9528	0.9368
HOMER	genbase	0.9802	**0.9822**	0.9796	0.9764	0.9582	0.9411
IBLR	genbase	0.9785	**0.9795**	0.9770	0.9671	0.9476	0.9214
RAkEL	genbase	**0.9842**	**0.9842**	0.9839	0.9782	0.9616	0.9456
CLR	llog	0.0338	0.0278	0.0239	0.0458	**0.0504**	0.0492
HOMER	llog	0.0866	0.0927	0.0889	**0.1038**	0.0992	0.0972
IBLR	llog	0.0328	0.0288	0.0232	**0.0352**	0.0327	0.0350
RAkEL	llog	0.1243	0.1261	0.1268	**0.1325**	0.1296	0.1258
CLR	mediamill	**0.4456**	0.4370	0.4342	0.4438	0.4368	0.4334
HOMER	mediamill	0.4026	0.3870	0.3842	**0.4089**	0.4037	0.4012
IBLR	mediamill	**0.4620**	0.4539	0.4486	0.4590	0.4504	0.4455
RAkEL	mediamill	0.4115	0.4013	0.3964	**0.4144**	0.4088	0.4072
CLR	slashdot	0.2816	0.2152	0.2024	**0.3194**	0.3117	0.2991
HOMER	slashdot	**0.3403**	0.3082	0.3216	0.3314	0.3226	0.3084
IBLR	slashdot	0.0880	0.0854	0.0811	0.1486	0.1364	**0.1548**
RAkEL	slashdot	0.3015	0.2352	0.2188	**0.3392**	0.3279	0.3159
CLR	tmc2007	**0.6061**	0.5938	0.5887	0.6020	0.5860	0.5774
HOMER	tmc2007	**0.5930**	0.5760	0.5690	0.5897	0.5801	0.5701
IBLR	tmc2007	**0.5266**	0.5168	0.5113	0.5184	0.5119	0.5072
RAkEL	tmc2007	**0.5950**	0.5806	0.5731	0.5913	0.5758	0.5676
CLR	yeast	0.4621	0.4556	0.4566	**0.4706**	0.4649	0.4549
HOMER	yeast	0.4294	0.4130	0.4086	**0.4312**	0.4294	0.4144
IBLR	yeast	0.5148	0.5048	0.5017	**0.5150**	0.5102	0.5074
RAkEL	yeast	0.4242	0.4144	0.4160	**0.4344**	0.4314	0.4234

Table A2
Undersampling algorithms results – micro-FMeasure.

Algorithm	Dataset	LP-RUS 10	LP-RUS 20	LP-RUS 25	ML-RUS 10	ML-RUS 20	ML-RUS 25
CLR	bibtex	0.7607	0.7429	0.7322	**0.7793**	0.7740	0.3339
HOMER	bibtex	0.3530	0.3346	0.3338	**0.3648**	0.3592	0.3408
IBLR	bibtex	**0.3610**	0.3178	0.2986	0.3494	0.3158	0.2423
RAkEL	bibtex	0.4878	0.4645	0.4573	**0.5136**	0.5045	0.3978
CLR	cal500	0.6227	0.6227	0.6227	**0.6258**	0.6151	0.2993
HOMER	cal500	0.3749	0.3822	**0.3858**	0.3732	0.3635	0.3787
IBLR	cal500	0.2827	0.2827	0.2827	0.2777	0.2720	**0.3166**
RAkEL	cal500	**0.4343**	**0.4343**	**0.4343**	0.4188	0.4163	0.3518
CLR	corel16k	0.4188	0.4054	0.3917	**0.4300**	0.4239	0.0878
HOMER	corel16k	0.2212	0.2152	0.2130	**0.2327**	0.2256	0.1868
IBLR	corel16k	0.2865	0.2356	0.2092	**0.3049**	0.2600	0.0498
RAkEL	corel16k	0.3464	0.3369	0.3339	0.3511	**0.3556**	0.1144
CLR	corel5k	0.4494	0.4440	0.4248	0.4512	**0.4540**	0.0714
HOMER	corel5k	0.2050	0.1964	0.1965	**0.2086**	0.2041	0.1708
IBLR	corel5k	0.0378	0.0356	0.0337	0.0434	0.0434	**0.0558**
RAkEL	corel5k	0.3716	**0.3756**	0.3652	0.3707	0.3717	0.1108
CLR	enron	**0.6842**	0.6677	0.6544	0.6780	0.6814	0.5467
HOMER	enron	0.5318	0.5111	0.5004	**0.5482**	0.5446	0.5086
IBLR	enron	**0.5660**	0.5418	0.5299	0.5588	0.5279	0.4312
RAkEL	enron	**0.6262**	0.6034	0.5858	0.6195	0.6124	0.5218
CLR	genbase	**0.9887**	**0.9887**	**0.9887**	0.9844	0.9852	0.9374
HOMER	genbase	0.9898	**0.9910**	0.9886	0.9852	0.9786	0.9420
IBLR	genbase	0.9768	**0.9799**	0.9767	0.9478	0.8998	0.8920
RAkEL	genbase	0.9893	0.9893	0.9893	0.9875	**0.9902**	0.9488
CLR	llog	0.5574	0.5198	0.4866	**0.5863**	0.5851	0.0802
HOMER	llog	0.1378	0.1468	0.1400	**0.1594**	0.1542	0.1452
IBLR	llog	0.0535	0.0440	0.0420	0.0580	0.0565	**0.0588**
RAkEL	llog	0.2698	0.2618	0.2634	**0.2880**	0.2871	0.1914
CLR	mediamill	0.7664	0.7560	0.7508	**0.7750**	0.7720	0.5713
HOMER	mediamill	0.5602	0.5266	0.5234	0.5882	**0.5944**	0.5372
IBLR	mediamill	0.7525	0.7398	0.7326	0.7701	**0.7708**	0.5728
RAkEL	mediamill	0.6279	0.6105	0.6013	0.6510	**0.6558**	0.5449
CLR	slashdot	0.6546	0.6520	**0.6639**	0.6315	0.6354	0.4084
HOMER	slashdot	0.5781	0.5683	0.5598	**0.5997**	0.5874	0.4001
IBLR	slashdot	0.6504	**0.6634**	0.6554	0.6505	0.6218	0.2242
RAkEL	slashdot	0.6985	**0.7146**	0.7098	0.6675	0.6672	0.4248
CLR	tmc2007	0.7400	0.7278	0.7228	**0.7500**	0.7454	0.6884
HOMER	tmc2007	0.6860	0.6691	0.6614	**0.6908**	0.6893	0.6664
IBLR	tmc2007	0.7132	0.7073	0.7036	**0.7221**	0.7220	0.6170
RAkEL	tmc2007	0.7243	0.7092	0.7016	**0.7337**	0.7263	0.6699
CLR	yeast	0.6462	0.6380	0.6286	0.6516	**0.6573**	0.6055
HOMER	yeast	0.5632	0.5504	0.5407	0.5680	**0.5795**	0.5612
IBLR	yeast	0.7133	0.7124	0.7083	**0.7137**	0.7129	0.6378
RAkEL	yeast	0.5839	0.5712	0.5644	0.5913	**0.5970**	0.5696

```
for i = 1 → |labelsets| do
    labelSetBag_i ← samplesWithLabelset(i)
end for
▷ Calculate the average number of samples per labelset
meanSize ← 1/|labelsets| * Σ_{i=1}^{|labelsets|} |labelSetBag_i|
▷ Obtain majority labels bags
for each labelSetBag_i in labelSetBag do
    if |labelSetBag_i| > meanSize then
        majBag_i ← labelSetBag_i
    end if
end for
meanRed ← samplesToDelete/|majBag|
majBag ← SortFromSmallestToLargest(majBag)
▷ Calculate # of instances to delete and remove them
for each majBag_i in majBag do
    rBag_i ← min(|majBag_i| − meanSize, meanRed)
    remainder ← meanRed − rBag_i
    distributeAmongBags_{j>i}(remainder)
    for n = 1 → rBag_i do
        x ← random(1, |majBag_i|)
        deleteSample(majBag_i, x)
    end for
end for
```

Although the imbalance level is evaluated by labelset, this method removes samples belonging to several label combinations, not only the most imbalanced one.

