2014 IEEE 26th International Conference on Tools with Artificial Intelligence

Multi-Label classification: Dealing with Imbalance by Combining Labels

Ming Fang*, YuQi Xiao*, ChongJun Wang† and JunYuan Xie†
National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210046, China
*{fm5821, xiao19910705}@163.com
†{chjwang, jyxie}@nju.edu.cn

***Abstract*—Data imbalance is a common problem both in single-label classification (*SLC*) and multi-label classification (*MLC*). There is no doubt that the predicting result suffers from this problem. Although, a broad range of studies associate with imbalance problem, most of them focus on *SLC* and for *MLC* is relatively less. Actually, this problem arising in *MLC* is more frequent and complex than in *SLC*. In this paper, we proceed from dealing with imbalance problem for *MLC* and propose a new approach called *DEML*. *DEML* transforms the whole labelset of multi-label dataset into some subsets and each subset is treated as a multi-class dataset with balanced class distribution, which not only addressing imbalance problem but also preserving dataset integrity and consistency. Extensive experiments show that *DEML* possesses highly competitive performance both in computation and effectiveness.**

***Keywords*-multi-label classification, imbalance, combining labels.**

I. Introduction

Both Single-Label Classification (*SLC*, including binary-class classification and multi-class classification) and Multi-Label Classification (*MLC*) are important research fields in supervised learning. However, neither of them can avoid imbalance problem which has negative effect on the performance of classifier. In *SLC*, usually dealing with imbalance problem adopts under-sampling and over-sampling methods [10]. Introducing cost sensitive classification [11] can also solve the problem in effect. However, addressing imbalance problem in *MLC* is more complex and it is even hard to define what kind of labels distribution is imbalanced. In [12], the author proposes three criteria to measure imbalance, *i.e. IRperLabel, MeanIR, CVIR*. Differently, we adopt a simple way as follows.

In *SLC*, determining the level of imbalance is relatively easy for some classes being too many or few. Similarly, if multi-label dataset is treated as q binary-datasets for each label that q is the size of labelset, the criterion of imbalance in *SLC* can be applied to *MLC* logically. In other words, if any one of q binary-datasets is imbalanced, we think the entire dataset is imbalanced.

The final goal of this paper is to improve the performance of multi-label classifier by dealing with labels imbalance. Comparing with *MLC* and *SLC*, the obvious difference is the relationship of labels in *MLC* is varied [7], but in *SLC* is tedious. In fact, many approaches have been proposed to exploit this relationship and results from [3], [6], [7], [8], [9] have shown that the relationship of labels can be treated as extra information to help classify. Our method also makes use of the relationship but in a different strategy. The basic idea is that, for binary-dataset the label is just composed of class-$\{0, 1\}$ which represent unpresence and presence respectively, so the cause for imbalance is very simple for the label containing more either 0 or 1. Instead of by under-sampling and over-sampling in *SLC*, we creates new class values to reduce overmuch classes through combining labels together as a subset. As shown Figure 1[i] it is difficult to find a line to sparate '+' and '-' in an imbalanced dataset. However, in Figure 1[ii] we transform some '+' into '*', then lines a and b could classify these points easily.

In this paper, we introduce the entropy as criterion to measure labels imbalance which is as simple and efficient as in *SLC*. At the same time, we propose a new approach named *DEML* (*D*ealing with labels imbalance by *E*ntropy for *M*ulti-*L*abel classification) converting multi-label classification task to multi-class classification task that not only addressing imbalance problem but also preserving dataset integrity and consistency. Extensive experiments show that *DEML* possesses highly competitive performance both in computation and effectiveness.

The rest of this paper is organized as follows. In section 2, we discuss related work on exploiting the relationship of labels. Section 3 describes *DEML* method in detail. Experiments present in section 4. Finally, section 5 shows conclusion.

II. Related work

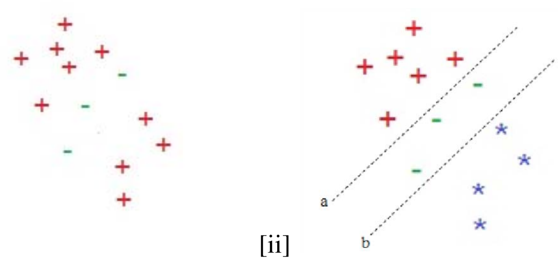
To describe *MLC* formally and better, we use $X = R^d$ denotes the d-dimensional instance space and $Y = \{y_1, y_2, \cdots, y_q\}, y_i \in \{0, 1\}$ denotes the label space [2]. The multi-label dataset can be expressed as

$$D = \{(\mathbf{x_i}, Y_i) | 1 \leq i \leq n, \mathbf{x_i} \in R^d, Y_i \subseteq Y\}$$

d, q, n represent the number of features, number of labels and number of instances.

