Learning and Nonlinear Models - Journal of the Brazilian Computational Intelligence Society, Vol. 11, Iss. 1, pp. 26–47, 2013.

RB: A NEW METHOD FOR CONSTRUCTING MULTI-LABEL CLASSIFIERS BASED ON RANDOM SELECTION AND BAGGING

Patrícia Pachiega da Gama[1], Flavia C. Bernardini[2] and Bianca Zadrozny[1,3]

[1]Computer Science Institute, Fluminense Federal University (UFF), Niterói, Rio de Janeiro – Brazil
[2]Laboratory of Innovation on Systems Development (LabIDeS), Institute of Science and Technology, Fluminense Federal University (UFF), Rio das Ostras, Rio de Janeiro – Brazil
[3]IBM Research Brasil, Rio de Janeiro, RJ – Brazil
pgama@ic.uff.br, fcbernardini@puro.uff.br, biancaz@br.ibm.com

Abstract

In many real world prediction problems, a classifier must, or should, assign more than one label to an instance, e.g. prediction of machine failures, musical genre classification, etc. For this kind of problem, multi-label classification methods are needed. One approach frequently used to learn multi-label predictors divides the problem into one or more multi-class classification problems, and combines the models constructed for each sub-problem to classify new instances with multiple labels. Although there are many multi-label learning methods, there is a need for exploring methods that can lead to improvement in prediction power. In this work, we propose and evaluate a new method, called RB (Random-Bagging), based on dataset transformation and combination of classifiers. Six real-world datasets were used to evaluate our method, which was compared to three existing methods. Results were considered promising.

Keywords – Multi-label learning classifiers, Bagging, Label Random Selection.

1 Introduction

One of the main purposes of machine learning is to construct models based on examples, or instances, collected from some domain. A group of these algorithms, known as supervised learning algorithms, are able to induce models, known as classifiers, based on labeled instances. When a new instance is given to a classifier, the model is able to predict a single class label for this instance. However, there are some domains in which the collected instances are labeled with more than one label. For instance, we can cite text, videos, images and music labeling; failure diagnosis, and so on. A common approach for solving this kind of problem is decomposing the original multi-label problem into one or various multiclass learning problems. There are many methods in the literature based on this approach, applied to many different domains [1–7]. There are some methods proposed in literature that usually decompose the initial problem into a number of independent binary problems, one for each possible problems [2, 8, 9]. Others consider the pairwise relations between labels, such as the interaction between any pair of labels [10, 11], or the full-order style of all other labels' influence on each label [12, 13]. Methods considering relation between (or among) labels can be very effective, however they can be computationally inefficient.

In this work, we propose a method to construct multi-label classifiers, that divides the original multi-label problem into multi-class subproblems and combines the induced classifiers. The combination method is based on bagging combination method [14]. The advantage of our proposed method is the simplicity of the transformation process, associated to the complementarity of the

constructed and combined multiclass classifiers. In the original bagging, multiple bootstrap samples, or subsets of the original datasets, are constructed, selecting random instances with replacement from the original dataset. A classifier is induced for each subset. There are other proposals in the literature to obtain varied classifiers from the same dataset, including random feature selection [15]. In this work, we propose our method based on random label selection, which allows transforming the original problem into multiclass subproblems, and also allows varied classifiers to be learned from the same original dataset. Our method was implemented using the Mulan library[1] [16], based on Weka [17]. Controlled experiments on six natural benchmark datasets, available with Mulan, were conducted to evaluate our proposed method, and show that our method is effective in some situations.

This work is organized as follows: Section 2 describes some machine learning definitions and concepts and some multi-label learning methods, used to compare to our proposed method. Section 3 describes the RB method, proposed in this work. Section 4 describes and analyzes experiments executed to evaluate the RB method. Finally, Section 5 concludes this work and discusses some future work.

2 Multiclass and Multi-label Supervised Machine Learning

In many real world classification problems the examples are associated with a single label. The input for single-label supervised learning algorithms is a single-labeled dataset S_s, with N instances $T_i, i = 1, ..., N$, chosen from a domain X with fixed, arbitrary and unknown distribution $\mathcal{D}$, of the form $(\mathbf{x}_i, y_i)$, with $i = 1, ..., N$, for some unknown function $f(x) = y$. x_i are vectors typically of the form $(x_{i1}, ..., x_{iM})$, with discrete or continuous values, where x_{ij} refers to the value of feature j, named X_j, of the instance T_i. In classification problems, the y_i is a single label value, and the possible values belong to a discrete set of labels L, *i.e* $y \in L = \{l_1, ..., l_{|L|}\}$. These values refer to the values of feature Y, frequently called class feature. For $|L| = 2$, we have a binary problem; for $|L| > 2$, we have a multiclass problem. Descriptions of many algorithms for supervised learning of single label classifiers can be found in [17, 18].

In multi-label problems which appear in many different domains, such as image, text, music, proteins and genome classification [1–4], or failure diagnosis [5], to cite just a few, the input to the multi-label learning algorithms is also a dataset S_m (dataset S with multiple labels), with N instances $T_i, i = 1, ..., N$, chosen from a domain X with fixed, arbitrary and unknown distribution $\mathcal{D}$, of the form $(\mathbf{x}_i, Y_i)$, with $i = 1, ..., N$, for some unknown function $f(\mathbf{x}) = Y$. L is the set of possible labels of the domain $\mathcal{D}$, and $Y_i \subseteq L$, *i.e.*, Y_i is the set of labels of the ith instance. The output of multi-label learning algorithms is a classifier $\mathbf{h}$ that labels an instance $\mathbf{x}_i$ with a set $Z_i = \mathbf{h}(\mathbf{x}_i)$, i.e., Z_i is the set of labels predicted by $\mathbf{h}$ for $\mathbf{x}_i$[2].

2.1 Characteristics and Statistics of Multi-label Datasets

In some multi-label datasets, the number of labels associated with an instance is small when compared to the total number of possible labels $|L|$. This number can be seen as a parameter that influences the performance of different multi-label methods. There are two measures for evaluating the characteristics of a dataset: cardinality $Card$ and density $Dens$ [19]. The cardinality of S_m is the mean of the number of labels of the instances that belong to S_m, defined by Equation 1, and the density of S_m is the mean of the number of labels of the instances that belong to S_m divided by $|L|$, defined by Equation 2. Two datasets with approximately the same cardinality but with great difference in density may not exhibit the same properties, which causes different behaviors in multi-label learning methods. The number of distinct labels is also important for many multi-label methods based on dataset transformation. It is thus important to observe such measures when using multi-label learning methods.

$$Card = \frac{1}{N} \sum_{i=1}^{N} |Y_i| \tag{1}$$

[1] Available at `http://mulan.sourceforge.net`.

[2] In this work, we use T_i to refer to an instance with associated label y_i ou Y_i, and we use $\mathbf{x}_i$ when we are not considering the associate label, or $\mathbf{x}_i$ does not have an associated label yet.

$$Dens = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i|}{|L|} \tag{2}$$

2.2 Evaluation Measures

There are three classes of measures to evaluate multi-label classifiers: based on instances, based on labels and based on ranking [19]. Measures based on instances used in this work are: *Hamming Loss* (Ham), Accuracy (Acc), $F1$ and *Subset Accuracy* ($SubAcc$), respectively defined by Equations 3 to 6[3]. It should be observed that $SubAcc$ is extremely conservative because it measures how many times the classifier predicts the exact set of labels associated to the instance.

$$Hamm(\mathbf{h}, S) = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \Delta Z_i|}{|L|} \tag{3}$$

$$Acc(\mathbf{h}, S) = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \tag{4}$$

$$F(\mathbf{h}, S) = \frac{1}{N}\sum_{i=1}^{N}\frac{2|Y_i \cap Z_i|}{|Z_i| + |Y_i|} \tag{5}$$

$$SubAcc(\mathbf{h}, S) = \frac{1}{N}\sum_{i=1}^{N} I(Z_i = Y_i) \tag{6}$$

Measures based on labels are calculated based on false positives f_p, false negatives f_n, true positives t_p and true negatives t_n, *i.e.*, measures $B(t_p, t_n, f_p, f_n)$ can be used in this case. t_{p_l}, t_{n_l}, f_{p_l} and f_{n_l} as true positives, true negatives, false positives and false negatives for each label $l \in L$, the micro and macro versions of these measures are given by Equations 7 and 8, respectively. In this work we used micro and macro versions of the measures $F1$ and AUC (*Area Under ROC curve*).

$$B_{micro}(\mathbf{h}, S) = \frac{1}{|L|}\sum_{i=1}^{|L|} B(t_{p_i}, t_{n_i}, f_{p_i}, f_{n_i}) \tag{7}$$

$$B_{macro}(\mathbf{h}, S) = \frac{1}{|L|} B(\sum_{i=1}^{|L|} t_{p_i}, \sum_{i=1}^{|L|} t_{n_i}, \sum_{i=1}^{|L|} f_{p_i}, \sum_{i=1}^{|L|} f_{n_i}) \tag{8}$$

Ranking based measures used in this work are *One-Error* ($1Err$) and *Ranking Loss* ($RankLoss$), defined respectively by Eqs. 9 and 10[4]. $1Err$ evaluates how many times the top-ranked label is not in the set of relevant labels of the instance; and $RankLoss$ expresses the number of times that irrelevant labels are ranked higher than relevant labels.

$$\begin{aligned} 1Err(\mathbf{h}, S) &= \frac{1}{N}\sum_{i=1}^{N}\delta(\arg\min_{l \in L} r_i(l)) \\ \text{where } \delta(l) &= \begin{cases} 0, & \text{if } l \notin Y_i \\ 1, & \text{otherwise.} \end{cases} \end{aligned} \tag{9}$$

$$\begin{aligned} RankLoss(\mathbf{h}, S) &= \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|Y_i||\overline{Y}_i|}|A(i)| \\ \text{where } A(i) &= \{(l_a, l_b) : r_i(l_a) > r_i(l_b), (l_a, l_b) \in Y_i \times \overline{Y}_i\} \end{aligned} \tag{10}$$

Each described evaluation measure in this section can be estimated using any performance estimation technique, such as *k-fold cross-validation*.

