MLeNN: A First Approach to Heuristic Multilabel Undersampling

Francisco Charte[1], Antonio J. Rivera[2], María J. del Jesus[2], and Francisco Herrera[1]

[1] Dep. of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain
[2] Dep. of Computer Science, University of Jaén, Jaén, Spain
{fcharte,herrera}@ugr.es,{arivera,mjjesus}@ujaen.es
http://simidat.ujaen.es,http://sci2s.ugr.es

Abstract. Learning from imbalanced multilabel data is a challenging task that has attracted considerable attention lately. Some resampling algorithms used in traditional classification, such as random undersampling and random oversampling, have been already adapted in order to work with multilabel datasets.

In this paper MLeNN (*MultiLabel edited Nearest Neighbor*), a heuristic multilabel undersampling algorithm based on the well-known Wilson's Edited Nearest Neighbor Rule, is proposed. The samples to be removed are heuristically selected, instead of randomly picked. The ability of MLeNN to improve classification results is experimentally tested, and its performance against multilabel random undersampling is analyzed. As will be shown, MLeNN is a competitive multilabel undersampling alternative, able to enhance significantly classification results.

Keywords: Multilabel Classification, Imbalanced Learning, Resampling, ENN.

1 Introduction

Multilabel classification (MLC) [1] has many real-world applications, being a subject which has drawn significant research attention. That most multilabel datasets (MLDs) are imbalanced is something taken for granted. Many existent methods deal with this problem through MLC algorithms adaptations [2], aiming to perform some kind of adjustment in the training phase to take into account the imbalanced nature of MLDs. There are also some proposals relying on data resampling [3], generating new instances in which minority labels appear or deleting instances associated to the majority ones.

Until now, most of the published multilabel resampling algorithms are random based, including random undersampling (RUS). The aim of this paper is to propose a multilabel undersampling method which heuristically, rather than randomly, selects the instances for removing. The heuristic is founded on the Edited Nearest Neighbor (ENN) rule [4] and relies on two measures to assess

E. Corchado et al. (Eds.): IDEAL 2014, LNCS 8669, pp. 1–9, 2014.

the imbalance level in MLDs, as well as on a distance metric between the sets of labels (labelsets) appearing in each pair of instances.

The behavior of the proposed algorithm, called MLeNN, will be experimentally proved by applying it to six MLDs, and analyzing the results produced by three MLC methods. The significance of these results will be statistically evaluated, and the benefits produced by MLeNN will be demonstrated.

This paper is structured as follows. In Section 2 a brief introduction to multilabel classification and imbalanced learning is provided. Section 3 describes the proposed method, while all the experimentation details can be found in Section 4. Finally, Section 5 offers the final conclusions.

2 Preliminaries

The main characteristic of multilabeled data, when compared with data used in traditional classification, consists of associating a group of relevant labels to each instance, instead of only one class. This group is a subset of the whole set of labels present in the MLD. Therefore, the goal on any MLC algorithm is to predict the subset of labels which should be associated to the instances in an MLD. A general introduction to MLC can be found in [5]. Additionally, a recent review of MLC methods is offered in [1].

Imbalanced learning is a problem thoroughly studied in traditional classification, and many solutions, based on different approaches, have been proposed to face it. This problem emerges when there are many instances belonging to some classes (majority), but only a few representing others (minority). Usually, classifiers tend to be biased to the majority classes, in detriment of the minority ones. A comprehensive review of traditional imbalanced learning solutions is provided in [6].

Most studies assume that all MLDs are imbalanced. A triad of measures directed to assess the imbalance level in MLDs are proposed in [3], along with two random resampling algorithms, one for oversampling (LP-ROS) and anoher for undersampling (LP-RUS). Two of the measures will be detailed in the following section, since the heuristic used by MLeNN rely on them. Several methods aimed to face the learning from multilabel imbalanced data, based on ensembles of MLC classifiers [7] and non-parametric probabilistic models [2], are also available.