During the past few years, a great number of approaches have been proposed which decompose labelset into subsets with different strategies. Labels in one subset are treated as

1082-3409/14 $31.00 © 2014 IEEE
DOI 10.1109/ICTAI.2014.42



[i] [ii]

Figure 1. [i] shows an imbalance binary-dataset and it is difficult to find a line to sparate '+' and '-'. [ii] transforms some '+' into '*' , then lines a and b could classify these points easily.

correlative dependence, while the relationship of subsets is conditionally independent. Broadly speaking, according to the size of subset, these approaches could be categorized into three families as following [2]: *First-order* strategy [1], [5], *Second-order* strategy [6], [8], *High-order* strategy [3], [7], [9].

In first-order strategy, *BR* [1] is the simplest algorithm for learning q binary-classification models independently, but in some cases, *BR* could achieve higher performance than other complex models. Usually, *BR* is used to compare with other algorithms as a benchmark. Another algorithm in first-order we interested is *MLkNN* [5] which basing on the kNN(k-nearest neighbors) lazy learning algorithm could learn labels distribution by computing prior probabilities and posterior probabilities. To some extent, *MLkNN* is imbalance insensitive.

CLR [8] belongs to second-order strategy. Firstly, it generates *q(q-1)/2* binary-datasets D_{ij} by pairwise comparison (y_i, y_j), where $(y_i = 0, y_j = 1)$ or $(y_i = 1, y_j = 0)$. Additional, q auxiliary binary-datasets will be induced for each new pair(y_i, y_V), y_V is a virtual label. After above steps, *CLR* could control labels distribution well for each binary classifier.

In high-order strategy, *LP* [1] algorithm transforms multi-label dataset into single-label dataset directly by mapping labelset into new class values with binary encoding. However, overly high complexity and few training examples for each class limit *LP* to be used widely. To overcome these drawbacks, *RAkEL* [9] is proposed which selects *k* labels randomly as a subset and learns *m LP* classifiers on each subset. One advantage of *RAkEL* against *LP* is downsizing the size of labelset for each classification model. *CC* [3] adopts another strategy who predicts labels through one by one like *BR*, but puts previous labels which have already predicted into the instance space as extra features. Therefore, it not only overcomes the disadvantage of *BR* ignoring the relationship of labels, but also maintains acceptable computational complexity.

III. DEML

To the best of our knowledge, *DEML* approach is the first classifier aiming at dealing with imbalance problem for multi-label dataset. Other methods like *MLkNN* and *CLR* could handle imbalance more or less but that are not their prime targets. At the same time, *DEML* is a flexible method belonging to high-order strategy which possesses stronger correlation-modeling capabilities than first-order and second-order strategy for real-world problem.

A. Imbalance Criterion

Determining the level of imbalance for one subset in an efficient way is very important. Here, we adopt the entropy (Eq.1) as the imbalance criterion, which is used to measure amount of information. According to the standpoint of informational theory, more even one subset, higher the entropy. On the other hand, to compare with different subsets owning different number of classes, the entropy needs to be normalized (Eq.2). Consequently, we have

$$E(subset_k) = -\sum p_i log_2(p_i) \tag{1}$$

$$NE(subset_k) = E(subset_k)/log_2(c) \tag{2}$$

where c statistics the number of appearing classes and p_i denotes the probability of $class_i$ in one subset.

Remarkably, when a subset contains more labels, more classes will be created. Inevitably, some classes will associate with few training instances usually less than 100, which will negatively influence the entire subset entropy. To tolerate this situation, we introduce a $t$ole$r$ant function:

$$tr_l = t^l \tag{3}$$

where l denotes the number of labels and tr_l will have different values, according to the size of subset. *DEML* uses the entropy to measure each subset. If one subset is imbalanced, $DEML$ needs to search labels from the rest until NE reaches tr_l.