4.1.2. LP-ROS: LP based random oversampling

LP-ROS is a multilabel oversampling method that clones random samples associated with minority labelsets, until the size of the MLD is *P%* larger than the original. The procedure followed is analogous to the described above for LP-RUS. In this case, a collection of minority groups $minBag_i$ with ($|labelsetBag_i| < meanSize$) is obtained, a $meanInc = samplesGenerate/minBag$ is calculated, and processing the minority groups from the largest to the smallest an individual increment for each $minBag_i$ is determined. If a $minBag_i$ reaches *meanSize* samples before $incrementBag_i$ instances have been added, the excess is distributed among the others *minBag*. Therefore, the labelsets with a lower representation will be benefited from a bigger

Table A3
Undersampling algorithms results – macro-FMeasure.

Algorithm	Dataset	LP-RUS 10	LP-RUS 20	LP-RUS 25	ML-RUS 10	ML-RUS 20	ML-RUS 25
CLR	bibtex	0.3328	0.3238	0.3240	0.3400	0.3409	**0.3448**
HOMER	bibtex	**0.2960**	0.2870	0.2871	0.2920	0.2890	0.2907
IBLR	bibtex	**0.2060**	0.1998	0.1904	0.2050	0.1977	0.1954
RAkEL	bibtex	0.3358	0.3328	0.3356	**0.3384**	0.3371	0.3371
CLR	cal500	**0.3323**	**0.3323**	**0.3323**	0.3128	0.3137	0.3067
HOMER	cal500	0.3194	0.3274	**0.3302**	0.3194	0.3172	0.3254
IBLR	cal500	0.2770	0.2770	0.2770	0.2744	0.2680	**0.2789**
RAkEL	cal500	0.2934	0.2934	0.2934	**0.3028**	0.3013	0.3021
CLR	corel16k	0.1001	0.0990	0.0967	0.1031	0.0968	**0.1054**
HOMER	corel16k	0.1272	0.1222	0.1168	0.1322	0.1374	**0.1380**
IBLR	corel16k	**0.1056**	0.0988	0.0946	0.1049	0.1054	0.0956
RAkEL	corel16k	0.1216	0.1180	0.1176	**0.1244**	0.1197	0.1218
CLR	corel5k	**0.1410**	0.1298	0.1328	0.1304	0.1386	0.1272
HOMER	corel5k	0.1682	0.1660	0.1628	0.1852	0.1840	**0.1856**
IBLR	corel5k	0.0939	0.0909	0.0840	0.1092	0.1069	**0.1104**
RAkEL	corel5k	0.1631	0.1652	0.1552	0.1792	**0.1831**	0.1775
CLR	enron	0.4208	0.4014	0.4184	0.4132	0.4055	**0.4306**
HOMER	enron	0.3702	0.3558	0.3641	**0.3798**	0.3746	0.3604
IBLR	enron	**0.3450**	0.3333	0.3300	0.3399	0.3296	0.3180
RAkEL	enron	0.4062	0.3968	0.4086	0.4039	0.3996	**0.4126**
CLR	genbase	**0.9846**	0.9845	0.9842	0.9675	0.9530	0.9303
HOMER	genbase	0.9796	**0.9836**	0.9775	0.9718	0.9517	0.9331
IBLR	genbase	0.9678	**0.9686**	0.9670	0.9424	0.9122	0.8929
RAkEL	genbase	**0.9890**	**0.9890**	0.9887	0.9834	0.9716	0.9501
CLR	llog	0.2224	0.2037	0.2032	0.2550	**0.2670**	0.2534
HOMER	llog	0.2210	0.2336	**0.2398**	0.2267	0.2166	0.2020
IBLR	llog	0.1738	0.1627	0.1589	**0.1998**	0.1786	0.1798
RAkEL	llog	0.2784	0.2791	**0.2927**	0.2670	0.2652	0.2538
CLR	mediamill	0.2294	**0.2358**	0.2340	0.2176	0.2218	0.2188
HOMER	mediamill	0.2374	**0.2444**	0.2348	0.2290	0.2115	0.2088
IBLR	mediamill	**0.2804**	0.2796	0.2776	0.2634	0.2452	0.2362
RAkEL	mediamill	0.2778	**0.2791**	0.2766	0.2692	0.2509	0.2434
CLR	slashdot	0.3800	0.3530	0.3472	**0.3898**	0.3834	0.3661
HOMER	slashdot	**0.4059**	0.3809	0.3825	0.3766	0.3733	0.3581
IBLR	slashdot	0.2276	0.2199	0.2192	0.2242	0.2218	**0.2530**
RAkEL	slashdot	0.3841	0.3675	0.3605	**0.3982**	0.3842	0.3750
CLR	tmc2007	**0.6091**	0.6089	0.6078	0.5954	0.5781	0.5717
HOMER	tmc2007	**0.5944**	0.5911	0.5861	0.5855	0.5722	0.5662
IBLR	tmc2007	0.4716	**0.4760**	0.4758	0.4406	0.4298	0.4260
RAkEL	tmc2007	**0.6022**	0.5966	0.5942	0.5878	0.5707	0.5652
CLR	yeast	0.4481	0.4538	**0.4562**	0.4483	0.4298	0.4285
HOMER	yeast	**0.4431**	0.4345	0.4274	0.4351	0.4238	0.4164
IBLR	yeast	**0.4990**	0.4828	0.4761	0.4597	0.4671	0.4570
RAkEL	yeast	0.4450	0.4407	0.4436	**0.4474**	0.4367	0.4357

number of clones, aiming to adjust the labelset frequency to a uniform distribution as in LP-RUS.

4.2. Individual label evaluation resampling algorithms

Although LP is easily applicable method and it has shown its effectiveness in many scenarios, it also has severe restrictions. Concerning the evaluation of the imbalance level, LP is limited by the labels sparseness in the MLD. There are MLDs with as many distinct label combinations as instances. This entails that all labelsets would be considered majority and minority cases at the same time, thus LP-ROS and LP-RUS hardly could fix the imbalance problem. For instance, cal500 has 502 instances assigned to 502 different label combinations. Hence, all labelsets in this MLD are unique. Although the *MeanIR* and *CVIR* values for this dataset are quite high, indicating that it is imbalanced, the LP approach would not be able to rebalance it as all the labelsets are equally represented.

An alternative way to accomplish this task, based on the measures proposed in Section 3, would be evaluating the individual imbalance level of each label. The labels whose *IRLbl* is higher than *MeanIR* would be considered as minority labels. This criterion will be used to extract instances to clone or to block from removing. All the other labels, those with *IRLbl* smaller than *MeanIR*, would be treated as majority labels.