[3] In Eq. 3, Δ represents the symmetric difference between two datasets. In Eq. 6, I(true) = 1 and I(false) = 0.

[4] In these measures, $r_i(l)$ is the ranking predicted for a label l referred to instance $\mathbf{x}_i$, and $\overline{Y}_i$ is the complementary set of Y_i related to the set L.

2.3 Description of Multi-label Learning Methods Used as Benchmark

Some approaches for multi-label learning transform the original problem into binary subproblems, *e.g.* the BR method, or transform the original problem into a single multiclass problem, *e.g.* the LP and SR methods [19]. These three methods are described next. These methods were used as comparison benchmarks because they are the most common methods used in literature and are the closest to our method.

2.3.1 BR — Binary Relevance

One possible solution to a multi-label learning problem is decomposing the original problem into various binary problems. A popular method that works with this type of decomposition is called *Binary Relevance* — BR —, used in [2]. In the BR method, a classifier for each class is constructed using a supervised machine learning, applicable to binary problems. To this end, initially the training dataset S_m is transformed into $|L|$ datasets S_{s_l}, where each dataset corresponds to a label $l_i, i = 1, ..., |L|$. Given a learning algorithm applicable to binary problems, a classifier $\mathbf{h}_l$ is induced using each dataset S_l. To classify a new instance $\mathbf{x}$, $\mathbf{x}$ is given to each classifier $\mathbf{h}_l, l = 1, ..., |L|$. $\mathbf{x}$ is classified with the set of labels for which $\mathbf{h}_l = 1$ (or = true).

2.3.2 LP — Label Powerset

The Label Powerset — LP — method, proposed in [20], transforms the original multi-label problem into a multiclass problem. Each set of labels Y_i in S_m is considered a class of the new multiclass problem. For instance, considering three labels l_1, l_2 and l_3 and a multi-label training dataset S_m, the instance $\mathbf{T}_1 \in S_m$ labeled with $Y_1 = \{l_1, l_2\}$, after the transformation is labeled with $y = l_{1,2}$; the instance $\mathbf{T}_2 \in S_m$ labeled with $Y_1 = \{l_1, l_3\}$, after the transformation is labeled with $y = l_{1,3}$; the instance $\mathbf{T}_3 \in S_m$ labeled with $Y_1 = \{l_1\}$, after the transformation is (still) labeled with $y = l_1$; and so on. With this new dataset S'_s, a multiclass classifier $\mathbf{h}$ is induced.

Given a new instance $\mathbf{x}$ to be labeled, the classifier $\mathbf{h}$ labels $\mathbf{x}$ with a set of labels that have probability higher than a threshold t. Supposing that the output of $\mathbf{h}$ is a probability distribution over all the possible classes, LP method can rank the original labels. For instance, let us consider that $\mathbf{h}$ outputs the following probability distribution: $l_{1,2} = 0.7$, $l_{2,3} = 0.2$ and $l_1 = 0.1$. So, the probability of $\mathbf{x}$ being labeled by $l_1 = 0.7 \times 1 + 0.2 \times 0 + 0.1 \times 1 = 0.8$; being labeled by $l_2 = 0.7 \times 1 + 0.2 \times 1 + 0.1 \times 0 = 0.9$; and being labeled by $l_3 = 0.7 \times 0 + 0.2 \times 1 + 0.1 \times 0 = 0.2$. Defining $t = 0.5$, $\mathbf{x}$ is labeled with the set $Z = \{l_1, l_2\}$.

2.3.3 SR — Select Random

A simple method for transforming a multi-label problem into a multiclass problem consists of replacing, for each instance $\mathbf{T}_i$ of the original dataset S_m, the label Y_i with a single label y randomly selected from Y_i. This simple transformation is called Select Random, described in [19]. For instance, given three labels l_1, l_2 and l_3 and a multi-label training set S_m, the instance $\mathbf{T}_1 \in S_m$ originally labeled with $Y = \{l_1, l_2\}$, after the transformation is labeled with $y = l_1$; the instance $\mathbf{T}_2 \in S_m$ originally labeled with $Y = \{l_1, l_3\}$, after the transformation is labeled with $y = l_2$; the instance $\mathbf{T}_3 \in S_m$ originally labeled with $Y = \{l_1\}$, after the transformation is (still) labeled with $y = l_1$; and so on. This new transformed dataset S'_s is used to induce a multiclass classifier $\mathbf{h}$.

Given a new instance $\mathbf{x}$ to be labeled, the classifier $\mathbf{h}$ labels $\mathbf{x}$ with a set of labels that have probability higher than a threshold t. Supposing that the output of $\mathbf{h}$ is a probability distribution over all the possible classes, the SR method also can rank the original labels. For instance, let us consider that $\mathbf{h}$ outputs the following probability distribution: $l_1 = 0.6$, $l_2 = 0.1$ e $l_3 = 0.3$. Defining $t = 0.2$, $\mathbf{x}$ is labeled with the set $Z = \{l_1, l_3\}$.

3 Our Proposed Method: RB — Random-Bagging

The RB method, proposed in this work, is based in the SR and Bagging methods. SR weakness is related to the loss of information due to absence of some (or many) labels in the induction of the classifier $\mathbf{h}$. To solve this problem, the SR transformation can be repeated C times and the induced classifiers can be combined. Our proposed method basically consists of these two steps.

A classical method that combines classifiers is Bagging [14]. The Bagging method combines classifiers' decisions by voting. The combined classifiers are induced using bootstrap samples of the original dataset S_m. In our work, instead of selecting the instances randomly with replacement, we randomly select labels for the instances.

Our proposed RB method consists of two steps: (1) induction of the component multiclass classifiers to compose the multi-class classifier and (2) combination of the component classifiers to predict new instances. Note that C and t are parameters of our method.

(1) Multi-label classifier construction: The input to RB is a dataset $S_m = \{(\mathbf{x}_i, Y_i), i = 1, ..., N\}$. From S_m, C datasets $S_{s_c}, c = 1, ..., C$ are constructed. Each dataset S_{s_c} is composed by all instances $\mathbf{x}_i \in S_m$, and each instance $\mathbf{x}_i$ is labeled with a single $y_i \in Y_i$, randomly selected with replacement. So, each dataset S_{s_c} is used to induce a multiclass classifier using a supervised machine learning algorithm.

(2) Instances prediction: The classifiers constructed in Step 1 offer the most probable label and a probability distribution over the possible labels that the classifier can predict[5] of the possible labels the classifier can predict. Given a new instance $\mathbf{x}$ to be classified, for each label $l \in L$ we calculate the mean of probability distribution offered for each classifier $\mathbf{h}_c$ — $\gamma_l = \frac{1}{C} \sum_{c=1}^{C} p(c|\mathbf{x})$. Finally, the instance $\mathbf{x}$ is labeled with labels for which γ_l is higher then a threshold t — $\gamma_l \geq t$.

To evaluate the RB method, we implemented RB using Mulan library [16] and Weka tools [17]. Weka is a free computational solution developed to aid the data mining process. Weka includes tools to support supervised and unsupervised learning, and other tasks. This solution has the interesting property of being implemented in Java, allowing portability. The Mulan library (Multi-label Learning) was proposed to attend the needs and specificities of multi-label problems. The Mulan library was implemented based on Weka. Mulan was used to support RB implementation.

4 Experiments and Results

Our experiments aim to (i) evaluate the prediction performance of our proposed method RB; (ii) evaluate the prediction performance of SR, since we did not find any results about this method; (iii) compare both methods to BR and LP. To induce the base (binary and multiclass) classifiers of all methods, we used three different machine learning algorithms implemented within Weka: J48 — the C4.5 algorithm for induction of decision trees [21], implemented on Weka; NB — the Naive Bayes algorithm [18], which uses bayesian statistics for classifier induction; and SMO — an algorithm that efficiently solves the optimization problem for inducing SVMs (Support Vector Machines) [22]. Six natural datasets were used in our experiments[6]: Emotions, Genbase, Scene, Yeast, Enron e Medical. Table 1 describes characteristics of these datasets, where #Inst. is the number of instances in the dataset; #Feat. Disc and #Feat. Cont. are, respectively, number of discrete and continuous features; #Labels is the total number of labels; $Card$ is the label cardinality value — Eq. 1; and $Dens$ is the label density value — Eq. 2.

RB behavior was evaluated using different number of component classifiers — $C = |L|$ e $C = 10|L|$ —, and different threshold values — 0.1, 0.2, 0.3, 0.4, 0.5, 0.7 and 0.9. All evaluation measures described in Section 2.2 were used. *k-fold cross-validation*, with $k = 10$, was used to evaluate each multi-label method behavior. Figures 1 to 10 show the results of 10-*fold cross-validation* for BR, LP, SR with threshold values $t = 0.1, 0.2, 0.3, 0.4, 0.5, 0.7, 0.9$, and RB with all combination of C and t listed, using all measures described in Section 2.2. Figures 1 to 4 show results using example based measures; Figures 5 to 8

[5] Algorithms that offers pertinence degree to each label as output should also be used in this work. However, we did not considered these algorithms in our experiments.