B. Algorithm

It is a challenging task to construct m even subsets. To reduce computational cost, we adopt a random strategy to search labels, see algorithm 1. The main idea is that for a given label checking whether it has been marked, if so meaning the label is already contained by an even subset and no additional calculation is needed (line4∼5), otherwise we add this label to a new subset (line9) and call this label as Main Label (ML) of the subset. Then, we start a loop from $l = 1$ to $\lceil q/2 \rceil$ (line7∼23). When $l = 1$, meaning this ML is the only one in the subset and its entropy is equal to the subset. The loop will stop if this ML is even enough. When $l \geq 2$, meaning that the subset needs to contain other labels to maintain balance. It will be time consuming if we adopt traverse strategy because of C_q^l

Algorithm 1 *DEML* description

Input:
Training dataset D;
Size q of labelset Y;
Entropy threshold t;
Unseen instance $\mathbf{x}$;
Output: Predict $\mathbf{Result}$

1: $Y^\star \leftarrow \{\}$; // marked label
2: $idx = 0$;
3: **for** $k = 1$ to q **do**
4:   **if** $y_k \in Y^\star$ **then**
5:     $goto(3)$
6:   **end if**
7:   **for** $l = 1$ to $[q/2]$ **do**
8:     **for** $r = 0$ to $(l-1) \times q$ **do**
9:       $subset_{idx} \leftarrow \{y_k\}$
10:       **while** $subset_{idx}.size() \neq l$ **do**
11:         //choose a label randomly except y_k
12:         $subset_{idx} \leftarrow$ rand(y_k);
13:       **end while**
14:       $tr_l = t^l$ // Eq.(3)
15:       // NE according to Eq.(2)
16:       **if** $NE(subset_{idx}) > tr_l$ **then**
17:         // marking even label
18:         $Y^\star \leftarrow Y^\star \cup subset_{idx}$;
19:         $idx = idx + 1$;
20:         $goto(3)$;
21:       **end if**
22:     **end for**
23:   **end for**
24: **end for**
25: train *LP* classifiers h_i on each subset;
26: train a *BR* classifier h_{i+1} on $\{Y \setminus Y^\star\}$;
27: //ensemble all classifiers' result.
28: **for** each h_i **do**
29:   $\mathbf{Result} \leftarrow h_i(\mathbf{x})$
30: **end for**

possibilities. So we use a random function to search $l-1$ different labels (line10∼12) until the subset entropy satisfies tr_l (line16). The variable r indicates the searching rounds for each l (line8). Then, marking all labels in the subset as even labels (line18) and starting to deal with the next label. After finishing subsets constructing, we train m *LP* classifiers on each subset and ensemble all outcomes as final predicting result for an unseen instance (line25∼30). Additionally, if l reaches $[q/2]$ which means there is no proper labels making the subset balance, the ML will be treated as independency (line26).

C. Complexity Discuss

DEML is an efficient algorithm for only training $m(m \leq q)$ multi-label classification models which implies linear complexity. The cost of constructing subsets which is $O(n \times q^3 \times m)$ in worst case is much less comparing with training a classification model. Consequently, the computational complexity of *DEML* is $O(h(D, C) \times m + n \times q^3 \times m)$, h is a multi-class classifier with C class values.

Dataset	n	d	q	LEnt(D)	Type
Emotions	593	72	6	0.8818	music
Yeast	2417	103	14	0.7146	music
CAL500	502	68	174	0.4763	music
Enron	1702	1001	53	0.2407	text
Genbase	662	1185	27	0.2291	biology
Mediamill	10000	120	101	0.1556	video
Medical	978	1449	45	0.1423	text
Bibtex	7395	1836	159	0.1077	music
Bookmarks	10000	2150	208	0.0745	text
Corel5k	5000	499	374	0.0614	images

Table 1. A collection of datasets with their statistics.

IV. Experiments

A. Evaluation Metrics

- $Micro - averaging$

$$B_{micro} = B(\sum_{i=1}^{q} Tp_i, \sum_{i=1}^{q} Fp_i, \sum_{i=1}^{q} Tn_i, \sum_{i=1}^{q} Fn_i) \tag{4}$$

- $Macro - averaging$

$$B_{macro} = \frac{1}{q} \sum_{i=1}^{q} B(Tp_i, Fp_i, Tn_i, Fn_i) \tag{5}$$

For both of $F1_{micro}$ and $F1_{macro}$, the larger the value, the better the performance.

