Learning and Nonlinear Models - Revista da Sociedade Brasileira de Redes Neurais (SBRN), Vol. XX, No. XX, pp. XX-XX, 2009

CARDINALITY AND DENSITY MEASURES AND THEIR INFLUENCE TO MULTI-LABEL LEARNING METHODS

Flavia Cristina Bernardini, Rodrigo Barbosa da Silva, Rodrigo Magalhães Rodovalho, Edwin Benito Mitacc Meza

Laboratório de Inovação no Desenvolvimento de Sistemas (LabIDeS)
Instituto de Ciência e Tecnologia
Universidade Federal Fluminense (UFF) — Rio das Ostras – RJ – Brazil
{fcbernardini,rodrigosilva,rodrigorodovalho,emitacc}@id.uff.br

Abstract – Two main characteristics of multi-label dataset are cardinality and density, related to the number of labels of (each instance of) a multi-label dataset. The relation between these characteristics and multi-label learning performance has been observed with different datasets. However, the difference in domain dataset attributes also interfere on multi-label learning performance. In this work, we use a real dataset, named The Million Song Dataset, available in the internet, which presents the property of having too many labels associated to their instances (songs), as well as so many instances. In this work we present the processed datasets used to conduct our experiments, and we also describe our experiments results.

Keywords – Multi-label Learning, Cardinality and Density Measures.

1 INTRODUCTION

Some real applications are related to the task of classification, such as diagnosis, fault detection, and so on. These problems are commonly treated by machine learning supervised algorithms, which induces classifiers, or predictors, such as neural networks, SVM and decision trees, to cite just a few. These classifiers usually identify just one class of a new instance, or case, from a set of possible labels. However, there are problems related to the task of predicting more than one class for each case. For example, we can mention images and music labeling, failure diagnosis, and others. These kind of problems are tackled by a special type of machine learning, called multi-label learning algorithms. Many multi-label learning methods have been proposed in literature, such as [1–5]. A survey describing some multi-label learning methods can be found in [6]. Two main characteristics analyzed in a multi-label dataset are cardinality and density, both related to the number of labels of each instance of a dataset and also of the entire dataset. Cardinality of a multi-label dataset is the mean of the number of labels of the instances that belong to the dataset, and density of a multi-label dataset is the mean of the number of labels of the instances that belong to the dataset divided by the number of dataset's labels.

Some papers in literature indicate that these dataset characteristics — cardinality and density — may cause different behaviors in multi-label learning methods. In [6], the authors affirm that 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. In [5] we studied the influence of these two characteristics on the performance of the multi-label learners used in our benchmark. We observed that there was a correlation between these characteristics and the results obtained with some datasets; however, the domain of that datasets are quite different, what leaded us to question how the domain features influenced the analysis. In [7], the authors proposed a new method called BRkNN, an adaptation of the kNN algorithm for multi-label classification based on Binary Relevance method, and compared this method with LPkNN, another adaptation method of the kNN algorithm based on Label Powerset multi-label method. The authors observed the influence between the LPkNN method and the influence of low density values, using three different datasets, with different domain features, but they could not safely argue that high density lead to improve performance of the LPkNN. These works analyze the relationship between cardinality (and density) and multi-label learning algorithms results using different datasets, with different cardinality and density values, and different domain dataset attributes. In this way, it is unknown how much the domain difference interferes in cardinality and density analysis. One issue that turns difficult this study is the unavailability of a dataset with the same features but different cardinality and density values.

In [8] the Million Song Dataset is presented, a freely-available collection of audio features and metadata for a million contemporary popular music tracks. The dataset does not include any kind of audio music, only the derived features from them. This collection is available as a relational database. This dataset is labeled by tags that can be seen as musical genres. Each song has more than one of these tags associated to. The main advantage of this dataset on other available multi-label datasets is the high number of labels, which allows to vary the number of labels without loosing the multi-label problems characteristic. One problem with this dataset is the transformation process to allow data mining on it using the available data mining and machine learning tools.