[6] These datasets and others are available at Mulan library site — `http://mulan.sourceforge.net/datasets.html`

Dataset	#Inst.	#Feat. Disc.	#Feat. Cont	#Labels	$Card$	$Dens$
Yeast	2417	0	103	14	4.237	0.303
Scene	2407	0	294	6	1.074	0.179
Emotions	593	0	72	6	1.869	0.311
Genbase	662	1186	0	27	1.252	0.046
Enron	1000	1001	0	53	3.378	0.064
Medical	978	1449	0	45	1.245	0.028

Table 1: Datasets Characteristics

show results using label based measures; and Figures 9 and 10 show results using ranking based measures. Figures showing SMO results do not have results for the Enron dataset using RB method with $C = 10|L|$ and the SMO base learner because we could not execute these experiments due to the high computational cost. In what follows, we analyze each group of figures, corresponding to each evaluation metric.

4.1 Analyzing Performance Methods per Measure

In what follows we describe the results obtained by analyzing each measure. To make it easier to interpret the results, we indicate with $\downarrow$ the measures for which the lower the value of the measure, the higher the performance of the method; and we indicate with $\uparrow$ the measures that the higher the values of the measure, the higher the performance of the method.

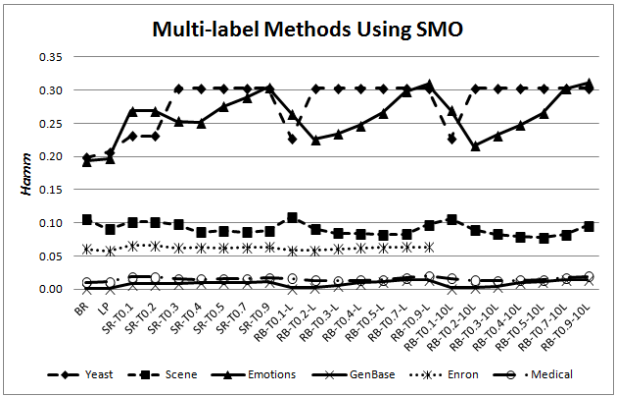
Hamming Measure ($\downarrow$): Figures 4.1, 4.1 and 4.1 show the performance of multi-label methods performance for the $Hamm$ measure. The plots show that, for the Scene dataset, the results are quite different when varying the base learner algorithm (SMO, J.48 or NB); for all the other datasets, the results are similar. They also show that the datasets with lowest density — Genbase, Enron and Medical — have the best results considering this measure for all multi-label and base learners. It is also interesting to notice that BR with the NB base learner obtained a much poorer result for the Enron dataset than the other learning algorithm.

Accuracy Measure ($\uparrow$): Figures 4.1, 4.1 and 4.1 show the performance of multi-label methods for the Acc measure. These plots show that, when using the NB base learner (Fig. 4.1), the results obtained with all multi-label learners were the poorest results, except for the Scene dataset, for which the results are similar, considering the variation of base learners.

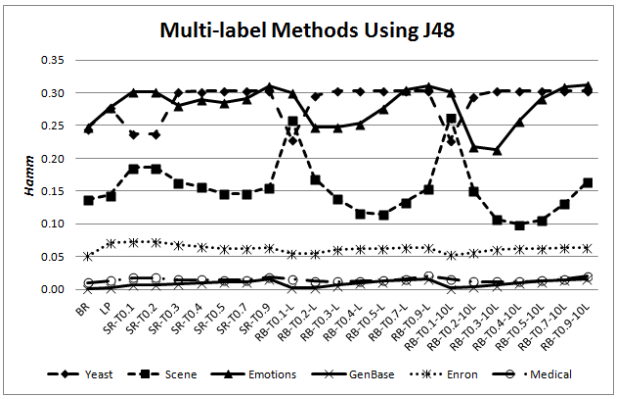
F Measure ($\uparrow$): Figures 4.1, 4.1 and 4.1 show the performance of multi-label methods for the F measure. These plots show that the results are similar when varying the base learner, except for the GenBase dataset, for which the multi-label learners decreased their performance with the NB base learner. It is interesting to notice that RB with $t = 0.1$ and the J.48 base learner shows improvement compared to the other multi-label learners for Emotions dataset, but for the same method — RB and J.48 base learner — the best results were obtained for $t = 0.2$ and $t = 0.3$ for the Scene dataset.

Subset Accuracy Measure ($\uparrow$): Figures 4.1, 4.1 and 4.1 show the performance of multi-label methods for the $SubAcc$ measure. This measure is very important to analyze the ability of multi-label predictors to classify the entire label set. We can observe that poor results were obtained when considering Emotions, GenBase and Yeast datasets. For this measure the NB base learner obtained the poorest results.

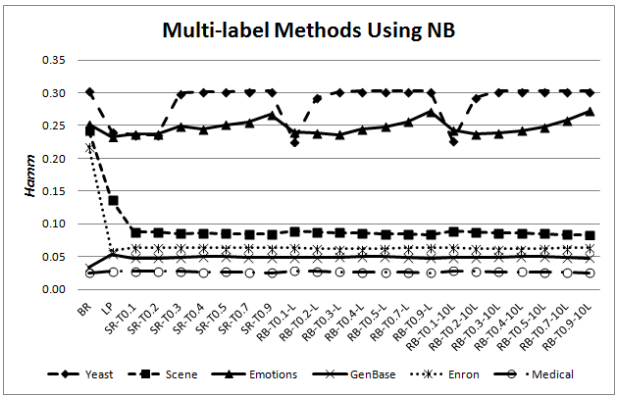
Micro Version of AUC ($\uparrow$): Figures 4.1, 4.1 and 4.1 show the performance of multi-label methods for the AUC_{micro} measure. All multi-label learners show good results for this measure, regardless of the base learner. The exception is the LP method, showing poorer results when using the SMO base learner than the other multi-label and base learners. For the other multi-label learners, the results are similar when using SMO and NB, with a subtle difference when using J.48.



(a) $Hamm$ using SMO

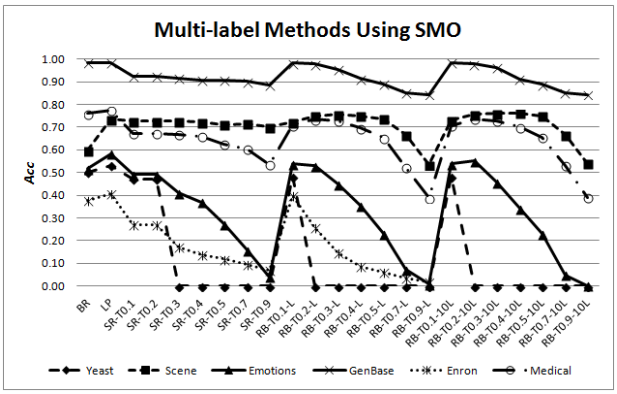


(b) $Hamm$ using J48

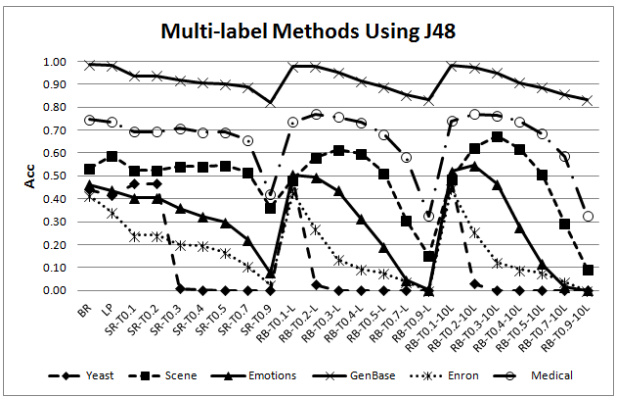


(c) $Hamm$ using NB

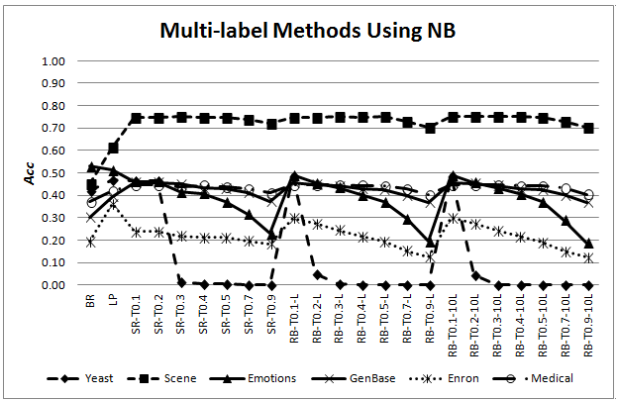
Figure 1: Results in all Test Scenarios Considering $Hamm$ measure