B. Dataset

For the experiment, we have collected ten multi-label datasets from the site[1]. Detailed statistics is summarized in Table 1 (we randomly select 10000 instances from Mediamill and Bookmarks dataset). In addition, the *Label Entropy* as shown Eq.6 is the mean value of labels entropy measuring the average of level of imbalance in a multi-label dataset.

- $LabelEntropy$

$$LEnt(D) = \frac{1}{q} \sum_{i=1}^{q} NE(y_i) \tag{6}$$

The goal of this experiment is testing the performance of *DEML* on a variety of datasets with variable distributions of labels. In other words, we concern on how the *LEnt* impact on predicting result.

[1]http://mlkd.csd.auth.gr/multilabel.html

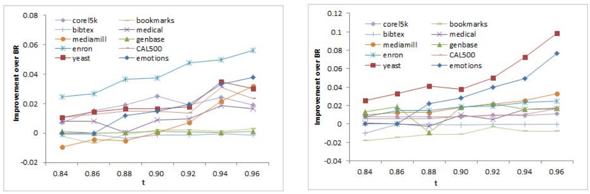


Figure 2. Micro $F1$ and Macro $F1$ measure with respect to t.

C. Setup

DEML is implemented within the MULAN [4] platform which is a popular open source tool and integrates a great number of multi-label algorithms. The algorithm SMO in Weka [13] Framework is introduced as single-label classifier with default parameters. All algorithms are running on Jdk-1.7 platform and 64-bit machine with I5 CPU, 4GB RAM.

For comparing methods, we choice *BR* as a benchmark, two ensemble methods *ECC* and *RAkEL*, two imbalance insensitive methods *MLkNN* and *CLR*, which are introduced in section2. Detail settings as follows: for *RAkEL* we set parameter $k = 3$ and $m = 2q$ [9] and ensemble iterator of *ECC* is set to 10 [3]. For *MLkNN*, the number of nearest neighbors *k* is set to 10 and other parameters are default. For *BR* and *CLR*, parameters are default. Ten-fold cross validation is applied to each experimental round.

Figure 2 show the improvement of *DEML* over *BR* in terms of $F1_{micro}$, $F1_{macro}$ with respect to the threshold t from 0.84 to 0.96 . We observe that the performance of *DEML* is enhancing while arising t and achieves the best when $t = 0.96$. However, the higher t which means stricter searching condition will lead to higher time consuming for constructing each subset. Experimentally, we set the threshold $t = 0.94$ getting the balance of performance and efficiency.

D. Result and Discuss

Tables 2 and 3 present the detailed results of all algorithms on each experimental dataset including the average and standard deviation. DNF indicates that the experiment Did Not Finish for the error out of memory. On the whole, *DEML* possesses higher average rank comparing with other classifiers both in the micro-$F1$ and macro-$F1$ measures. For time consuming, we assume the cost of *BR* is 1 as a benchmark, detailed comparing is shown in Figure 3 .

1) Ensemble Methods: On the Emotions dataset, *DEML* achieves worse performance than *ECC* and *RAkEL*. The reason is that the Emotions is a balance dataset according to the *LEnt* metric, but our method is aimed at handling with imbalance. Actually, *DEML* will degenerate into *BR* when each label in one dataset is even enough. On the other hand, the classical ensemble strategy could get outstanding results on the balance dataset.

On extreme-imbalance datasets Bibtex, Bookmarks and COREL5K whose *LEnt* are roughly less than 0.1, the performance of *DEML* is not stable. Particularly, in the micro measure, *DEML* is inferior to *RAkEL* on the Bibtex and loses to *ECC* on the COREL5K. In the macro measure, *DEML* is defeated by *RAkEL* on the Bookmarks and *ECC* on the COREL5K respectively. What's more, the time cost of *DEML* is huge against other methods on the COREL5K. The reason is that it is the most imbalanced dataset amount all and *DEML* must search deeply to construct even subsets what is time consuming.

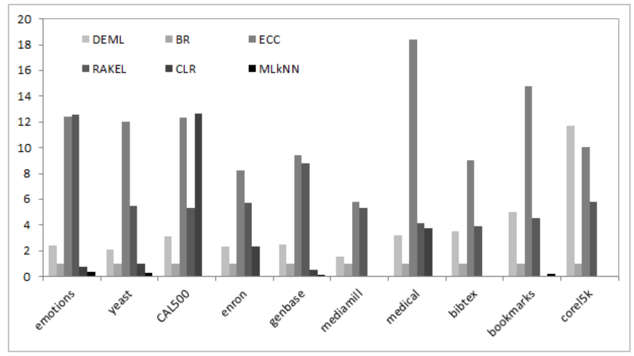


Figure 3. Time consuming, the cost of *BR* is 1 as a benchmark.