Unlike the LP-based resampling proposals, the ML methods described here rely on the measures proposed in Section 3. The ML-RUS and ML-ROS algorithms are based on the technique just described above. These two algorithms are also aimed to resample MLDs. However, the instances to delete or clone are selected using an individual evaluation of each label, instead of full labelsets. The *MeanIR* and *IRLbl* measures are their cornerstones.

4.2.1. ML-ROS: individual label random oversampling

ML-ROS (see Algorithm 2) uses the *IRLbl* measure to obtain bags of instances in which minority labels (whose *IRLbl* is higher than *MeanIR*) appear. The instances to clone are randomly picked from those bags. The method updates the *IRLbl* in each loop cycle, and excludes from the processing any minority label which reaches *MeanIR*. It should be considered that the cloned instances may also contain non-minority labels, and as consequence the process could increase their frequency in the MLD.

Algorithm 2. ML-ROS algorithm's pseudo-code.

Inputs: ⟨Dataset⟩ *D*, ⟨Percentage⟩ *P*
Outputs: Preprocessed dataset
1: *samplesToClone* ← |*D*| /100∗*P* ▷ *P%* size increment
2: *L* ← labelsInDataset(*D*) ▷ Obtain the full set of labels

Table A4
Oversampling algorithms results – accuracy.

Algorithm	Dataset	LP-ROS 10	LP-ROS 20	LP-ROS 25	ML-ROS 10	ML-ROS 20	ML-ROS 25
CLR	bibtex	0.1670	0.1688	0.1724	0.2364	0.2367	**0.2388**
HOMER	bibtex	0.1515	0.1542	0.1504	**0.2677**	0.2614	0.2548
IBLR	bibtex	0.1007	0.1014	0.1038	0.1768	0.1767	**0.1783**
RAkEL	bibtex	0.2072	0.2064	0.2066	**0.2925**	0.2882	0.2888
CLR	cal500	**0.2150**	0.2147	0.2110	0.2038	0.2138	0.2140
HOMER	cal500	0.2040	0.2062	0.2030	**0.2210**	0.2116	0.2128
IBLR	cal500	0.1940	0.1896	0.1896	0.1900	**0.1941**	0.1940
RAkEL	cal500	0.2060	0.2036	0.2047	**0.2121**	0.2102	0.2109
CLR	corel16k	0.0555	0.0576	**0.0584**	0.0480	0.0508	0.0500
HOMER	corel16k	0.0764	0.0759	0.0760	**0.1107**	0.1039	0.1026
IBLR	corel16k	0.0454	0.0472	**0.0475**	0.0292	0.0324	0.0344
RAkEL	corel16k	0.0590	0.0594	0.0603	**0.0700**	0.0698	0.0690
CLR	corel5k	0.0417	0.0426	**0.0444**	0.0390	0.0416	0.0429
HOMER	corel5k	0.0696	0.0700	0.0727	**0.0996**	0.0958	0.0946
IBLR	corel5k	0.0347	0.0361	**0.0368**	0.0327	0.0340	0.0351
RAkEL	corel5k	0.0621	0.0608	0.0608	0.0612	0.0634	**0.0650**
CLR	enron	0.3167	0.3160	0.3170	**0.4068**	0.4019	0.4030
HOMER	enron	0.2665	0.2653	0.2598	**0.4024**	0.3840	0.3926
IBLR	enron	0.2530	0.2457	0.2466	0.3155	**0.3182**	0.3147
RAkEL	enron	0.2802	0.2797	0.2783	**0.3890**	0.3838	0.3808
CLR	genbase	0.9755	0.9770	0.9764	0.9842	0.9844	**0.9849**
HOMER	genbase	0.9776	0.9792	0.9800	0.9834	**0.9849**	0.9820
IBLR	genbase	0.9798	0.9809	0.9809	**0.9842**	0.9836	0.9841
RAkEL	genbase	0.9820	0.9842	0.9844	0.9864	0.9866	**0.9871**
CLR	llog	0.0272	0.0258	0.0246	**0.0470**	0.0437	0.0443
HOMER	llog	0.0642	0.0672	0.0672	**0.1105**	0.1031	0.0961
IBLR	llog	0.0497	**0.0533**	0.0524	0.0357	0.0367	0.0353
RAkEL	llog	0.0922	0.0950	0.0940	**0.1324**	0.1286	0.1315
CLR	mediamill	0.3876	0.3884	0.3881	**0.4559**	0.4558	0.4556
HOMER	mediamill	0.2749	0.2770	0.2776	**0.4002**	0.3946	0.3925
IBLR	mediamill	0.3205	0.3210	0.3214	**0.4644**	0.4633	0.4624
RAkEL	mediamill	0.2808	0.2816	0.2816	**0.4114**	0.4062	0.4046
CLR	slashdot	0.1993	0.2212	0.2298	0.3260	0.3258	**0.3302**
HOMER	slashdot	0.2610	0.2713	0.2585	**0.3550**	0.3432	0.3488
IBLR	slashdot	0.1646	0.1700	**0.1723**	0.1343	0.1384	0.1392
RAkEL	slashdot	0.2116	0.2370	0.2458	0.3496	0.3518	**0.3578**
CLR	tmc2007	0.4845	0.4858	0.4850	0.6148	**0.6164**	0.6152
HOMER	tmc2007	0.4020	0.4056	0.4073	0.6012	0.6012	**0.6016**
IBLR	tmc2007	0.3481	0.3483	0.3492	**0.5281**	0.5232	0.5232
RAkEL	tmc2007	0.4170	0.4180	0.4182	0.6022	**0.6023**	0.6008
CLR	yeast	0.3928	0.3894	0.3866	**0.4614**	0.4567	0.4572
HOMER	yeast	0.3316	0.3293	0.3290	**0.4053**	0.3979	0.3982
IBLR	yeast	0.3910	0.3936	0.3946	**0.5142**	0.5048	0.4998
RAkEL	yeast	0.3422	0.3380	0.3394	**0.4101**	0.4022	0.4063

```
3:    MeanIR ← calculateMeanIR(D, L)
4:    for each label in L do ▷ Bags of minority labels samples
5:      IRLbl_label ← calculateIRperLabel(D, label)
6:      if IRLbl_label > MeanIR then
7:        minBag_{i++} ← Bag_label
8:      end if
9:    end for
10:   while samplesToClone > 0 do ▷ Instances cloning loop
11:     ▷ Clone a random sample from each minority bag
12:     for each minBag_i in minBag do
13:       x ← random(1, |minBag_i|)
14:       cloneSample(minBag_i, x)
15:       if IRLbl_minBag_i <= MeanIR then
16:         minBag → minBag_i ▷ Exclude from cloning
17:       end if
18:       - -samplesToClone
19:     end for
20:   end while
```

4.2.2. ML-RUS: individual label random undersampling

ML-RUS uses the minority bags to prevent the deletion of samples that belong to minority labels. The instances contained in these bags are excluded from the removing process, which works randomly choosing from the remaining samples. These only can contain majority labels, whose *IRLbl* is lower than *MeanIR*. As ML-ROS, ML-RUS reassess the *IRLbl* of the labels affected by the operation, banning those that reach the *MeanIR* value.

5. Experimentation and analysis

The performance of the four proposed resampling methods has been tested using several MLDs and MLC algorithms. The following experimental framework is described and the results obtained are analyzed.

5.1. Experimental framework

The proposed resampling methods were tested using the MLDs shown in Table 1. This table shows some basic characterization measures: number of attributes, samples, and labels, and the average number of labels per sample, along with the proposed measures related to the imbalance level. Emotions and scene datasets were not used, as their *MeanIR* < 1.5 denote them as well-balanced MLDs. As can be seen, there are datasets with a variety of values in *Card/Dens*, as well as some big differences in the number of labels, attributes, samples, and the imbalance

Table A5
Oversampling algorithms results – micro-FMeasure.