(a) *Acc* using SMO

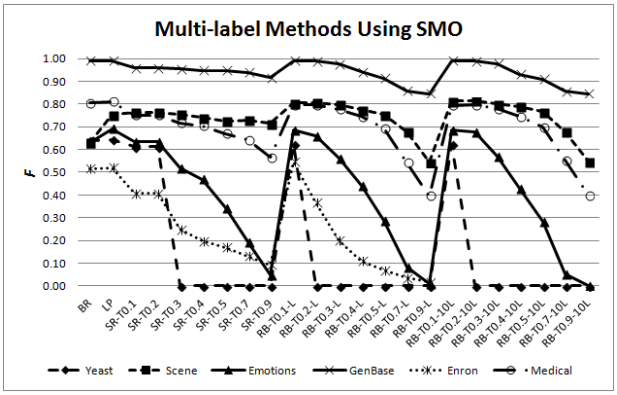


(b) *Acc* using J48

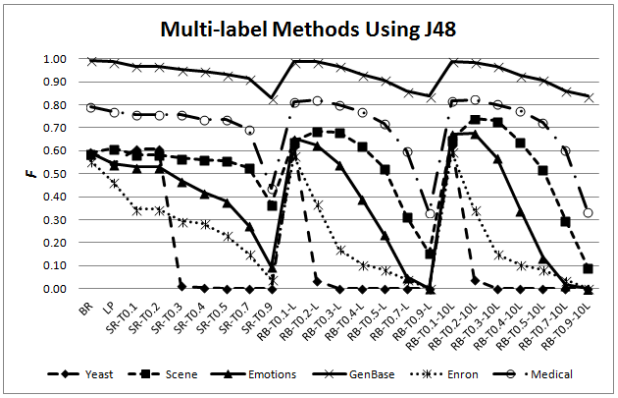


(c) *Acc* using NB

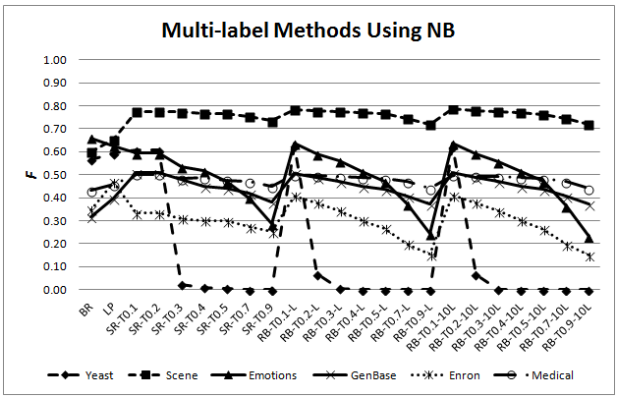
Figure 2: Results in all Test Scenarios Considering *Acc* measure



(a) F using SMO

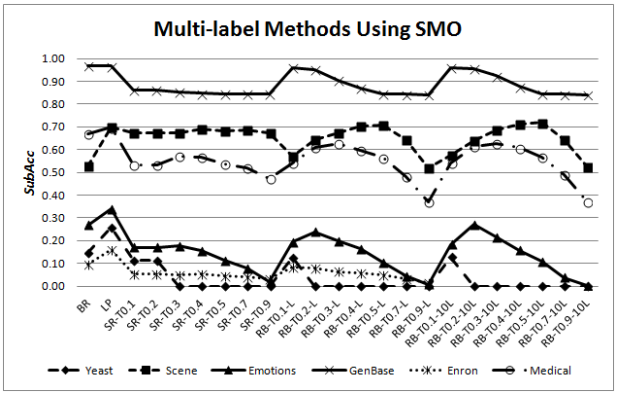


(b) F using J48

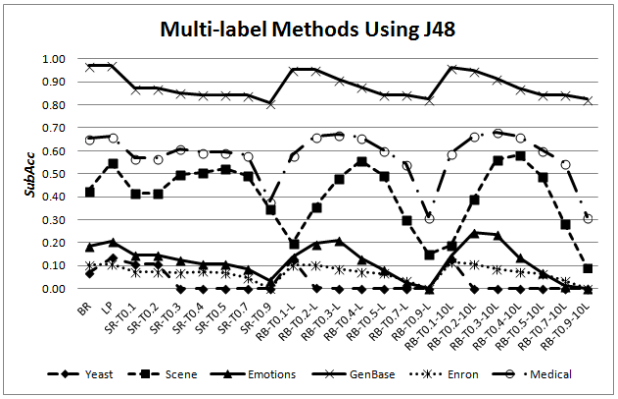


(c) F using NB

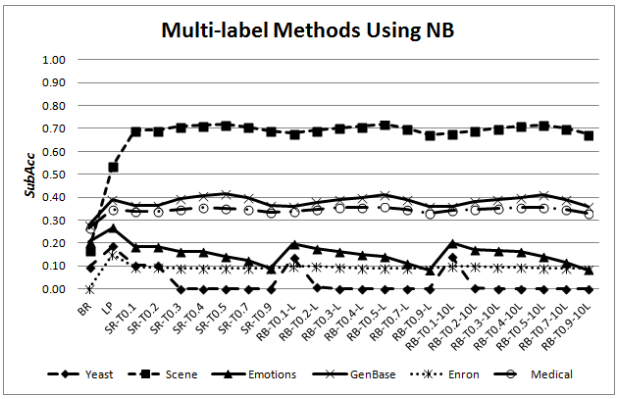
Figure 3: Results in all Test Scenarios Considering F measure



(a) $SubAcc$ using SMO

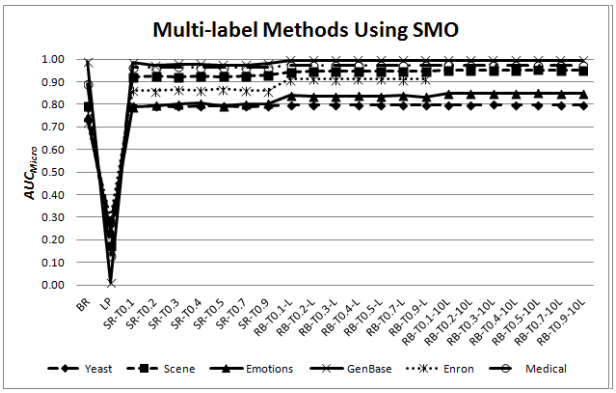


(b) $SubAcc$ using J48

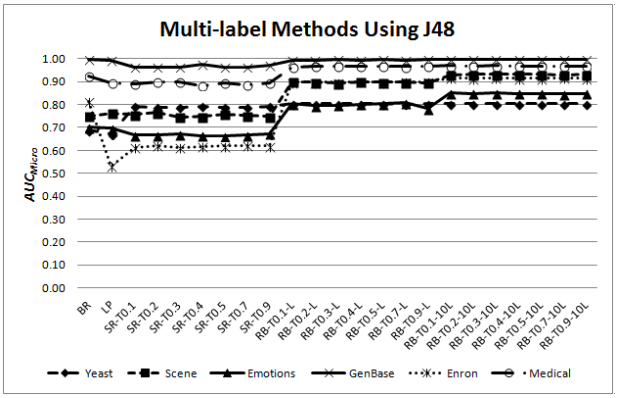


(c) $SubAcc$ using NB

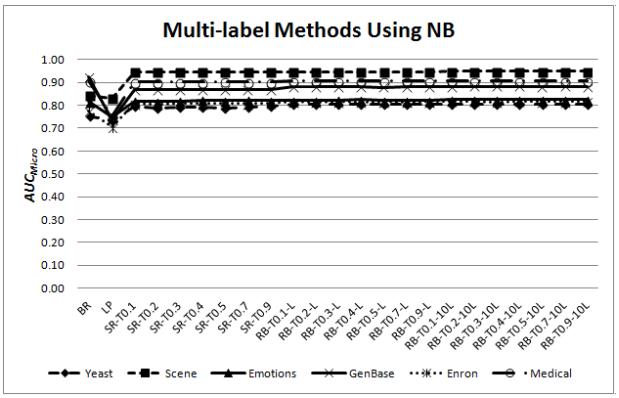
Figure 4: Results in all Test Scenarios Considering $SubAcc$ measure



(a) AUC_{micro} using SMO



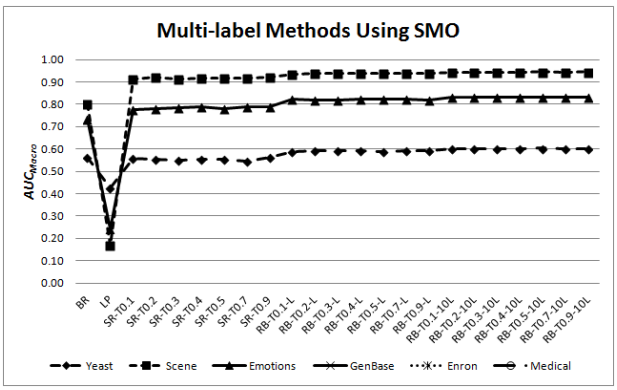
(b) AUC_{micro} using J48



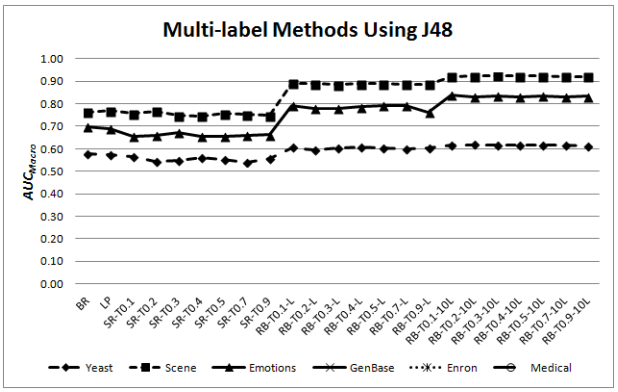
(c) AUC_{micro} using NB

Figure 5: Results in all Test Scenarios Considering AUC_{micro} measure

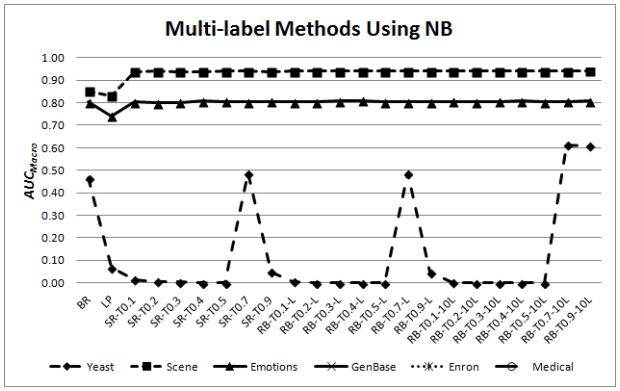
Macro Version of AUC (↑): Figures 4.1, 4.1 and 4.1 show the performance of multi-label methods the AUC_{macro} measure. Unexpectedly, the results for AUC_{macro} are quite different from AUC_{micro}, except when using LP with SMO: the results are very poor also for the macro version of AUC. In general, the results of this version of AUC are worse than the results of the micro version of AUC. It is surprising that for Medical and Enron the results are so poor. Using NB base learner, the results are still much poorer than using the other base learners.