The aim of this work is to present an analysis of the influence of the cardinality and density measures to multi-label learning. To allow this study, we pre-processed the Million Song Dataset. In this work, we present this dataset and the data pre-processing step of the Million Song Dataset. To induce the multi-label classifiers, we used the Mulan library[1] [9], based on Weka [10]. To induce the base classifiers, we used Naïve Bayes and J48 algorithms, because their low time consumption for induction of the classifiers and its lack of requirement for parameters adjustment. We present the results obtained for MSD-based datasets, as well as for six datasets used in [5]. We analyze the relation between (i) cardinality and (ii) density and the results obtained by each method.

This work is organized as follows: Section 2 describes Multi-Label Machine Learning concepts and notations. Section 3 describes the Million Song Dataset, as well as our pre-process step of this dataset. Section 4 describes the conducted experiments and results we obtained. Section 5 concludes this work.

2 MULTI-LABEL LEARNING

Multi-label problems appear in different domains, such as image, text, music, proteins and genome classification [1–3], and failure diagnosis [4]. In multi-label problems, the input to the multi-label learning algorithms is a dataset 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 $f(\mathbf{x}) = Y$. In this work, we call domain attributes datasets the attributes that compose X. 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].

The number of labels $|L|$ is frequently seen as a parameter that influences the performance of different multi-label methods. There are two measures for evaluating the characteristics of a dataset, objects of this study: cardinality $Card$ and density $Dens$ [6]. The cardinality of S is the mean of the number of labels of the instances that belong to S, defined by Eq. 1, and the density of S is the mean of the number of labels of the instances that belong to S divided by $|L|$, defined by Eq. 2.

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

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

2.1 EVALUATION MEASURES

Multi-label machine learning methods can be divided into two categories [6]: problem transformation and algorithm adaptation. In the first category, the multi-label problem is transformed to (many) multiclass (or binary) machine learning problems, and each sub-problem is given to a classic (binary or multiclass) supervised machine algorithm. These (binary or multiclass) classifiers are called in this work base classifiers. In the second category, the machine learning algorithm is adapted to deal with multi-label problems. In this work, we use three methods, commonly used in multi-label learning, named BR, LP and RAkEL. The BR method transforms the original multi-label problem, with $|L|$ labels, into $|L|$ binary problems. The LP method transforms the original multi-label problem into a multiclass problem, considering the relation among the labels. The RAkEL method constructs an ensemble of LP classifiers, which are trained using a small random subset of the set of labels constructed by LP. A more complete description of these (and others) multi-label methods can be found in [6].

[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 or 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.

Multi-label machine learners need classifiers evaluation. For this task, there are three groups of measures to evaluate induced multi-label classifiers: based on instances, based on labels and based on ranking [6]. In this work, we use the first two groups of measure, because multi-label ranking is not the aim of this work. In the first group, we use in this work *Hamming Loss* (Ham), Subset Accuracy ($SAcc$), Accuracy (Acc) and F, defined by Eqs. 3 to 6[3], respectively. In the second group, we use the micro and macro versions of $F1$ measure. 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 of the type $B(t_p, t_n, f_p, f_n)$ can be used in this case. Given that t_{p_l}, t_{n_l}, f_{p_l} and f_{n_l} are true positives, true negatives, false positives and false negatives for each label $l \in L$, the micro version of B measures is denoted by B_- and given by Eq. 7, whereas the macro version of B measures is denoted by B^- and given by Eq. 8. In this work, we use $F1$ and AUC as B measure. $F1(t_p, t_n, f_p, f_n)$ is given by Eq. 9. In [11] there is an explanation about how to calculate Area Under ROC Curves (AUC).