Algorithm	Dataset	LP-ROS 10	LP-ROS 20	LP-ROS 25	ML-ROS 10	ML-ROS 20	ML-ROS 25
CLR	bibtex	0.6529	0.6514	0.6535	**0.7690**	0.7555	0.7542
HOMER	bibtex	0.2084	0.2104	0.2098	**0.3656**	0.3532	0.3520
IBLR	bibtex	0.1661	0.1698	0.1728	0.4070	0.4053	**0.4074**
RAkEL	bibtex	0.2999	0.2961	0.2946	**0.4756**	0.4567	0.4444
CLR	cal500	0.5526	0.5569	0.5526	**0.5911**	0.5729	0.5669
HOMER	cal500	0.3385	0.3404	0.3358	**0.3512**	0.3451	0.3460
IBLR	cal500	0.2790	0.2737	0.2724	0.2802	0.2822	**0.2828**
RAkEL	cal500	0.3387	0.3356	0.3376	**0.3709**	0.3510	0.3520
CLR	corel16k	0.3140	0.3209	0.3209	**0.4232**	0.4168	0.4070
HOMER	corel16k	0.1339	0.1348	0.1337	**0.2128**	0.2032	0.2002
IBLR	corel16k	0.1156	0.1204	0.1208	**0.2718**	0.2456	0.2366
RAkEL	corel16k	0.1352	0.1336	0.1335	**0.2998**	0.2602	0.2446
CLR	corel5k	0.3489	0.3484	0.3527	**0.4402**	0.4326	0.4290
HOMER	corel5k	0.1335	0.1310	0.1334	**0.2040**	0.1966	0.1915
IBLR	corel5k	0.0500	0.0523	0.0530	0.0512	0.0570	**0.0604**
RAkEL	corel5k	0.1980	0.1861	0.1839	**0.3113**	0.2932	0.2888
CLR	enron	0.5929	0.5952	0.5929	**0.6772**	0.6752	0.6762
HOMER	enron	0.3840	0.3852	0.3858	**0.5237**	0.5108	0.5074
IBLR	enron	0.4513	0.4419	0.4438	0.5934	**0.5975**	0.5914
RAkEL	enron	0.4336	0.4297	0.4254	**0.5924**	0.5845	0.5706
CLR	genbase	0.9850	0.9840	0.9846	0.9868	**0.9874**	0.9868
HOMER	genbase	0.9794	0.9845	0.9869	0.9904	**0.9916**	0.9821
IBLR	genbase	0.9814	0.9836	0.9843	0.9863	**0.9868**	0.9862
RAkEL	genbase	0.9880	0.9881	0.9887	0.9898	**0.9904**	0.9898
CLR	llog	0.3919	0.4208	0.4098	0.5974	0.5949	**0.6476**
HOMER	llog	0.0955	0.0898	0.0941	**0.1645**	0.1512	0.1520
IBLR	llog	0.0560	0.0604	0.0606	0.0688	**0.0757**	0.0752
RAkEL	llog	0.1220	0.1219	0.1203	**0.2525**	0.2429	0.2493
CLR	mediamill	0.6112	0.6141	0.6138	**0.7650**	0.7608	0.7600
HOMER	mediamill	0.3611	0.3629	0.3646	**0.5516**	0.5379	0.5354
IBLR	mediamill	0.4084	0.4068	0.4064	**0.7386**	0.7176	0.7109
RAkEL	mediamill	0.3663	0.3677	0.3681	**0.6024**	0.5863	0.5817
CLR	slashdot	0.5188	0.5272	0.5272	**0.6537**	0.6448	0.6528
HOMER	slashdot	0.3381	0.3346	0.3456	0.5554	**0.5934**	0.5775
IBLR	slashdot	0.2881	0.2911	0.2934	**0.6385**	0.5937	0.5712
RAkEL	slashdot	0.4196	0.4308	0.4281	**0.6848**	0.6786	0.6786
CLR	tmc2007	0.5973	0.6016	0.6032	**0.7530**	0.7515	0.7512
HOMER	tmc2007	0.4740	0.4752	0.4774	**0.6941**	0.6912	0.6891
IBLR	tmc2007	0.4519	0.4522	0.4529	**0.7135**	0.7049	0.7038
RAkEL	tmc2007	0.4917	0.4958	0.4970	**0.7283**	0.7214	0.7192
CLR	yeast	0.5389	0.5388	0.5356	**0.6359**	0.6338	0.6332
HOMER	yeast	0.4604	0.4604	0.4546	**0.5475**	0.5440	0.5445
IBLR	yeast	0.4976	0.5000	0.5015	**0.7039**	0.6843	0.6767
RAkEL	yeast	0.4748	0.4715	0.4733	**0.5639**	0.5578	0.5606

measures. The goal is to analyze how the proposed resampling methods work with MLDs that are not similar, but quite different. A 2×5 folds cross validation scheme was used, and the training partitions have been preprocessed with the four proposed methods using three different *Percentage* values: 10, 20, and 25.

Regarding the MLC algorithms, the following were chosen: CLR [37], RAkEL [34], IBLR [38], and HOMER [35]. The C4.5 tree-based classification algorithm was used as underlying classifier where a binary/multiclass classification algorithm was needed. The number of clusters for HOMER was set to the minimum between 4 and the number of labels in the MLD. Default values were used for the rest of parameters. Each MLC algorithm was run over the base datasets, without any preprocessing, as well as using the datasets once they have been preprocessed with LP-RUS, LP-ROS, ML-RUS, and ML-ROS.

The performance of a multilabel classifier can be evaluated using a large range of measures. As stated in [39], it is important to use several of them to assess predictive performance. These measures are grouped into three categories [1]: sample-based measures, label-based measures, and ranking-based measures. To assess the influence of each separated label in the obtained results the measures to use are label-based. These measures reflect the correct classification of majority or minority labels better than others, and therefore are a way to follow when the interest is in evaluating how a resampling algorithm has changed the predictions made by a classifier.

The label-based measures are grouped into two categories, called *macro*-measures and *micro*-measures. For each group there are several measures, *precision*, *recall* and *F-measure* (also known as F_1-*score* or simply F_1) among them. A *macro*-measure is computed as shown in Eq. (6), evaluating the underlying measure separately, once for each label, and eventually calculating the mean as result. As stated in [40], this approach is heavily affected by the results of "rare categories" (minority labels), the labels whose classification results the resampling applied aims to improve. In contrast, a *micro*-measure begins by aggregating the predictions of all labels, and evaluates the measure at the end, as can be seen in Eq. (7). Therefore, the bad or good results in classification of minority labels are diluted among the much more abundant predictions from majority labels, being a type of measure more appropriate to estimate the global performance of the classifier. In both equations, TP stands for *True Positives*, FP for *False Positives*, TN for *True Negatives*, and FN for *False Negatives*:

$$MacroM = \frac{1}{L}\sum_{i=1}^{L} evalM(TP_i, FP_i, TN_i, FN_i) \tag{6}$$

$$MicroM = evalM\left(\sum_{i=1}^{L} TP_i, \sum_{i=1}^{L} FP_i, \sum_{i=1}^{L} TN_i, \sum_{i=1}^{L} FN_i\right) \tag{7}$$

Table A6
Oversampling algorithms results – macro-FMeasure.