(a) AUC_{macro} using SMO



(b) AUC_{macro} using J48



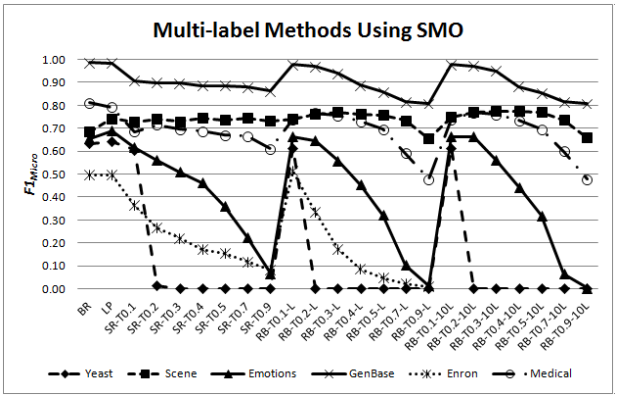
(c) AUC_{macro} using NB

Figure 6: Results in all Test Scenarios Considering AUC_{macro} measure

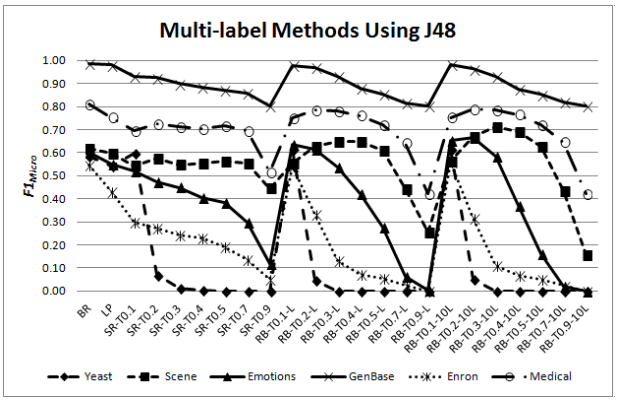
Micro Version of $F1$ (↑): Figures 4.1, 4.1 and 4.1 show the performance of multi-label methods for the $F1_{micro}$ measure. SMO and J.48 learners show similar results. Once more, NB shows poorest results than the other learners, especially for the GenBase dataset, for which the drop in performance is visible.

Macro Version of $F1$ (↑): Figures 4.1, 4.1 and 4.1 show the performance of multi-label methods for the $F1_{macro}$ measure. The SMO and J.48 learners show similar results. Again, NB shows poorer results than the other learners, especially for the GenBase dataset, for which the drop in performance is visible.

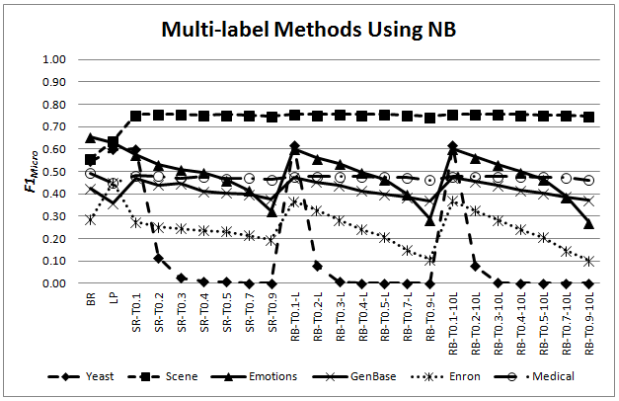
One Error (↓) and RankLoss (↓): Figures 4.1, 4.1 and 4.1 show the performance of multi-label methods for the $1Err$ measure; and Figures 4.1, 4.1 and 4.1 show the performance of multi-label methods for the $RankLoss$ measure. We show these results together because they are very similar. Only for these measures the NB learner shows better results than SMO and J.48 in general. LP multi-label learner shows the poorest results, compared to the other multi-label learners. RB and SR show similar results compared to BR, but using J.48 base learners and for some datasets we notice improvement of RB compared to SR — Figs. 4.1 and 4.1.



(a) $F1_{micro}$ using SMO

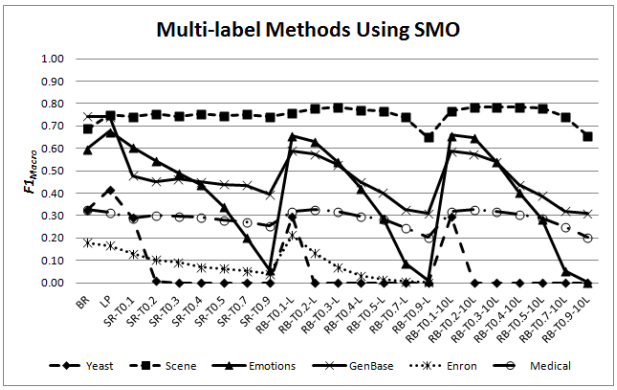


(b) $F1_{micro}$ using J48

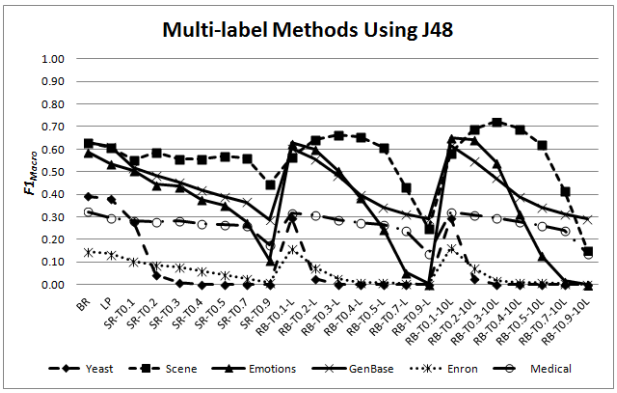


(c) $F1_{micro}$ using NB

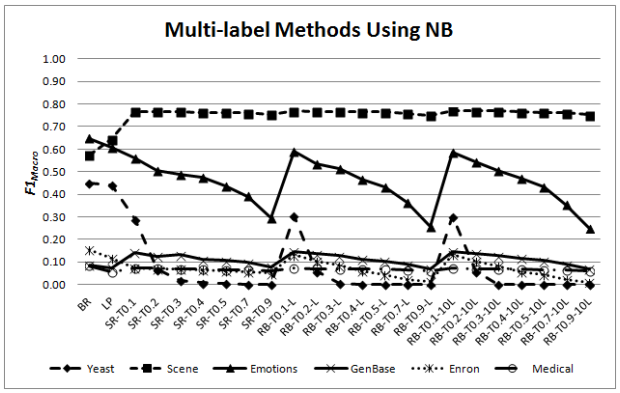
Figure 7: Results in all Test Scenarios Considering $F1_{micro}$ measure



(a) $F1_{macro}$ using SMO

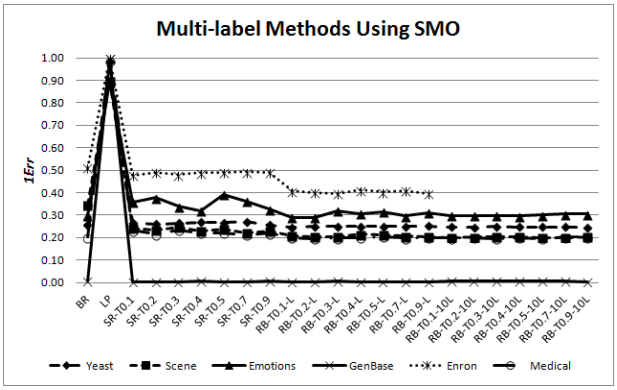


(b) $F1_{macro}$ using J48

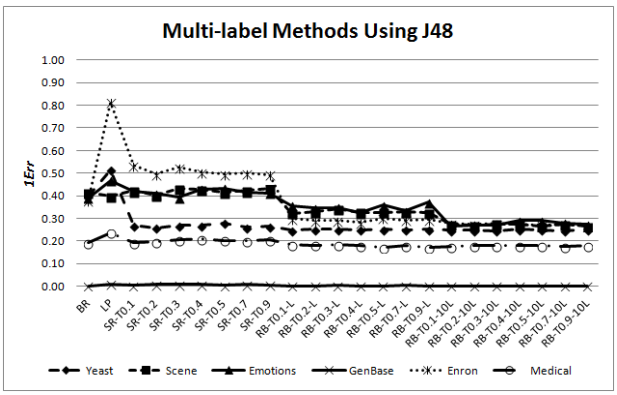


(c) $F1_{macro}$ using NB

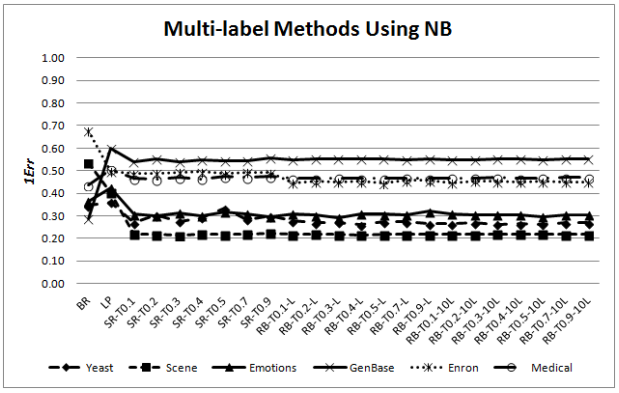
Figure 8: Results in all Test Scenarios Considering $F1_{macro}$ measure