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

$$SAcc(\mathbf{h}, S) = \frac{1}{N}\sum_{i=1}^{N} I(Z_i = Y_i) \tag{4}$$

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

$$F(\mathbf{h}, S) = \frac{1}{N}\sum_{i=1}^{N}\frac{2|Y_i \cap Z_i|}{|Z_i| + |Y_i|} \tag{6}$$

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

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

$$F1(t_p, t_n, f_p, f_n) = \frac{2 \times f_p}{2 \times t_p + f_n + f_p} \tag{9}$$

2.2 DESCRIPTION OF MULTI-LABEL LEARNING METHODS USED IN THIS WORK

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 [6]. 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.

2.2.1 Binary Relevance — BR

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).

[3] In Eq. 3, Δ represents the symmetric difference between two datasets.

2.2.2 Label Powerset — LP

The Label Powerset — LP — method, proposed in [12], 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 $Z = \{l_1, l_2\}$.

2.2.3 RAndom K-labELsets — RAkEL

The RAndom k-labELsets (RAkEL) algorithm constructs an ensemble of multi-label classifiers. Each member of the ensemble is constructed by considering a small random subset of labels and learning LP multi-label classifier, *i.e.*, a single-label classifier for the prediction of each element in the powerset of this subset. In this way, the RAkEL aims to take into account label correlations using single-label classifiers that are applied on subtasks with manageable number of labels and adequate number of examples per label [13].

2.2.4 Hierarchy Of Multilabel classifiERs — HOMER

Problems with large number of labels can be found in several domains. For example, the version of the Million Song Dataset that we use in this work, contains 726 genre music labels. The high dimensionality of the label space may challenge a multi-label learning algorithm in many ways. Firstly, the number of training examples annotated with each particular label will be significantly less than the total number of examples. This is similar to the class imbalance problem in single-label data [14]. Secondly, the computational cost of training a multi-label model may be strongly affected by the number of labels. There are simple algorithms, such as BR with linear complexity with respect to $|L|$, but there are others, such as LP, whose complexity is worse. Thirdly, although the complexity of using a multi-label model for prediction is linear with respect to q in the best case, this may still be inefficient for applications requiring fast response times. Finally, methods that need to maintain a large number of models in memory, may fail to scale up to such domains.

HOMER constructs a Hierarchy Of Multilabel classifiERs, each one dealing with a much smaller set of labels compared to $|L|$ and a more balanced example distribution. This leads to improved predictive performance along with linear training and logarithmic testing complexities with respect to $|L|$. At a first step, HOMER automatically organizes labels into a tree-shaped hierarchy. This is accomplished by recursively partitioning the set of labels into a number of nodes using a balance clustering algorithm. It then builds one multi-label classifier at each node apart from the leafs, following the Hierarchical Binary Relevance (HBR) approach. The HBR approach works as follow. Given a label hierarchy, a binary classifier is trained for each non-root label l of this hierarchy, using as training data those examples of the full training set that are annotated with $par(l)$. During testing, these classifiers are called in a top-down manner, calling a classifier for l only if the classifier for $par(l)$ has given a positive output. The multilabel classifiers predict one or more meta-labels m_l, each one corresponding to the disjunction of a child node's labels.

2.2.5 Classifier Chains — CC

The widely known binary relevance method for multi-label classification, which considers each label as an independent binary problem, has often been overlooked in the literature due to the perceived inadequacy of not directly modelling label correlations. The CC method combines the computational efficiency of BR method and the possibility to use dependency between labels for classification. For each binary model, the space of domain features is extended with 0/1 relevant labels of all former classifiers, building a classifier chain [15].

3 DATASETS DESCRIPTION

In this work, we used the Million Song Dataset and some other multi-label dataset found in the internet. We describe all these datasets in what follows.

3.1 The MSD Dataset

The MSD — The Million Song Dataset[4] [8] — is a freely-available collection of audio features and metadata for a million contemporary popular music tracks. The core of the dataset is composed of features and metadata extracted from one million songs, provided by The Echo Nest[5]. The dataset does not include any kind of audio music, only the derived features from them. Each data music is stored using HDF5 format, which is a data model, library, and file format for storing and managing data. These HDF5 files were constructed using an API provided by The Echo Nest. Each file consists of features extracted from a music, such as version, artist and two types of genres collection associated to each music: (i) Terms, which are tags provided by The Echo Nest, and they can come from a number of places, but mostly from blogs; and (ii) Mbtags, which are tags provided from MusicBrainz specifically applied by humans to a particular artist. Particularly, Mbtags are cleaner than terms for genre recognition.