In general, the imbalance level in MLDs is noticeably higher than in traditional datasets. Moreover, algorithms aiming to cope with this problem have to take into account that each sample belongs to multiple labels. Thus, procedures such as the creation of new instances or deletion of existing ones will influence several labels, rather than only one class.

3 Heuristic Multilabel Undersampling with MLeNN

Undersampling algorithms usually perform worse than oversampling ones [8], since they cause a loss of information by removing instances. The information loss is even greater when undersampling is applied to MLDs, as each removed

sample is representing not only one class but several labels. As a result, choosing the right instances to delete is of critical importance. Adapting ENN to work with MLDs needs to resolve two key points, how the candidates are selected and how the class differences among them and their neighbors are considered. MLeNN settles these points by firstly limiting the samples which can act as candidates to those in which none minority label appears, and secondly defining a metric distance to know what is the difference between any pair of labelsets.

Unlike LP-RUS [3], which has a random behavior, the MLeNN algorithm takes the samples to remove using a heuristic based on the two following bases:

- None of the minority labels can appear in the instance taken as reference to candidate for deletion.
- The labelset of the reference instance has to be different to that of its neighbors.

The second condition is based on the ENN rule [4], and adapted to the multilabel scenario as explained below. Algorithm 1 shows the MLeNN algorithm pseudo-code. The measures used and implementation details are discussed in the following subsections.

3.1 Candidate Selection

In order to choose which samples will act as candidates for removing, a method to know what labels are in minority is needed. Those instances in which any minority label appears will never be candidates, avoiding that some of the few samples representing a minority label are lost.

To complete this task, MLeNN relies on the measures proposed in [3]. Let D be an MLD, Y the full set of labels in it, and Y_i the labelset of the i-th instance. *IRLbl* (Equation 1) is a measure calculated individually to assess the imbalance level for each label. The higher is the *IRLbl* the larger would be the imbalance, allowing to know what labels are in minority or majority. *MeanIR* (Equation 2) is the average *IRLbl* for an MLD, useful to estimate the global imbalance level.

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

$$MeanIR = \frac{1}{|Y|} \sum_{y=Y_1}^{Y_{|Y|}} (IRLbl(y)). \tag{2}$$

MLeNN will iterate through all the MLD samples, taking as candidates those whose labelset does not contain any label with $IRLbl > MeanIR$. This way, all the instances containing a minority label will be preserved.

Algorithm 1. MLeNN algorithm pseudo-code.

Inputs: <Dataset> D, <Threshold> HT, <NumNeighbors> NN
Outputs: Preprocessed dataset

```
1: for each sample in D do
2:     for each label in getLabelset(D) do
3:         if IRLbl(label) > MeanIR then
4:             Jump to next sample              ▷ Preserve instance with minority labels
5:         end if
6:     end for
7:     numDifferences ← 0
8:     for each neighbor in nearestNeighbors(sample, NN) do
9:         if adjustedHammingDist(sample, neighbor) > HT then
10:            numDifferences ← numDifferences+1
11:        end if
12:    end for
13:    if numDifferences≥NN/2 then
14:        markForRemoving(sample)
15:    end if
16: end for
17: deleteAllMarkedSamples(D)
```

3.2 Labelset difference Evaluation

Wilson's ENN rule has been extensively used in traditional classification. The basic idea behind it is the following:

- Select a sample C as candidate.
- Look for $C's$ NN nearest neighbors. Usually $NN = 3$.
- If C class differs from the class of at least half of their neighbors (that is 2 when $NN = 3$), mark C for removing.

Since there is exclusively one class to compare with, the difference between the candidate class and that of any of its neighbors is either 0% (same class) or 100% (different classes). Therefore, the candidate will be removed when there is a 100% difference between its class and the class of half or more of its neighbors.