(a) $1Err$ using SMO



(b) $1Err$ using J48



(c) $1Err$ using NB

Figure 9: Results in all Test Scenarios Considering $1Err$ measure

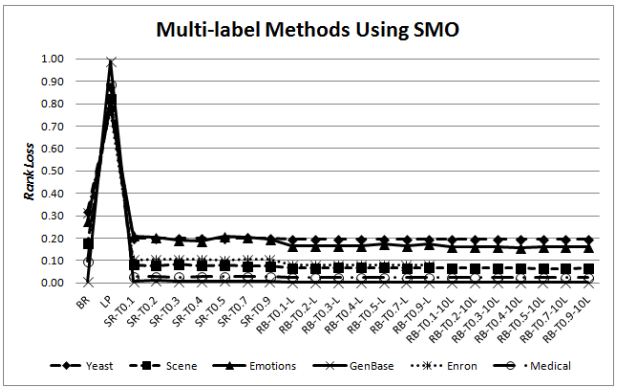
We can observe in these plots that all experiments using SR and RB with $t = 0.7$ and $t = 0.9$ shows the worst results when compared to the other values of t. So, results using $t = 0.7$ and $t = 0.9$ were not considered in our hypotheses tests, described next. Also, we can conclude, from all these graphics and analyses, that the NB base learner does not lead to very good results when compared to the other base learners and multi-label methods on these datasets.

4.2 Hypotheses Tests

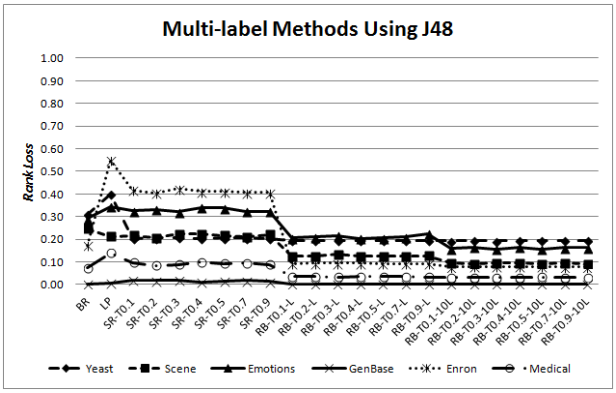
To check for a significant statistical difference among the multi-label methods, we considered different null hypotheses to analyze the behavior of each variable t and C for RB, the behavior of t for SR, and we compared RB with the three other methods. We executed the Wilcoxon test[7] when the null hypothesis is relative to two variables — comparison of a metric using two different variables —, and Friedman[8] when the null hypothesis is relative to more than two variables. For the hypotheses tests, we considered the results obtained using 10-*fold cross-validation*. Each execution of the BR and LP methods using the three learning algorithms — J48, NB or SMO — was considered an independent execution. Also, each combination of the three learning algorithms and the different values of t was considered different executions of the SR method, and we called each execution SR-T0.1-J48, SR-T0.2-SMO, and so on. Finally, each combination of the three learning algorithms with the different values of t and different numbers of classifiers C was considered an independent executions of the RB method, and we called each execution RB-T0.1-L-J48, RB-T0.2-10L-SMO, and so on. In what follows, we describe each hypotheses test and the obtained results.

[7]Wilcoxon test is a non-parametric alternative to comparison of two learning algorithms [23].

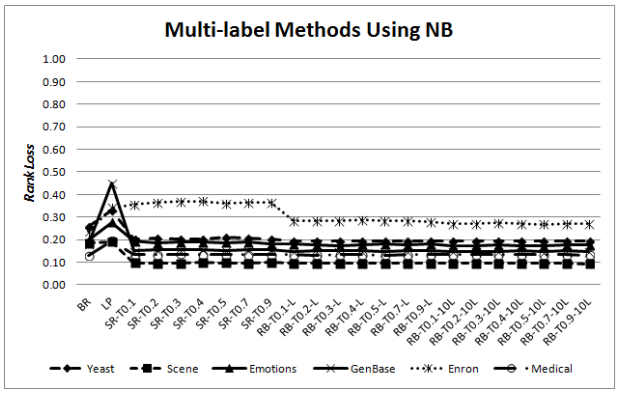
[8]Friedman test is a non-parametric alternative for ANOVA test [23].



(a) $RankLoss$ using SMO



(b) $RankLoss$ using J48



(c) $RankLoss$ using NB

Figure 10: Results in all Test Scenarios Considering $RankLoss$ measure

Hypothesis H1: The results obtained for RB method are comparable considering $C = |L|$ and $C = 10|L|$. The Wilcoxon test was executed, rejecting the hypothesis with a confidence level of 95% for measures F, $SubAcc$, $1Err$ and $RankLoss$, and RB considering $C = |L|$ shows the best results to these methods and these datasets. On the other hand, considering AUC_{Micro} measure, the null hypothesis was also rejected, but RB considering $C = 10|L|$ shows the best results. For all the other measures, this null hypothesis was not rejected. Because RB considering $C = 10|L|$ shows the best result for only one measure in 10 (ten), and RB considering $C = L$ shows the best result for 4 (four) measures in 10 (ten), and considering that the computational cost of RB with $C = 10|L|$ is much higher than RB with $C = |L|$, we considered that our method RB shows the best results with $C = |L|$ in our experimental scenarios, and used this result for RB in the following hypotheses tests.

Hypothesis H2: The SR and RB (with $C = |L|$) methods are comparable. The Wilcoxon test was also executed, rejecting the null hypothesis with a confidence level of 95% for the measures $Hamm$, AUC_{Micro}, $1Err$ and $RankLoss$, and the RB method shows the best results for all this 4 (four) measures. For all the other measures, the null hypothesis was not rejected. Considering that RB shows the best result for 4 (four) measures in 10 (ten), we considered that the RB method shows better results than SR in our experimental scenarios, and used this result in the following hypothesis tests.

Hypothesis H3: RB with $t = 0.1$, $t = 0.2$, $t = 0.3$, $t = 0.4$ and $t = 0.5$ are comparable using $C = |L|$. In this case, we renamed the RB method considering all values of t to make the comparison analysis more readable — RB-T01, RB-T02, RB-T03, RB-T04 and RB-T05. The Friedman test was executed, which rejected the null hypothesis for measures Acc, F, $SubAcc$, $F1_{Mic}$ and $F1_{Mac}$ with a confidence level of 95%, and for measure $1Err$ with a confidence level of 90% . Figures 4.2 to 4.2 show the obtained results to Nemenyi post-hoc test for these 6 (six) measures. We can observe in these figures that the RB method for $t = 0.1$ and $t = 0.2$ are the best ranked for all 6 (six) measures, but only considering the F measure there is significant statistical difference. Considering these results, we established that RB with $t = 0.1$ and $t = 0.2$ give the best results, and these values of t were selected for the following hypothesis test H3.

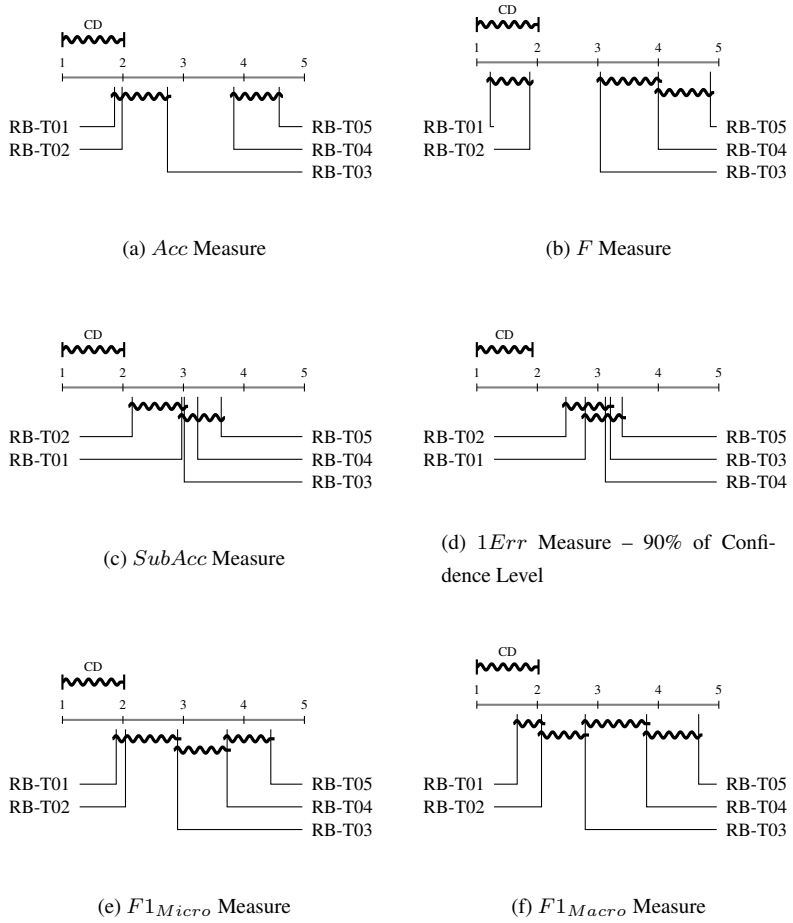
Hypothesis H4: The LP, BR and RB methods are comparable, considering two scenarios of RB execution: (i) $C = |L|$ and $t = 0.1$ — RB-T01-L — and (ii) $C = |L|$ and $t = 0.2$ — RB-T02-L. The Friedman test was executed, rejecting the null hypothesis with a confidence level of 95% for measures F, $Hamm$, $SubAcc$, AUC_{Micro}, $1Err$ and $RankLoss$. Figures 4.2 to 4.2 shows the results of Nemenyi post-hoc test for these measures. In these figures, we can observe that for 3 (three) out of 6 (six) measures — AUC_{Micro}, $1Err$ e $RankLoss$ —, the BR method is better ranked, but there is not significant statistical difference, and for 2 (two) measures — F e $SubAcc$ —, LP shows the best results with significant statistical difference. So, RB can show better results than BR and LP in some cases.