On the rest of regular-imbalance datasets whose *LEnt* are between 0.1 and 0.8, *DEML* performance is superior to *ECC* and *RAkEL* with accepted time consuming. Wonderingly, *DEML* does not act well as ideal on the CAL500, although this dataset is regular-imbalance. In our point, *DEML* encounters the same problem like *LP* for classes in one subset associating with few training examples. Therefore, it is very outstanding that *DEML* could gain higher performance when the dataset is regular-imbalance with middle or large scale.

2) Imbalance-insensitive Methods: DEML had a clear advantage over two imbalance-insensitive methods in terms of micro-*F1* and macro-*F1* measures, but *MLkNN* owns peak efficiency. For *CLR*, the experiments on large-scale datasets have failed because of out of memory error.

V. Conclusion

In this paper, a novel method *DEML* for *MLC* is proposed to deal with labels imbalance. The entropy is employed to measure the level of imbalance which is as simple and efficient as in *SLC*. The subset constructing algorithm adopts random strategy giving consideration to accuracy and efficiency. What's more, the number of final classification models is linear with the size of labelset ensured *DEML* possessing low complexity. Extensive experiments have shown that our approach is able to adapt flexibly to datasets with different labelsets and proves superior to other algorithms when the dataset is imbalanced.

In the future, we will explore if there exist a better method to handle extreme-imbalance dataset efficiently.

Dataset	DEML	BR	ECC	RAKEL	CLR	MLkNN
Emotions	68.12±4.23 (3)	65.50±4.15 (6)	70.19±3.95 (1)	69.95±4.46 (2)	66.58±4.13 (4)	65.98±4.23 (5)
Yeast	66.83±1.83 (1)	63.46±2.12 (6)	65.43±2.13 (3)	65.45±2.28 (2)	64.01±2.09 (5)	64.71±2.45 (4)
CAL500	36.04±1.22 (2)	33.17±1.19 (5)	36.62±1.17 (1)	33.88±1.55 (3)	33.56±1.20 (4)	32.09±1.68 (6)
Enron	56.53±2.88 (2)	51.40±2.69 (5)	55.51±2.13 (3)	53.18± 2.50 (4)	56.61±2.44 (1)	47.78±2.26 (6)
Genbase	99.19±0.72 (1)	99.08±0.45 (2)	99.03±0.62 (3)	99.01±0.45 (4)	98.85±0.78 (5)	94.62±3.19 (6)
Mediamill	57.04±1.06 (2)	54.15±0.74 (5)	55.28±1.14 (3)	54.16± 0.77 (4)	DNF	59.98±0.71 (1)
Medical	82.94±1.99 (1)	81.12± 2.02 (3)	81.38±1.56 (2)	81.05± 2.14 (4)	81.04±1.76 (5)	68.00±4.01 (6)
Bibtex	33.19±3.49 (3)	33.34±3.07 (1)	30.45±3.92 (4)	33.24±2.90 (2)	DNF	22.18 ±1.57 (5)
Bookmarks	24.40±1.22 (1)	24.11±1.04 (3)	23.76±1.16 (4)	24.30±1.11 (2)	DNF	18.01±1.19 (5)
Corel5k	21.30±1.71 (2)	19.40±1.31 (4)	21.90±2.02 (1)	19.44±1.37 (3)	DNF	6.31±0.58 (5)
average rank	1.8	4	2.5	3	4	4.9

Table 2. Performance in term of micro $F1$(mean±std%).