Algorithm	Dataset	LP-ROS 10	LP-ROS 20	LP-ROS 25	ML-ROS 10	ML-ROS 20	ML-ROS 25
CLR	bibtex	0.2927	0.2923	0.2974	**0.3386**	0.3358	0.3379
HOMER	bibtex	0.2180	0.2212	0.2210	**0.2970**	0.2945	0.2863
IBLR	bibtex	0.1533	0.1558	0.1550	**0.2200**	0.2176	0.2184
RAkEL	bibtex	0.2754	0.2739	0.2755	**0.3288**	0.3229	0.3236
CLR	cal500	0.3236	0.3238	0.3199	0.3202	**0.3318**	0.3221
HOMER	cal500	0.2888	0.2900	0.2841	**0.3019**	0.2967	0.2958
IBLR	cal500	**0.2756**	0.2705	0.2742	0.2700	0.2750	0.2755
RAkEL	cal500	0.2915	0.2906	0.2924	0.2966	0.2971	**0.2972**
CLR	corel16k	0.1059	0.1046	0.1075	0.1033	0.1076	**0.1080**
HOMER	corel16k	0.0861	0.0893	0.0910	**0.1363**	0.1292	0.1273
IBLR	corel16k	0.0778	0.0805	0.0804	**0.1094**	0.1055	0.1085
RAkEL	corel16k	0.0835	0.0832	0.0822	**0.1278**	0.1176	0.1192
CLR	corel5k	0.1366	0.1374	0.1395	0.1355	0.1384	**0.1431**
HOMER	corel5k	0.1418	0.1398	0.1451	**0.1896**	0.1884	0.1877
IBLR	corel5k	0.0987	0.1012	0.1008	0.1157	0.1192	**0.1235**
RAkEL	corel5k	0.1555	0.1552	0.1521	0.1784	**0.1851**	0.1850
CLR	enron	0.3819	0.3760	0.3792	**0.4220**	0.3968	0.4082
HOMER	enron	0.2957	0.3117	0.2988	**0.3740**	0.3597	0.3575
IBLR	enron	0.3088	0.3046	0.3092	**0.3580**	0.3447	0.3485
RAkEL	enron	0.3188	0.3252	0.3177	**0.3930**	0.3856	0.3880
CLR	genbase	0.9732	0.9754	0.9755	0.9800	0.9802	**0.9804**
HOMER	genbase	0.9746	0.9746	0.9754	0.9814	**0.9877**	0.9778
IBLR	genbase	0.9730	0.9737	0.9734	**0.9799**	0.9756	0.9787
RAkEL	genbase	0.9834	0.9876	0.9878	0.9890	0.9893	**0.9894**
CLR	llog	0.2241	0.1763	0.1823	**0.2508**	0.2308	0.2418
HOMER	llog	0.2020	0.2040	0.1942	0.2495	**0.2510**	0.2494
IBLR	llog	0.1662	0.1694	0.1730	0.2096	**0.2240**	0.2124
RAkEL	llog	0.2316	0.2318	0.2372	0.2921	0.2744	**0.3018**
CLR	mediamill	0.2020	0.2037	0.2073	0.2322	0.2318	**0.2336**
HOMER	mediamill	0.1565	0.1572	0.1582	**0.2422**	0.2350	0.2348
IBLR	mediamill	0.2086	0.2091	0.2106	0.2800	0.2813	**0.2834**
RAkEL	mediamill	0.1705	0.1721	0.1685	**0.2618**	0.2579	0.2556
CLR	slashdot	0.3410	0.3463	0.3602	0.4061	0.4069	**0.4203**
HOMER	slashdot	0.3298	0.3272	0.3288	0.3907	**0.3978**	0.3941
IBLR	slashdot	0.2256	0.2274	0.2284	0.2319	**0.2328**	0.2112
RAkEL	slashdot	0.3280	0.3407	0.3516	0.4002	0.4024	**0.4137**
CLR	tmc2007	0.5731	0.5782	0.5777	0.6332	**0.6440**	0.6430
HOMER	tmc2007	0.4673	0.4692	0.4727	0.6068	0.6082	**0.6114**
IBLR	tmc2007	0.3880	0.3891	0.3886	0.4740	0.4765	**0.4844**
RAkEL	tmc2007	0.4870	0.4895	0.4916	0.6138	**0.6180**	0.6162
CLR	yeast	0.4206	0.4260	0.4159	0.4537	0.4464	**0.4590**
HOMER	yeast	0.3985	0.3925	0.3963	**0.4314**	0.4134	0.4182
IBLR	yeast	0.4433	0.4460	0.4467	0.4566	0.4622	**0.4704**
RAkEL	yeast	0.4233	0.4166	0.4150	**0.4528**	0.4512	0.4507

Because *F*-measure (8) is in fact the harmonic mean of precision (9) and recall (10), it gives a way to do a weighted evaluation of these two factors using the obtained results. Macro-FMeasure will be used to determine the changes produced in those results placing more emphasis in minority labels, while Micro-FMeasure will better take in account the influence of majority labels. In these equations Y_i is the set of real labels associated with the instance x_i, whereas $h(x_i)$ would be the set of labels predicted by the multi-label classifier:

$$F\text{-}measure = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{8}$$

$$Precision = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \cap h(x_i)|}{|h(x_i)|} = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \cap h(x_i)|}{|Y_i|} = \frac{TP}{TP + FN} \tag{10}$$

In addition, the common Accuracy measure, Eq. (11), will also be used to obtain a general view of the methods' performance. Accuracy is a measure that assess the positive and negative predictive performance of the classifier:

$$Accuracy = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \cap h(x_i)|}{|Y_i \cup h(x_i)|} \tag{11}$$

5.2. Results and analysis

The experimentation has been done in three stages. The first two allowed us to choose the best undersampling configuration and the best oversampling configuration. The final step compares these two best configurations with the results obtained without resampling.

In each phase, the statistical analysis of results was performed in two steps, as recommended in [41,42]. In the first step, the Friedman test is used to rank the methods, and to establish if any statistical differences exist. The second step performs a multiple comparison using Holm and Shaffer *post hoc* procedures, able to elucidate pair-wise differences among the algorithms. The full statistical study is provided as supplementary material.[2]

[2] Dataset partitions, tables of results and full statistical analysis are available to download at http://simidat.ujaen.es/NeucomIDMLC.

Table A7
Base vs best methods – accuracy.