A HDF5 file has 55 features, and the most important features to use for representing this domain are segment-pitches and segments-timbre. Pitch is the sound property that classifies it as low or high in pitch, or, in other word, bass or sharp sound, respectively. This feature is related to frequency of the signal sound: Higher frequencies, or high pitches, correspond to lower wave length, or sharp sound; Lower frequencies, or low pitches, correspond to higher wave length, or bass sound. Timbre is the sound property dependent from the complexity of the signal sound. Perceiving timbre is affected either by frequencies domain aspects, *i.e.* the way the signal can be decomposed in elementary periodical signals, or time domain aspects, *i.e.* the way the signal amplitude varies with time. Timbre is usually defined as the color of the sound, because by timbre we can identify a sound produced by different fonts, such as two musical instruments playing the same accord or two people singing the same melody [16]. Other important features are artist name (the singer of the music), title of the music, location (where the music was recorded), year when the music was recorded, time duration, segments-start, bars start, similar artists, terms and mbtags — MusicBrainz tags, provided by MusicBrainz[6]. The last five listed features, jointly to segments-timbres and segments-pitches, are multi-valued. segments-start is a list of V values, where V is variable among songs. Each value of segments-start corresponds to the start, in seconds, of intervals, or segments, of the music. segments-pitches and segments-timbres are arrays of two dimensions, where the first one has 12 positions, and each of these positions has V values.

Because MSD contains many multi-valued features, a database-oriented approach to propositionalization is necessary [17]. In [8], they propositionalized only segments-timbre for year prediction task. As described before, segments-timbre has 12 lists, *i.e*, $segT\_list_1, ..., segT\_list_{12}$. In this case, the authors aggregate each list calculating 12 mean values, one for each list, generating the features $mean_{segT\_list_1}$, ..., $mean_{segT\_list_{12}}$. Also, the authors calculate the covariance matrix for the twelve lists. The purpose of this covariance matrix was to verify the variance between each pair of $segT\_list$. The covariance matrix is a matrix whose elements in the (i, j) position is the covariance cov between two random variables x and y; in this case, x is the list $segT\_list_i$, y is the list $segT\_list_j$, $i, j = \{1, ..., 12\}$. The covariance between two random variables x and y, $cov(x, y)$, is defined by the linear correlation coefficient $\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$. When $x \neq y$, $cov(x, y) = cov(y, x)$; and when $x = y$, $cov(x, y) = cov(x, x) = \sigma_x^2$. In this case, where there are 12 lists, instead of generating all the $12^2 = 144$ matrix values, only $\sigma^2_{segt\_list_i}, i = \{1, ..., 12\}$ and $\rho_{segt\_list_i segt\_list_j}, i, j = \{1, ..., 12\}, i > j$ are calculated, what means generating 12 variance features and 60 correlation or covariance features, totalizing 78 covariance features. So, in [8], they generated 90 features from the Million Song Dataset.

In this work, we did not only consider these 90 features, but we also considered the segments-pitches multi-valued feature, because we believe that the pitch of the music may influence its genre definition. The same procedure used to generate the features extracted from segments-timbre was used to generate features from segments-pitches. In this way, three features subsets are constructed:

1. Means of segments-timbre lists, represented by $\{mean_{segP\_list_1}, ..., mean_{segP\_list_{12}}\}$;

[4] http://labrosa.ee.columbia.edu/millionsong/

[5] http://echonest.com/.