Table 1. Difference between the labelsets of two instances

label index	1	2	3	4	5	6	...	375	376
labelset1	0	1	1	0	0	1	**0**	0	0
labelset2	1	0	1	0	0	1	**0**	1	0
Dif. count	1	1	0	0	0	0	**0**	1	0

Multilabel instances have multiple labels associated, thus the difference between two samples labelsets could be 100%, but also any value below that and

above 0%. Let us consider the two labelsets shown in Table 1 and how their differences could be evaluated. They belong to an MLD with 376 different labels, but each instance is associated only to 3 or 4 of them. This is the case of the corel5k dataset.

Using the Hamming distance, it could be concluded that a difference of 3 exists between the two labelsets. As the total length (number of labels) is 376, this would give a 0.798% difference. However, if only the active labels (those which are relevant) in either labelset are considered the result would be totally different, since there are only 7 active labels in total. The percentage of difference would be 42.857%, far higher than the previous one. There are many MLDs with hundreds of distinct labels, but a low number of active labels in each sample. Calculating differences using the usual Hamming distance will always produce extremely low values. Thus, considering only active labels makes sense when it comes to evaluate differences among labelsets.

MLeNN calculates an *adjusted* Hamming distance between the candidate and their neighbors labelsets, counting the number of differences and dividing it by the number of active labels. As a result, a value in the range of [0,1] is obtained. Applying a configurable threshold (HT in Algorithm 1), the algorithm determines which of its neighbors will be considered as distinct.

4 Experimentation and Analysis

This section describes the experimental framework used to test the performance of the proposed algorithm. Afterwards, obtained results and their analysis are provided.

4.1 Experimental Framework

The paper experimentation has been structured in two phases. The first goal is to determine if MLeNN is able to improve classification results. It compares the output of classifiers before and after preprocessing the same set of datasets. The second phase aims to compare MLeNN performance against that of multilabel random undersampling, preprocessing the original datasets with LP-RUS [3].

The six datasets whose characteristics are shown in Table 2 were used to experimentally assess MLeNN[1]. All of them are from the MULAN repository [9], and have been repeatedly used in the literature. They have been partitioned following a 2x5 strategy. As can be inferred from the *MeanIR* values, five of these datasets are truly imbalanced, with values ranging from 7.20 to 256.40. On the contrary, the emotions dataset could not be actually considered as imbalanced. It has been included in the experimentation to test the behavior of MLeNN when used with non-imbalanced MLDs. MLeNN was applied to all of them with $NN = 3$ (3 neighbors) and $HT = 0.75$ (75% labelset difference threshold).

[1] This paper has an associated website at `http://simidat.ujaen.es/mlenn`. Both dataset partitions and the MLeNN program can be downloaded from it. This website also offers full tables of results.

Regarding the MLC algorithms used, a basic Binary Relevance (BR [10]) transformation method and two more advanced classifiers, Calibrated Label Ranking (CLR [11]) and Random k-labelsets (RAkEL [12]), were selected. The well-known C4.5 tree-based classification algorithm was used as underlying classifier where needed.

Table 2. Characteristics of the datasets used in experimentation

	Dataset	Instances	Attributes	Labels	MaxIR	MeanIR
1	corel5k	5000	499	374	1120.00	189.57
2	corel16k	13766	500	161	126.80	34.16
3	emotions	593	72	6	1.78	1.48
4	enron	1702	753	53	913.00	73.95
5	mediamill	43907	120	101	1092.55	256.40
6	yeast	2417	198	14	53.41	7.20

4.2 Results and Analysis

The outputs produced by the MLC algorithms, learning from the base MLDs and those obtained after preprocessing with MLeNN and ML-RUS, have been evaluated with three measures: Accuracy, Macro-FMeasure and Micro-FMeasure. The former is a sample-based evaluation measure, thus all labels are given equal weight, whereas the other two are label-based. Macro-averaging measures tend to emphasize the results of rare labels, while micro-averaging does the opposite. How these measures assess prediction performance can be found in [5]. An intuitive visual representation of those results is offered in Figure 1.