Algorithm	Dataset	Base	ML-RUS 10	ML-ROS 10
CLR	bibtex	0.2316	0.2292	**0.2364**
HOMER	bibtex	**0.2714**	0.2618	0.2677
IBLR	bibtex	0.1746	0.1684	**0.1768**
RAkEL	bibtex	**0.3002**	0.2966	0.2925
RAkEL	cal500	**0.2135**	0.1708	0.2038
HOMER	cal500	**0.2496**	0.2346	0.2210
CLR	cal500	0.1787	0.1898	**0.1900**
IBLR	cal500	0.1922	0.2101	**0.2121**
CLR	corel16k	0.0456	0.0453	**0.0480**
HOMER	corel16k	**0.1138**	0.1118	0.1107
IBLR	corel16k	0.0253	0.0256	**0.0292**
RAkEL	corel16k	0.0645	0.0633	**0.0700**
RAkEL	corel5k	**0.0586**	0.0355	0.0390
HOMER	corel5k	**0.1029**	0.1016	0.0996
CLR	corel5k	**0.0360**	0.0296	0.0327
IBLR	corel5k	0.0315	0.0589	**0.0612**
RAkEL	enron	0.4010	**0.4184**	0.4068
HOMER	enron	**0.4110**	0.4085	0.4024
CLR	enron	**0.4171**	0.3005	0.3155
IBLR	enron	0.3226	**0.4034**	0.3890
RAkEL	genbase	**0.9842**	0.9716	**0.9842**
HOMER	genbase	0.9792	0.9764	**0.9834**
CLR	genbase	0.9837	0.9671	**0.9842**
IBLR	genbase	0.9790	0.9782	**0.9864**
CLR	llog	0.0456	0.0458	**0.0470**
HOMER	llog	0.1053	0.1038	**0.1105**
IBLR	llog	0.0322	0.0352	**0.0357**
RAkEL	llog	**0.1419**	0.1325	0.1324
CLR	mediamill	0.4490	0.4438	**0.4559**
HOMER	mediamill	0.4088	**0.4089**	0.4002
IBLR	mediamill	**0.4660**	0.4590	0.4644
RAkEL	mediamill	**0.4194**	0.4144	0.4114
CLR	slashdot	0.3236	0.3194	**0.3260**
HOMER	slashdot	0.3534	0.3314	**0.3550**
IBLR	slashdot	0.1269	**0.1486**	0.1343
RAkEL	slashdot	0.3452	0.3392	**0.3496**
CLR	tmc2007	0.6132	0.6020	**0.6148**
HOMER	tmc2007	**0.6029**	0.5897	0.6012
IBLR	tmc2007	**0.5322**	0.5184	0.5281
RAkEL	tmc2007	**0.6044**	0.5913	0.6022
RAkEL	yeast	0.4338	**0.4706**	0.4614
HOMER	yeast	0.4292	**0.4312**	0.4053
CLR	yeast	0.4698	**0.5150**	0.5142
IBLR	yeast	**0.5210**	0.4344	0.4101

Table A8
Base vs best methods – micro-FM.

Algorithm	Dataset	Base	ML-RUS 10	ML-ROS 10
CLR	bibtex	0.3371	**0.7793**	0.7690
HOMER	bibtex	0.3568	0.3648	**0.3656**
IBLR	bibtex	0.2628	0.3494	**0.4070**
RAkEL	bibtex	0.4021	**0.5136**	0.4756
RAkEL	cal500	0.3488	**0.6258**	0.5911
HOMER	cal500	**0.3978**	0.3732	0.3512
CLR	cal500	**0.2977**	0.2777	0.2802
IBLR	cal500	0.3184	**0.4188**	0.3709
CLR	corel16k	0.0846	**0.4300**	0.4232
HOMER	corel16k	0.1866	**0.2327**	0.2128
IBLR	corel16k	0.0504	**0.3049**	0.2718
RAkEL	corel16k	0.1145	**0.3511**	0.2998
RAkEL	corel5k	0.1096	**0.4512**	0.4402
HOMER	corel5k	0.1744	**0.2086**	0.2040
CLR	corel5k	**0.0706**	0.0434	0.0512
IBLR	corel5k	0.0542	**0.3707**	0.3113
RAkEL	enron	0.5334	**0.6780**	0.6772
HOMER	enron	0.5265	**0.5482**	0.5237
CLR	enron	0.5596	0.5588	**0.5934**
IBLR	enron	0.4561	**0.6195**	0.5924
RAkEL	genbase	0.9867	0.9844	**0.9868**
HOMER	genbase	0.9820	0.9852	**0.9904**
CLR	genbase	0.9852	0.9478	**0.9863**
IBLR	genbase	0.9768	0.9875	**0.9898**
CLR	llog	0.0734	0.5863	**0.5974**
HOMER	llog	0.1491	0.1594	**0.1645**
IBLR	llog	0.0560	0.0580	**0.0688**
RAkEL	llog	0.2062	**0.2880**	0.2525
CLR	mediamill	0.5928	**0.7750**	0.7650
HOMER	mediamill	0.5493	**0.5882**	0.5516
IBLR	mediamill	0.5987	**0.7701**	0.7386
RAkEL	mediamill	0.5622	**0.6510**	0.6024
CLR	slashdot	0.4416	0.6315	**0.6537**
HOMER	slashdot	0.4429	**0.5997**	0.5554
IBLR	slashdot	0.2042	**0.6505**	0.6385
RAkEL	slashdot	0.4598	0.6675	**0.6848**
CLR	tmc2007	0.7228	0.7500	**0.7530**
HOMER	tmc2007	**0.6982**	0.6908	0.6941
IBLR	tmc2007	0.6447	**0.7221**	0.7135
RAkEL	tmc2007	0.7063	**0.7337**	0.7283
RAkEL	yeast	0.5796	**0.6516**	0.6359
HOMER	yeast	**0.5763**	0.5680	0.5475
CLR	yeast	0.6168	**0.7137**	0.7039
IBLR	yeast	**0.6502**	0.5913	0.5639

5.2.1. Best undersampling method selection

Tables A1, A2 and A3 in Appendix A show classification results for undersampling methods, LP-RUS and ML-RUS, respectively. Best values for each classifier-dataset combination are highlighted in bold. The results obtained from the statistical analysis are summarized in Table 2. There is a column for each evaluation measure, showing the average ranking for the algorithms. The meaning of the symbols to the right of the values is the following:

- $\star$: A star denotes that Friedman test has rejected the null hypothesis. Therefore, statistically significant differences exist among some of the algorithms. The one to the left of the star has the best ranking.
- $\leftrightarrow$: There is not statistically significant difference between this method and the best one.
- $\downarrow$: This method is statistically worse than the best one, with $p{-}value < 0.1$.
- $\downarrow\downarrow$: This method is statistically worse than the best one, with $p{-}value < 0.05$.

Table 2 shows that ML-RUS 10 is the winner for Accuracy and Micro-FMeasure, whereas for Macro-FMeasure appears as second but without statistical difference from the best one (LP-RUS 10). Thus, it can be concluded that ML-RUS 10 is on average the best undersampling method.

5.2.2. Best oversampling method selection

Tables A4, A5 and A6 correspond to classification results for oversampling methods. From Table 3, which shows the statistical analysis, that a clear statistical difference between ML-ROS and LP-ROS methods exists can be seen. For Accuracy and Micro-FMeasure the winner is ML-ROS 10, whereas for Macro-FMeasure it is ML-ROS 25. In the last case, there is not statistical difference with ML-ROS 10. On the contrary, for Micro-FMeasure ML-ROS 10 appears as statistically better than ML-ROS 25. Thus, that ML-ROS 10 is the best oversampling method on average can be concluded.

After finishing the two first experimental steps, it is possible to deduce that ML-ROS/ML-RUS are overall better than their counterparts LP-ROS/LP-RUS, even though not always a statistical difference between them exists. Therefore, the resampling based on the evaluation of imbalance by individual labels seems superior to the LP based approach.

5.2.3. Base MLC algorithms vs best configurations

The third experimentation phase compares ML-RUS 10 and ML-ROS 10 with classification without preprocessing results, denoted

Table A9
Base vs best methods – macro-FM.