4.3 Correlations among Multi-label Model Predictions and Cardinality and Density of Datasets

In this section, we aim to analyze if there is some relation between the cardinality $Card$, inherent to each multi-label dataset, and the measure values obtained for each multi-label learning method and each dataset, as well as if there is some relation between the density $Dens$ and the measure values. To compute the correlation, we considered that $Card$ and $Dens$ are variables, and the correlation was calculated between each of them and each of the evaluation measures. Table 2 shows the sum of the number of times that the correlation between $Card$ and each evaluation measure obtained with each multi-label learning method using SMO, J.48 and NB as base learning algorithms is high — higher than 0.7 or lower than -0.7. Similarly, Table 3 shows the sum of the number of times that the correlation between $Dens$ and each measure values obtained with each multi-label learning method using SMO, J.48 and NB as base learning algorithms is high — higher than 0.7 or lower than -0.7.

We expected to find high correlation for the $SubAcc$ measure, and Table 2 shows high correlation between $SubAcc$ measure and $Card$ for all multi-label learning methods — and, in this case, the higher the $Card$ values, the lower the $SubAcc$ values. Further, the same occurred with AUC_{macro} — high correlation with $Card$ for all multi-label learning methods. Surprisingly, Table 2 shows that LP, RB and SR are sensible to $Card$ for all the other measures, except for the $Hamm$ and $1Err$ measures, what does not occur with the BR method. In all cases, the higher the $Card$, the worse the measurement values.

$Hamm$ was the only one measure which exhibited high correlation with $Dens$ for all methods and all base learners in



(a) Acc Measure

(b) F Measure

(c) $SubAcc$ Measure

(d) $1Err$ Measure – 90% of Confidence Level

(e) $F1_{Micro}$ Measure

(f) $F1_{Macro}$ Measure

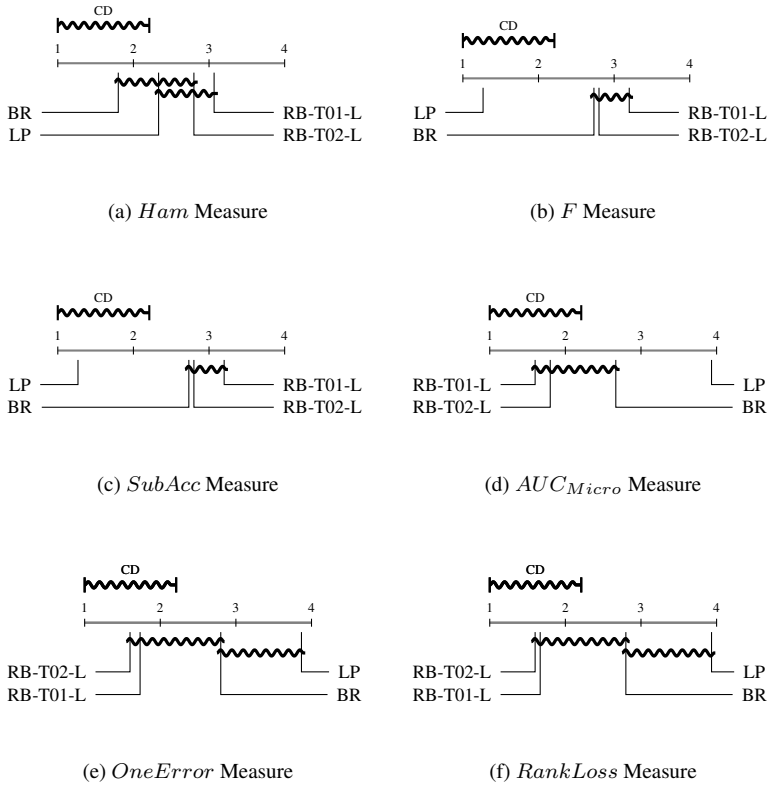
Figure 11: Post-Hoc Hypothesis Test H3

$Card$	Hamm	Acc	F	SubAcc	F1Micro	AUCMicro	F1Macro	AUCMacro	1Err	RankLoss
BR	0/3 (0.0%)	0/3 (0.0%)	0/3 (0.0%)	3/3 (100.0%)	0/3 (0.0%)	1/3 (33.3%)	0/3 (0.0%)	3/3 (100.0%)	0/3 (0.0%)	2/3 (66.7%)
LP	0/3 (0.0%)	1/3 (33.3%)	1/3 (33.3%)	3/3 (100.0%)	1/3 (33.3%)	2/3 (66.7%)	0/3 (0.0%)	3/3 (100.0%)	0/3 (0.0%)	1/3 (33.3%)
SR	0/21 (0.0%)	17/21 (81.0%)	17/21 (81.0%)	21/21 (100.0%)	18/21 (85.7%)	13/21 (61.9%)	12/21 (57.1%)	21/21 (100.0%)	0/21 (0.0%)	2/21 (9.5%)
RB	6/42 (14.3%)	33/42 (78.6%)	34/42 (81.0%)	34/42 (81.0%)	35/42 (83.3%)	35/42 (83.3%)	22/42 (52.4%)	42/42 (100.0%)	0/42 (0.0%)	21/42 (50.0%)

Table 2: Correlation Between Cardinality and Each Measure — Sum of Number of Times, Considering SMO, J.48 and NB, (and Percentage) when Correlation is High (> 0.7 or < -0.7).

$Dens$	Hamm	Acc	F	SubAcc	F1Micro	AUCMicro	F1Macro	AUCMacro	1Err	RankLoss
BR	3/3 (100.0%)	1/3 (33.3%)	1/3 (33.3%)	0/3 (0.0%)	1/3 (33.3%)	1/3 (33.3%)	1/3 (33.3%)	1/3 (33.3%)	0/3 (0.0%)	1/3 (33.3%)
LP	3/3 (100.0%)	0/3 (0.0%)	1/3 (33.3%)	0/3 (0.0%)	1/3 (33.3%)	0/3 (0.0%)	1/3 (33.3%)	1/3 (33.3%)	1/3 (33.3%)	0/3 (0.0%)
SR	21/21 (100.0%)	0/21 (0.0%)	0/21 (0.0%)	0/21 (0.0%)	0/21 (0.0%)	7/21 (33.3%)	0/21 (0.0%)	14/21 (66.7%)	7/21 (33.3%)	7/21 (33.3%)
RB	42/42 (100.0%)	8/42 (19.0%)	7/42 (16.7%)	8/42 (19.0%)	6/42 (14.3%)	28/42 (66.7%)	0/42 (0.0%)	23/42 (54.8%)	21/42 (50.0%)	28/42 (66.7%)

Table 3: Correlation Between Density and Each Measure — Sum of Number of Times, Considering SMO, J.48 and NB, (and Percentage) when Correlation is High (> 0.7 or < -0.7).



(a) Ham Measure

(b) F Measure

(c) $SubAcc$ Measure

(d) AUC_{Micro} Measure

(e) $OneError$ Measure

(f) $RankLoss$ Measure

Figure 12: Post-Hoc Hypothesis Test H4

all experimentation scenarios. In fact, observing Figures 4.1, 4.1 and 4.1, there are not many overlaps between the curves, except considering Medical and GenBase datasets, and both have $Dens$ values very similar — 0.028 and 0.064, respectively. Surprisingly, the lower the $Dens$, the better the $Hamm$ results. Other measures show some correlation with the RB method, but it is more sparse — AUC_{micro}, AUC_{macro}, $1Errr$ and $RankLoss$.

4.4 Analysis of C Lower than $|L|$

To analyze values of C when it is lower than L for the RB method, we realized experiments with RB using $t = 0.2$, the J.48 base learning algorithm and values of C in the set $\{0.1|L|, 0.3|L|, 0.5|L|, 0.7|L|, 0.9|L|\}$. We want to analyze if there is improvement in the measure values when C increases. For this analysis, we also calculated Pearson correlation between the number of constructed models and the measurement values obtained. Figure 13 shows the correlation values from the measures' perspective, and Figure 14 shows the correlation values from the datasets' perspective. In both figures, correlations (bars) bellow the bottom dot line indicate high negative correlations, *i.e.*, the higher the number of models, the lower the measure value; correlations (bars) above the upper dot line indicate high positive correlations, *i.e.*, the higher the number of models, the higher the measure value.

Figure 13 shows that, except for Yeast and Scene datasets, $Hamm$ decreases when increasing the number of models, and Emotions, GenBase, Enron and Medical datasets have higher numbers of labels than Yeast and Scene (Table 1). Also, AUC_{micro}, $1Err$ and $RankLoss$ are measures that shows better results when incrementing the number of C in RB. The other measures in general have improvements when increasing C, but there is some exceptions.

Figure 14 shows that the Emotions dataset is impacted positively on increasing C because correlation is negatively high for $Hamm$, $1Err$ and $RankLoss$, as it should be. On the other hand, for the Yeast dataset there is a negative impact when increasing C. The other datasets have mixed positive and negative impacts.