Dataset	DEML	BR	ECC	RAKEL	CLR	MLkNN
Emotions	65.49±4.56 (3)	60.14±3.38 (6)	68.50±3.96 (1)	67.36±5.19 (2)	62.51±3.47 (4)	62.43±4.13 (5)
Yeast	39.26±1.18 (1)	32.53±0.93 (6)	35.47±1.12 (4)	36.53±1.36 (3)	33.84±1.18 (5)	38.03±1.81 (2)
CAL500	18.67±1.35 (2)	17.43±1.50 (5)	21.40±1.63 (1)	17.75±1.52 (4)	17.68±1.51 (3)	17.14±1.68 (6)
Enron	34.79±4.51 (1)	32.41±3.07 (5)	33.54±3.90 (4)	33.60±3.75 (3)	33.71±4.12 (2)	25.69±5.05 (6)
Genbase	96.41±2.89 (1)	96.04±1.64 (2)	95.98±2.28 (3)	95.67±1.64 (4)	95.41±2.46 (5)	84.03±8.67 (6)
Mediamill	24.42±3.60 (2)	23.26±3.57 (5)	23.63±3.55 (3)	23.28±3.58 (4)	DNF	32.33±2.00 (1)
Medical	77.70±5.57 (1)	76.02±5.71 (2)	75.90±5.92 (3)	75.85±6.07 (4)	75.75±5.64 (5)	65.74±4.37 (6)
bibtex	31.24±4.15 (1)	30.26±4.43 (3)	29.94±4.60 (4)	31.15±4.34 (2)	DNF	7.57±0.56 (5)
Bookmarks	16.94±1.76 (3)	17.13±1.59 (2)	16.30±1.73 (4)	17.20±1.52 (1)	DNF	5.22±0.77 (5)
Corel5k	52.11±1.96 (2)	51.02±1.73 (4)	52.94±2.43 (1)	51.07±1.68 (3)	DNF	50.94±2.10 (5)
average rank	1.7	4	2.8	3	4	4.7

Table 3. Performance in term of macro $F1$(mean±std%).

VI. ACKNOWLEDGEMENT

This paper is supported by the National Natural Science Foundation of China(Grant No.61375069, 61105069) and the Science and Technology Support Foundation of Jiangsu Province(Grant No. BE2012161).

REFERENCES

[1] Tsoumakas G, Katakis I, Vlahavas I. *A review of multi-label classification methods*. Procee dings of the 2nd ADBIS Workshop on Data Mining and Knowledge Discovery (ADMKD 2006). 2006: 99-109.

[2] Min-Ling Zhang, Zhi-Hua Zhou. *A Review on Mutil-Label Learning Algorithms*. Transactions on Knowledge and Data Engineering, 31 May 2013.

[3] J. Read, B. Pfahringer, G.Holmes, and E. Frank. *Classifier Chains for Multi-label Classification*. In Proc. of the ECML/PKDD 2009, Bled, Slovenia, 2009, pp. 254-269.

[4] G. Tsoumakas, E. Spyromitros, J. Vilcek, and I. Vlahavas. *Mulan: A Java Library for Multi-Label Learning*. Journal of Machine Learning Research, vol.12,pp,2411-2414,2011.

[5] Min-Ling Zhang, Zhi-Hua Zhou. *A k-nearest neighbor based algorithm for multi-label classification*. Granular Computing, 2005 IEEE International Conference on. IEEE, 2005, 2: 718-721.

[6] Min-Ling Zhang, Zhi-Hua Zhou. *Multilabel neural networks with applications to functional genomics and text categorization*. Knowledge and Data Engineering, IEEE Transactions on, 2006, 18(10): 1338-1351.

[7] M. L. Zhang, K. Zhang. *Multi-label learning by exploiting label dependency*. In Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Pages 999-1007, Washington, DC, 2010.

[8] J. Fuernkranz, E. Huellermeier, E. L. Mencia, K. Eneldo. *multilabel classification via calibrated label ranking*. Machine Learning, 73(2):133-153, 2008.

[9] G. Tsoumakas, I. Vlahavas. *Random k-labelsets: an ensemble method for multilabel classification*. In ECML '07: 18th European Conference on Machine Learning, pages 406-417. Springer-Verlag, 2007.

[10] N Japkowicz. *Learning from imbalanced data sets: A comparison of various strategies*. pp. 10-15 AAAI Press(2000).

[11] F. Provost, T. Fawcett *Robust classification for imprecise environments*. Machine Learning 42, 203-231(2001).

[12] F. Charte, A. Rivera, Ma. Jesus, F. Herrera. *A First Approach to Deal with Imbalance in Multi-label Datasets*. HAIS 2013, LNAI 8073, pp. 150C160, 2013.

[13] I. H. Witten, E. Frank. *Data Mining: Practiacal machine learning tools and techniques*. Morgan Kaufmann, San Francisco, second edition, 2005.