First, it can be seen that the undersampling performed by MLeNN has improved base results in many cases. However, there are some exceptions. The most remarkable is that of the emotions dataset, whose results tend to be worse after MLeNN has been applied. This led to the conclusion that undersampling, whether random or heuristic, should not be applied to MLDs which are not truly imbalanced. Another fact that can be visually confirmed in Fig. 1 is that the performance of MLeNN is almost always better than that of ML-RUS.

Aiming to formally endorse these results, a Wilcoxon non-parametric statistical test was used to compare MLeNN with base results, firstly, and with ML-RUS, secondly. The results of these tests are shown in Table 3 and Table 4. A star at the right of a value denotes that it is the best ranking for a given measure. The symbol $\leftrightarrows$ indicates that there is no statistical difference between this ranking and the best one, whereas the symbol $\downdownarrows$ states that a significant difference exists.

From the analysis of these results, that MLeNN is a competitive multilabel undersampling algorithm can be concluded, since it always achieves better performance than ML-RUS from an statistical point of view. Moreover, the undersampling conducted by MLeNN is able to improve classification results when compared with those obtained without preprocessing. MLeNN produced a statistically significant improvement in two of the three measures, despite the inclusion of an MLD such as emotions, which is not imbalanced, into the statistical study.

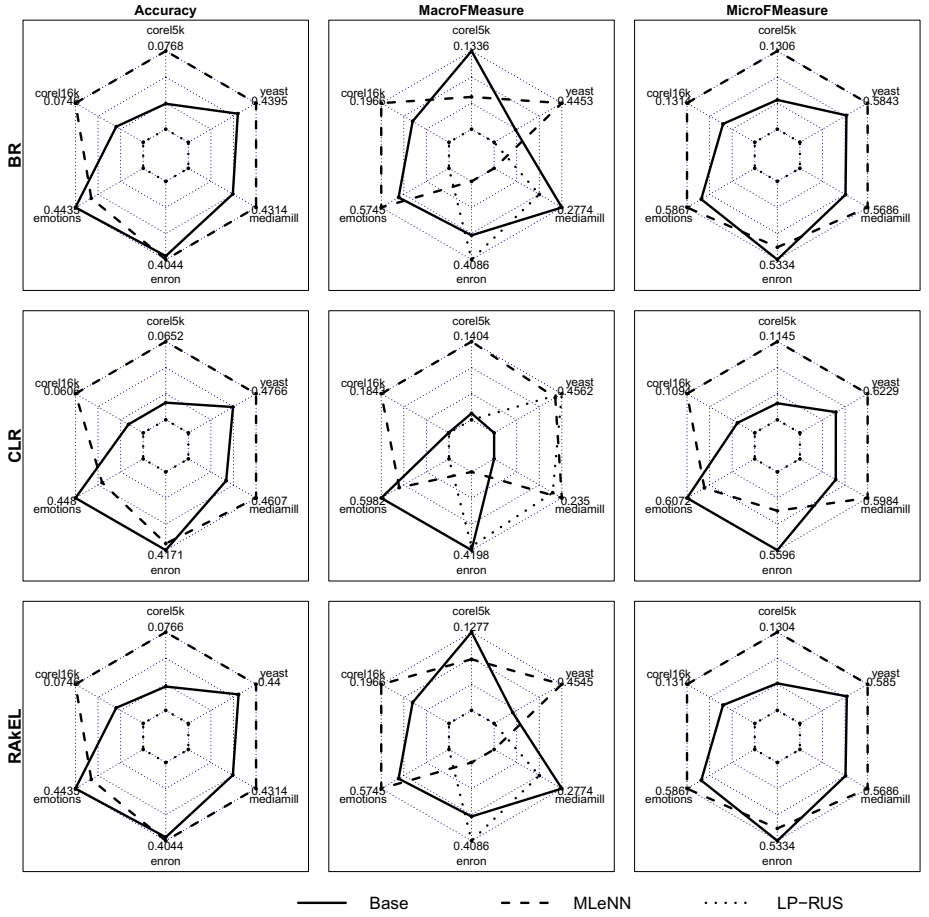

Fig. 1. Each plot shows classification results corresponding to a measure/algorithm combination