Algorithm	Dataset	Base	ML-RUS 10	ML-ROS 10
CLR	bibtex	0.3342	**0.3400**	0.3386
HOMER	bibtex	**0.3042**	0.2920	0.2970
IBLR	bibtex	0.2140	0.2050	**0.2200**
RAkEL	bibtex	0.3368	**0.3384**	0.3288
RAkEL	cal500	0.2934	0.3128	**0.3202**
HOMER	cal500	**0.3316**	0.3194	0.3019
CLR	cal500	**0.3323**	0.2744	0.2700
IBLR	cal500	0.2772	**0.3028**	0.2966
CLR	corel16k	0.1003	0.1031	**0.1033**
HOMER	corel16k	**0.1363**	0.1322	**0.1363**
IBLR	corel16k	**0.1141**	0.1049	0.1094
RAkEL	corel16k	0.1277	0.1244	**0.1278**
RAkEL	corel5k	**0.1774**	0.1304	0.1355
HOMER	corel5k	**0.1916**	0.1852	0.1896
CLR	corel5k	**0.1330**	0.1092	0.1157
IBLR	corel5k	0.1059	**0.1792**	0.1784
RAkEL	enron	0.4029	0.4132	**0.4220**
HOMER	enron	0.3790	**0.3798**	0.3740
CLR	enron	**0.4198**	0.3399	0.3580
IBLR	enron	0.3458	**0.4039**	0.3930
RAkEL	genbase	**0.9890**	0.9675	0.9800
HOMER	genbase	0.9780	0.9718	**0.9814**
CLR	genbase	**0.9848**	0.9424	0.9799
IBLR	genbase	0.9655	0.9834	**0.9890**
CLR	llog	0.2330	**0.2550**	0.2508
HOMER	llog	0.2380	0.2267	**0.2495**
IBLR	llog	0.1830	0.1998	**0.2096**
RAkEL	llog	0.2824	0.2670	**0.2921**
CLR	mediamill	0.2276	0.2176	**0.2322**
HOMER	mediamill	0.2404	0.2290	**0.2422**
IBLR	mediamill	**0.2806**	0.2634	0.2800
RAkEL	mediamill	**0.2774**	0.2692	0.2618
CLR	slashdot	0.3982	0.3898	**0.4061**
HOMER	slashdot	**0.3996**	0.3766	0.3907
IBLR	slashdot	**0.2382**	0.2242	0.2319
RAkEL	slashdot	**0.4038**	0.3982	0.4002
CLR	tmc2007	0.6073	0.5954	**0.6332**
HOMER	tmc2007	0.5968	0.5855	**0.6068**
IBLR	tmc2007	0.4668	0.4406	**0.4740**
RAkEL	tmc2007	0.6015	0.5878	**0.6138**
RAkEL	yeast	0.4466	0.4483	**0.4537**
HOMER	yeast	0.4334	**0.4351**	0.4314
CLR	yeast	0.4480	**0.4597**	0.4566
IBLR	yeast	**0.4944**	0.4474	0.4528

as *Base* in all tables. Tables A7, A8 and A9 contain this final stage results. The output from the statistical tests (Table 4) indicates that ML-ROS 10 performs statistically better than ML-RUS 10 in Accuracy and Macro-FMeasure. Although it is also better than Base in these two measures, the differences have not statistical significance. For Micro-FMeasure both ML-RUS 10 and ML-ROS 10 are statistically better than Base, but there are not meaningful differences between them.

Overall, ML-ROS is noticeably better than ML-RUS. Even though for Micro-FMeasure ML-RUS obtains the best rank, the difference with ML-ROS is not significant. On the other hand, for Accuracy and Macro-FMeasure the performance of ML-ROS is significantly better than ML-RUS from a statistical point of view. Theoretically, removing majority instances should produce a similar effect that adding new minority ones. The two actions tend to balance label representation in the MLD. However, multilabel instances are representatives of a set of labels, not only one class as in traditional classification. Removing an instance has a side effect over a potential large number of labels. Thus, the loss of information caused by ML-RUS comes in detriment of the results when compared with ML-ROS.

From this exploratory experimentation on how the classical resampling techniques could be adapted to work with MLDs, it is possible to infer the following consequences:

- Although only the most basic resampling methods based on random removing and cloning of samples have been applied, a comprehensive improvement of the results is obtained, sometimes even with statistical significant differences.
- The best undersampling method performs significantly worse than the best oversampling in two of the three evaluation measures. This result is consistent with published studies such as [43] and with the nature of MLDs, as the deletion of one instance does not influence only the evaluated label, but all the other labels which appear in this sample.
- Focusing in the oversampling techniques, the ML approach is clearly superior to the LP one (see Table 3). The use of full labelsets to assess the imbalance incurs the risk of increasing only the number of instances with majority labels, as they can generate many different combinations. The minority labels could appear usually together, becoming more frequent labelsets than those associated with the majority ones. The individual imbalance evaluation in the ML approach, extracting minority bags, guarantees that all the cloned samples include some minority label, although they can also include majority ones.

6. Conclusion

The learning from imbalanced datasets problem has been deeply studied in recent years in the domain of traditional classification. In this paper, a new group of measures aimed to evaluate the imbalance level in MLDs, along with four resampling algorithms, have been proposed, and the experimentation made to validate them has been described. LP-RUS is a random undersampling algorithm, whereas LP-ROS does random oversampling, in both cases taking as class value the labelset assigned to each data instance. ML-RUS and ML-ROS are also undersampling and oversampling methods, but work with an individual imbalance evaluation per label, instead of using full labelsets.

The proposed measures can be used to assess the imbalance level, and being able to decide if a certain MLD could be benefited from the proposed resampling methods. Using these measures, we stated that emotions and scene should not be preprocessed as they do not suffer from imbalance. These measures are also the foundation of the ML-RUS and ML-ROS algorithms, which use them to decide the instances that will be cloned/removed.

Among the resampling algorithms proposed, ML-ROS with a 10% of oversampling obtains the best overall results considering different quality measures. The multilabel oversampling accomplished by ML-ROS is able to improve classification results when it is applied to imbalanced MLDs, whatever MLC algorithm is used.

Most of all, the probationary implementation of resampling algorithms and the conducted experimentation have proved that resampling techniques can be an alternative to the published proposals when it comes to work with imbalanced MLDs, opening a new path to face this problem. As in traditional classification, a further step would be the study of new algorithms able to generate artificial multilabel samples.

Acknowledgments

F. Charte is supported by the Spanish Ministry of Education under the FPU National Program (Ref. AP2010-0068). This paper is partially supported by the projects TIN2012-33856 and TIN2011-28488 of the Spanish Ministry of Science and Technology, and the projects P10-TIC-6858 and P11-TIC-7765 of the Andalusian Research Plan.

Appendix A. Tables of results

See Tables A1–Tables A9.

References

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

[2] M.-L. Zhang, Multilabel neural networks with applications to functional genomics and text categorization, IEEE Trans. Knowl. Data Eng. 18 (10) (2006) 1338–1351. http://dx.doi.org/10.1109/TKDE.2006.162.

[3] A. Wieczorkowska, P. Synak, Z. Raś, Multi-label classification of emotions in music, in: Intelligent Information Processing and Web Mining, AISC, vol. 35, 2006, pp. 307–315 (Chapter 30). http://dx.doi.org/10.1007/3-540-33521-8_30.

[4] N. Japkowicz, S. Stephen, The class imbalance problem: a systematic study, Intell. Data Anal. 6 (5) (2002) 429–449.

[5] 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. http://dx.doi.org/10.1016/j.knosys.2013.01.018.

[6] 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. http://dx.doi.org/10.1016/j.ins.2013.07.007.

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

[8] 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. http://dx.doi.org/10.1016/j.patcog.2012.03.014.

[9] 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. http://dx.doi.org/10.1371/journal.pone.0037155.