[6] http://musicbrainz.org/

2. Variances of segments-timbre lists, represented by $\{\sigma^2_{segP\_list_1}, ..., \sigma^2_{segP\_list_{12}}\}$; and
3. Correlation coefficients of segments-timbre lists, represented by $\{\rho_{segP\_list_1 segP\_list_2}, ..., \rho_{segP\_list_1 segP\_list_{12}}, \rho_{segP\_list_2 segP\_list_3}, ..., \rho_{segP\_list_2 segP\_list_{12}}, ..., \rho_{segP\_list_{11} segP\_list_{12}}\}$.

Considering the aggregations of segments-timbre and segments-pitches, the description features totalize 180 domain features. Each instance was classified by the tags given by MusicBrainz, as described earlier.

The original dataset contains 1 million songs. The authors also made available a sample of the original dataset containing 10.000 songs, which was used for this work. When analyzing this dataset sample, we observed that (i) there were instances without any label; and (ii) there were labels with too few instances associated to them, as well as there were labels with too many of them. Instances without any label were discarded, resulting 3.710 instances. Labels with too few instances associated to them could be considered noisy labels. Next section describes the experiments realized in this work.

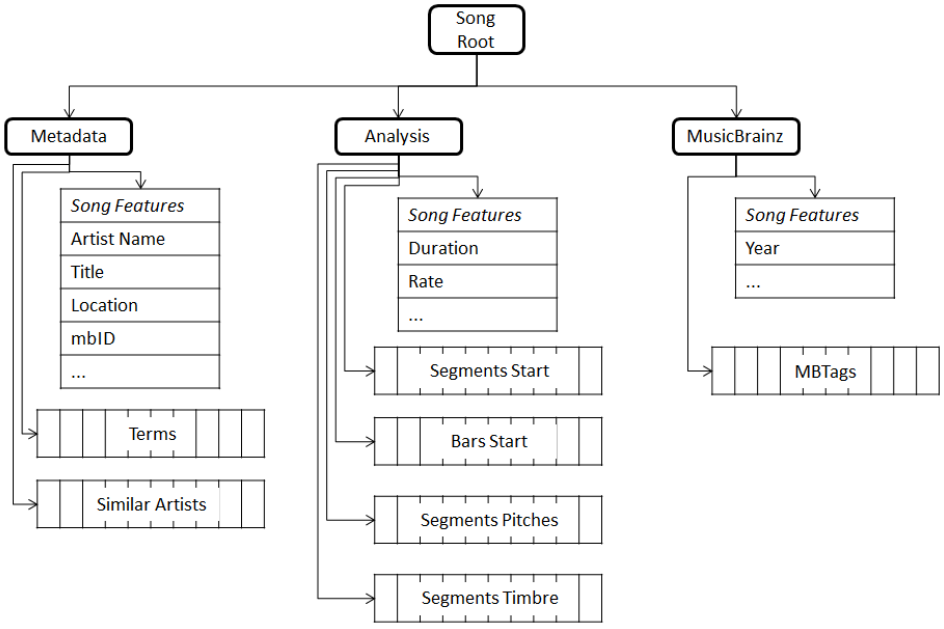


Figure 1: A visualization of the structure of each HDF5 file of the MSD dataset.

In this work, we used MSD to vary cardinality and density values. For this task, we considered that each label should be linked to a minimum of N_0 instances on the dataset. We considered the following values as minimum instances to each label: $N_0 \in \{0, 5, 15, 25, 35, 45, 65, 75, 85, 95, 145, 195\}$, where $N_0 = 0$ means that all the labels were considered; $N_0 = 5$, only labels with 5 or more instances associated with it were considered; $N_0 = 15$, only labels with 15 or more instances associated with it were considered; and so on. Each generated dataset was renamed to MSD-000, MSD-005, MSD-015, MSD-025, MSD-035, MSD-045, MSD-055, MSD-065, MSD-075, MSD-085, MSD-095, MSD-145 and MSD-195[7]. Table 1 describes the main characteristics of each generated datasets, where Min #Inst indicates the minimum number of instances a label has to be associated to be considered; #Inst represents the number of instances resulted after disconsidering labels that do not satisfy the Min #Inst Per Label condition; #Labels represents the number of remaining labels; $Card$ is the label cardinality value — Eq. 1; and $Dens$ is the label density value — Eq. 2. We should remember that each dataset has 180 domain dataset attributes, all numerical ones.