Table 3. First phase - Average Rankings

Algorithm	Accuracy		Macro-FM		Micro-FM	
Base	1.778	⇊	1.5556	⇆	1.778	⇊
MLeNN	1.222	⋆	1.4444	⋆	1.222	⋆

Table 4. Second phase - Average Rankings

Algorithm	Accuracy		Macro-FM		Micro-FM	
LP-RUS	2.000	⇊	1.6667	⇊	2.000	⇊
MLeNN	1.000	⋆	1.3333	⋆	1.000	⋆

5 Conclusions

The learning from imbalanced MLDs presents some serious difficulties. Several approaches have been proposed to overcome them, including random undersampling algorithms. This kind of methods does not usually work well, since they cause a significant loss of information potentially useful in the training process.

In this paper MLeNN, a novel multilabel heuristic undersampling algorithm, has been presented. The deleted instances are thoughtfully selected, instead of being randomly chosen. As the obtained results show, it is a technique able to improve classification results when applied to truly imbalanced MLDs. Moreover, it performs significant better than the random undersampling implemented by LP-RUS.

Acknowledgments. F. Charte is supported by the Spanish Ministry of Education under the FPU National Program (Ref. AP2010-0068). This work was partially supported by the Spanish Ministry of Science and Technology under projects TIN2011-28488 and TIN2012-33856 (FEDER Founds), and the Andalusian regional projects P10-TIC-06858 and P11-TIC-9704.

References

1. Zhang, M.-L., Zhou, Z.-H.: A review on multi-label learning algorithms. IEEE Trans. Knowl. Data Eng. (2013)
2. He, J., Gu, H., Liu, W.: Imbalanced multi-modal multi-label learning for subcellular localization prediction of human proteins with both single and multiple sites. PloS One 7(6), 7155 (2012)
3. Charte, F., Rivera, A., del Jesus, M.J., Herrera, F.: A first approach to deal with imbalance in multi-label datasets. In: Pan, J.-S., Polycarpou, M.M., Woźniak, M., de Carvalho, A.C.P.L.F., Quintián, H., Corchado, E. (eds.) HAIS 2013. LNCS, vol. 8073, pp. 150–160. Springer, Heidelberg (2013)
4. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans. on SMC-2(3), 408–421 (1972)
5. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining Multi-label Data. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, ch. 34, pp. 667–685. Springer US, Boston (2010)
6. Haibo, H., Yunqian, M.: Imbalanced Learning: Foundations, Algorithms, and Applications. Wiley-IEEE Press (2013)
7. Tahir, M.A., Kittler, J., Bouridane, A.: Multilabel classification using heterogeneous ensemble of multi-label classifiers. Pattern Recognit. Lett. 33(5), 513–523 (2012)
8. García, V., Sánchez, J., Mollineda, R.: On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. Knowl. Based Systems 25(1), 13–21 (2012)
9. Tsoumakas, G., Xioufis, E.S., Vilcek, J., Vlahavas, I.: MULAN: A Java Library for Multi-Label Learning. J. Mach. Learn. Res. 12, 2411–2414 (2011)

10. Godbole, S., Sarawagi, S.: Discriminative Methods for Multi-labeled Classification. In: Dai, H., Srikant, R., Zhang, C. (eds.) PAKDD 2004. LNCS (LNAI), vol. 3056, pp. 22–30. Springer, Heidelberg (2004)
11. Fürnkranz, J., Hüllermeier, E., Loza Mencía, E., Brinker, K.: Multilabel classification via calibrated label ranking. Mach. Learn. 73, 133–153 (2008)
12. Tsoumakas, G., Vlahavas, I.: Random *k*-labelsets: An ensemble method for multilabel classification. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 406–417. Springer, Heidelberg (2007)