[10] S. Diplaris, G. Tsoumakas, P. Mitkas, I. Vlahavas, Protein classification with multiple algorithms, in: Proceedings of the 10th Panhellenic Conference on Informatics, Volos, Greece, PCI'05, 2005, pp. 448–456. http://dx.doi.org/10.1007/11573036_42.

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

[12] E. Hüllermeier, J. Fürnkranz, W. Cheng, K. Brinker, Label ranking by learning pairwise preferences, Artif. Intell. 172 (16) (2008) 1897–1916. http://dx.doi.org/10.1016/j.artint.2008.08.002.

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

[14] 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, Freiburg, Germany, PKDD'01, vol. 2168, 2001, pp. 42–53. http://dx.doi.org/10.1007/3-540-44794-6_4.

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

[16] M.-L. Zhang, Ml-rbf: RBF neural networks for multi-label learning, Neural Process. Lett. 29 (2009) 61–74. http://dx.doi.org/10.1007/s11063-009-9095-3.

[17] A. Elisseeff, J. Weston, A kernel method for multi-labelled classification, in: Advances in Neural Information Processing Systems, vol. 14, MIT Press, Cambridge, MA, USA, 2001, pp. 681–687.

[18] M.L. Zhang, Z.H. Zhou, A review on multi-label learning algorithms, IEEE Trans. Knowl. Data Eng. 8 (2014) 1819–1837. http://dx.doi.org/10.1109/TKDE.2013.39.

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

[20] B. Krawczyk, M. Woźniak, G. Schaefer, Cost-sensitive decision tree ensembles for effective imbalanced classification, Appl. Soft Comput. 14 (2014) 554–562. http://dx.doi.org/10.1016/j.asoc.2013.08.014.

[21] S.B. Kotsiantis, P.E. Pintelas, Mixture of expert agents for handling imbalanced data sets, Ann. Math. Comput. Teleinform. 1 (2003) 46–55.

[22] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, Smote: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357. http://dx.doi.org/10.1613/jair.953.

[23] H. He, Y. Ma, Imbalanced Learning: Foundations, Algorithms, and Applications, Wiley-IEEE Press, Hoboken, NJ, USA, 2013.

[24] G. Tsoumakas, E.S. Xioufis, J. Vilcek, I. Vlahavas, MULAN Multi-Label Dataset Repository. URL ⟨http://mulan.sourceforge.net/datasets.html⟩.

[25] I. Katakis, G. Tsoumakas, I. Vlahavas, Multilabel text classification for automated tag suggestion, in: Proceedings of ECML PKDD'08 Discovery Challenge, Antwerp, Belgium, 2008, pp. 75–83.

[26] 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. http://dx.doi.org/10.1109/TASL.2007.913750.

[27] 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, ECCV'02, 2002, pp. 97–112. http://dx.doi.org/10.1007/3-540-47979-1_7.

[28] 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.

[29] B. Klimt, Y. Yang, The Enron Corpus: a new dataset for email classification research, in: Proceedings of ECML'04, Pisa, Italy, 2004, pp. 217–226. http://dx.doi.org/10.1007/978-3-540-30115-8_22.

[30] J. Read, P. Reutemann, MEKA Multi-Label Dataset Repository. URL ⟨http://meka.sourceforge.net/#datasets⟩.

[31] 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: Proceeding of the 14th Annual ACM International Conference on Multimedia, Santa Barbara, CA, USA, MULTIMEDIA'06, 2006, pp. 421–430. http://dx.doi.org/10.1145/1180639.1180727.

[32] A.N. Srivastava, B. Zane-Ulman, Discovering recurring anomalies in text reports regarding complex space systems, in: Aerospace Conference, IEEE, Big Sky, MT, USA, 2005, pp. 3853–3862. http://dx.doi.org/10.1109/AERO.2005.1559692.

[33] F. Charte, A. Rivera, M.J. Jesus, F. Herrera, A first approach to deal with imbalance in multi-label datasets, in: Hybrid Artificial Intelligent Systems, Lecture Notes in Computer Science, vol. 8073, 2013, pp. 150–160. http://dx.doi.org/10.1007/978-3-642-40846-5_16.

[34] G. Tsoumakas, I. Vlahavas, Random k-labelsets: an ensemble method for multilabel classification, in: Proceedings of the 18th European Conference on Machine Learning, Warsaw, Poland, ECML'07, vol. 4701, 2007, pp. 406–417. http://dx.doi.org/10.1007/978-3-540-74958-5_38.

[35] G. Tsoumakas, I. Katakis, I. Vlahavas, Effective and efficient multilabel classification in domains with large number of labels, in: Proceedings of ECML/PKDD Workshop on Mining Multidimensional Data, Antwerp, Belgium, MMD'08, 2008, pp. 30–44.

[36] K. Sechidis, G. Tsoumakas, I. Vlahavas, On the stratification of multi-label data, in: Machine Learning and Knowledge Discovery in Databases, Springer, Athens, Greece, 2011, pp. 145–158. http://dx.doi.org/10.1007/978-3-642-23808-6_10.

[37] J. Fürnkranz, E. Hüllermeier, E. Loza Mencía, K. Brinker, Multilabel classification via calibrated label ranking, Mach. Learn. 73 (2008) 133–153. http://dx.doi.org/10.1007/s10994-008-5064-8.

[38] W. Cheng, E. Hüllermeier, Combining instance-based learning and logistic regression for multilabel classification, Mach. Learn. 76 (2–3) (2009) 211–225. http://dx.doi.org/10.1007/s10994-009-5127-5.

[39] 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. http://dx.doi.org/10.1016/j.patcog.2012.03.004.

[40] L. Tang, S. Rajan, V.K. Narayanan, Large scale multi-label classification via metalabeler, in: Proceedings of the 18th International Conference on World Wide Web, WWW '09, 2009, pp. 211–220. http://dx.doi.org/10.1145/1526709.1526738.

[41] S. Garcıa, F. Herrera, An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons, J. Mach. Learn. Res. 9 (2677–2694) (2008) 66.

[42] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, Inf. Sci. 180 (10) (2010) 2044–2064. http://dx.doi.org/10.1016/j.ins.2009.12.010.

[43] 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. http://dx.doi.org/10.1016/j.knosys.2011.06.013.

Francisco Charte received his B.Eng. degree in Computer Science from the University of Jaén, in 2010, and his M.Sc. in Soft Computing and Intelligent Systems from the University of Granada, in 2011. He is currently a pre-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 28 Ph.D. students. He has published more than 240 papers in international journals. He is a Coauthor of the book "Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases" (World Scientific, 2001). He currently acts as an Editor in Chief of the international journal "Progress in Artificial Intelligence" (Springer). He acts as an Area Editor of the International Journal of Computational Intelligence Systems and Associated Editor of the journals: IEEE Transactions on Fuzzy Systems, Information Sciences, Knowledge and Information Systems, Advances in Fuzzy Systems, and International Journal of Applied Metaheuristics Computing; and he serves as a member of several journal editorial boards, among others: Fuzzy Sets and Systems, Applied Intelligence, Information Fusion, Evolutionary Intelligence, International Journal of Hybrid Intelligent Systems, Memetic Computation, and Swarm and Evolutionary Computation. 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 Paper Award (bestowed in 2011), and 2011 Lotfi A. Zadeh Prize Best paper Award of the International Fuzzy Systems Association. His current research interests include computing with words and decision making, bibliometrics, data mining, big data, cloud computing, data preparation, instance selection and generation, imperfect data, fuzzy rule based systems, genetic fuzzy systems, imbalanced classification, knowledge extraction based on evolutionary algorithms, memetic algorithms and genetic algorithms, biometrics.