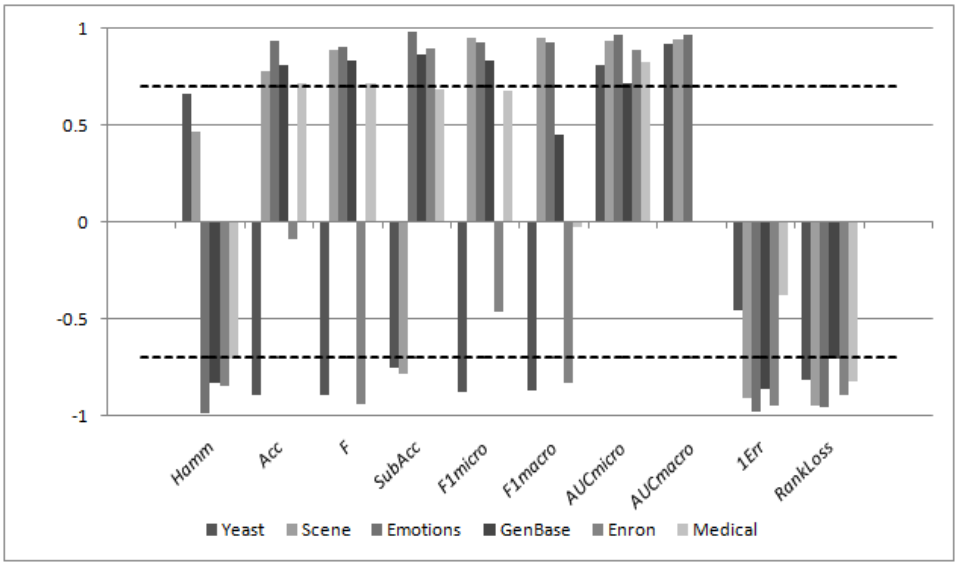


Figure 13: Correlation Values from Measure Perspective.

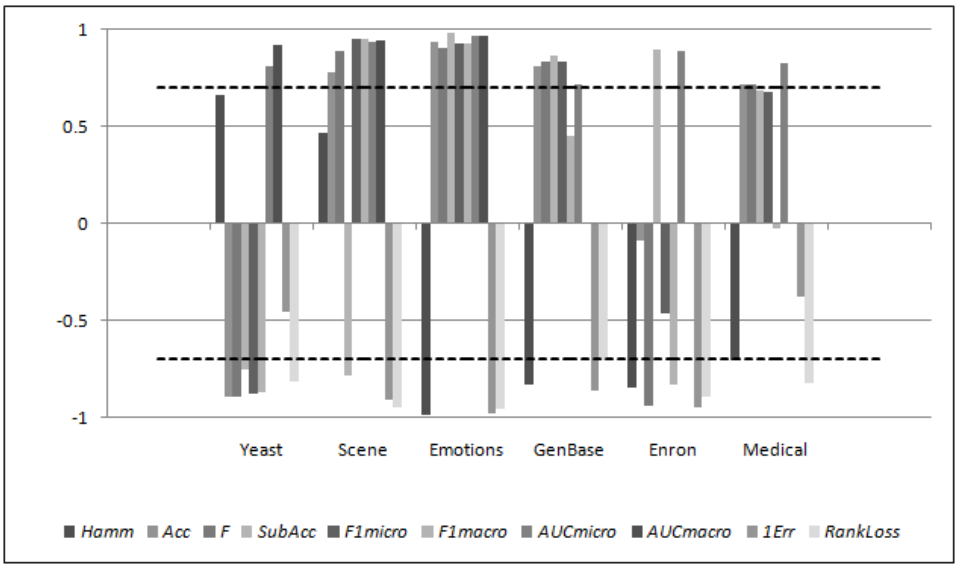


Figure 14: Correlation Values from Dataset Perspective.

5 Conclusions and Future Work

In this work, we propose a method for construction of multi-label classifiers, named RB — Random-Bagging. The RB method is based on the simple SR method for multi-label problems and on the Bagging method for combination of classifiers. The method was implemented using the Mulan library and the Weka tools, on Java language. 6 (six) datasets were used as benchmarks to evaluate our proposed method, as well as to evaluate the SR method, and to compare both methods to LP and BR, benchmark methods used for multi-label learning. We could observe that RB shows better results than the other methods — BR, LP and SR — considering some evaluation measures used for multi-label classifiers.

Correlation measurements obtained in this work indicate that methods that take into account $Card$ and $Dens$ may lead to better results when multi-label datasets with high values of $Card$ and very low values of $Dens$ are available, since in general the correlation occurs when the results are worse for increment of $Card$ values and/or decrement of $Dens$ values.

In our experiments we could also observe that, in general, there are many positive gains when approximating the number of base classifiers of RB to the number of labels existing in a dataset. We believe that this occurs because in this case all labels may occur in some training dataset. On the other hand, turning the number of models of RB very large does not bring a high improvement to the method that justifies the computational cost. So, RB using number of models equals to the number of labels of the dataset could be a default choice.

In future work, we intend to investigate methods to construct multi-label classifiers reducing $Dens$ to induce better classifiers. Also, we also intend to investigate further the relation between the measures $Dens$ and $Card$ and the performance of the classifiers.

Acknowledgements

The authors thank Prof. Ronaldo Cristiano Prati (UFABC) and Prof. Alexandre Plastino (UFF) for their valuable observations and contributions, and to Jean Metzz (USP), for his help in executing our hypotheses tests.

References

[1] R. E. Schapire and Y. Singer. “BoosTexter: a boosting-based system for text categorization”. *Machine Learning*, vol. 39, no. 2–3, pp. 135–168, 2000.

[2] X. Shen, M. Boutell, J. Luo and C. Brown. “Multi-label machine learning and its application to semantic scene classification”. In *Proc. 2004 Int. Symposium on Electronic Imaging – EI 2004*, pp. 18–22, 2004.

[3] F. Sebastiani. “Machine learning in automated text categorization”. *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[4] A. Dimou, G. Tsoumakas, V. Mezaris, I. Kompatsiaris and I. Vlahavas. “An Empirical Study Of Multi-Label Learning Methods For Video Annotation”. In *7th Int. Workshop on Content-Based Multimedia Indexing, IEEE*, pp. 19–24, 2009.

[5] F. Bernardini, A. Garcia and I. Ferraz. “Artificial intelligence based methods to support motor pump multi-failure diagnostic”. *Engineering Intelligent Systems*, vol. 17, no. 2, 2009.

[6] K. N. Calembo, F. C. Bernardini and C. B. Martins. “Proposta de um método de combinação de classificadores para construção de classificadores multi-rótulo”. In *Conferência Latinoamericana de Informática — CLEI’2011*, 2011.

[7] E. Spyromitros-Xioufis, M. Spiliopoulou, G. Tsoumakas and I. Vlahavas. “Dealing with Concept Drift and Class Imbalance in Multi-label Stream Classification”. In *Proc. 22nd International Conference on Artificial Intelligence (IJCAI 2011)*, pp. 1583–1588, Barcelona, Spain, 2011. AAAI press. `http://ijcai.org/papers11/Papers/IJCAI11-266.pdf`.

[8] A. Clare and R. D. King. "Knowledge discovery in multi-label phenotype data". In *LNCS*, edited by L. D. Raedt and A. Siebes, volume 2168, pp. 42–53, Berlin, 2001. Springer.

[9] F. D. Comité, R. Gilleron and M. Tommasi. "Learning multi-label altenating decision tree from texts and data". In *LNCS*, edited by P. Perner and A. Rosenfeld, volume 2734, pp. 35–49, Berlin, 2003. Springer.

[10] N. Ghamrawi and A. McCallum. "Collective multi-label classification". In *Proc. 14th ACM Int. Conf. on Information and Knowledge Management*, pp. 195–200, Bremen, Germany, 2005.

[11] S. Zhu, X. Ji, W. Xu and Y. Gong. "Multi-labelled classification using maximum entropy method". In *Proc. 28th Int. ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 274–281, 2005.

[12] W. Cheng and E. Hüllermeier. "Combining instance-based learning and logistic regression for multilabel classification". *Machine Learning*, vol. 76, no. 2–3, pp. 211–225, 2009.

[13] S. Godbole and S. Sarawagi. "Discriminative methods for multi-labeled classification". In *LNAI*, edited by H. Dai, R. Srikant and C. Zhang, volume 3056, pp. 22–30, Berlin, 2004. Springer.

[14] L. Breiman. "Bagging Predictors". *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[15] R. Bryll, R. Gutierrez-Osuna and F. Quek. "Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets". *Pattern Recognition*, vol. 36, pp. 1291–1302, 2003.

[16] G. Tsoumakas, J. Vilcek, E. Spyromitros and I. Vlahavas. "Mulan: A Java Library for Multi-Label Learning". *Journal of Machine Learning Research*, vol. 12, pp. 2411–2414, 2010.

[17] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, second edition, 2005.

[18] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[19] G. Tsoumakas, I. Katakis and I. Vlahavas. *Data Mining and Knowledge Discovery Handbook*, chapter Mining Multi-label Data. Springer, second edition, 2010.

[20] J. Read. "A pruned problem transformation method for multi-label classification". In *Proc. 2008 New Zealand Computer Science Research Student Conference (NZCSRS 2008)*, pp. 143–150, 2008.

[21] J. Quinlan. *C4.5 Programs for Machine Learning*. Morgan Kaufmann Publishers, 1988.

[22] J. C. Platt. *Advances in kernel methods: support vector learning*, chapter Fast training of support vector machines using sequential minimal optimization, pp. 185–208. MIT Press, 1999.

[23] J. Demšar. "Statistical Comparisons of Classifiers over Multiple Data Sets". *J. Machine Learning Research*, vol. 7, pp. 1–30, 2006.