3.2 Natural Datasets

We used six natural datasets in our experiments, also used in [5][8]: Emotions, Genbase, Scene, Yeast, Enron e Medical. Table 2 describes characteristics of these datasets, where #Inst. is the number of instances in the dataset;

[7]The generated datasets are available at `http://www.professores.uff.br/fcbernardini/papers/compl/MSD_MR/`

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

Table 1: MSD-Generated Datasets Characteristics

	Min #Inst	#Inst	#Labels	$Card$	$Dens$
MSD-000	0	3710	726	3.8919	0.0054
MSD-005	5	3669	483	3.7817	0.0078
MSD-015	15	3587	272	3.4767	0.0128
MSD-025	25	3541	202	3.2937	0.0163
MSD-035	35	3506	161	3.1954	0.0198
MSD-045	45	3466	140	3.1056	0.0222
MSD-055	55	3408	122	2.9759	0.0244
MSD-065	65	3372	107	2.9517	0.0276
MSD-075	75	3345	98	2.8906	0.0295
MSD-085	85	3340	90	2.8189	0.0313
MSD-095	95	3256	84	2.8443	0.0339
MSD-145	145	3080	62	2.6182	0.0422
MSD-195	195	2904	47	2.4938	0.0531

#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.

Table 2: Datasets Characteristics

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

Figures 2 and 3 show, respectively, cardinality and density values of each dataset used in this work. We can observe in Figure 3 that density values in MSD datasets are much lower than density values of the natural datasets.

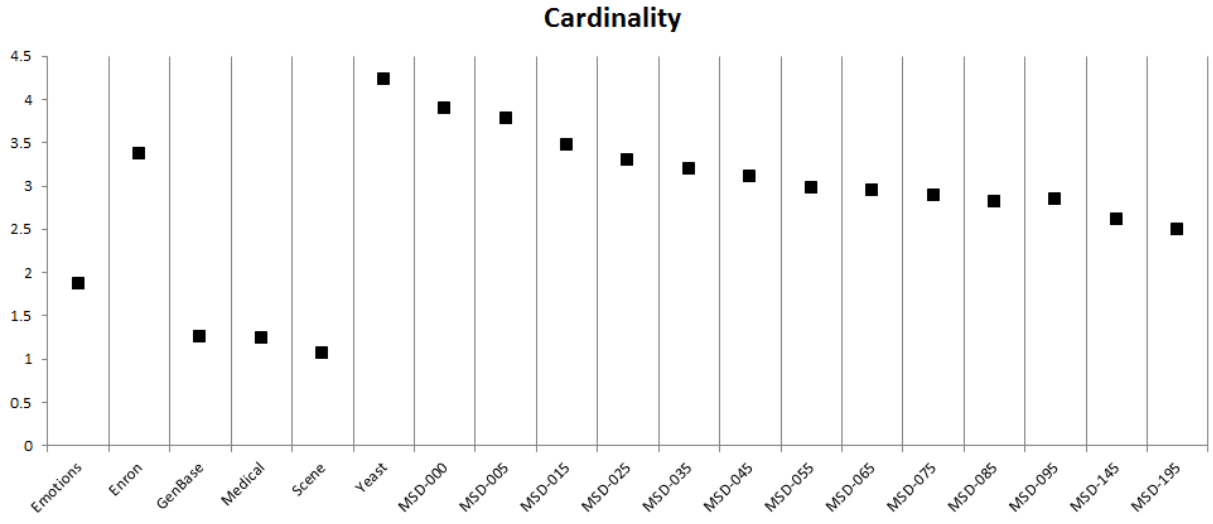


Figure 2: Cardinality Values of Each Dataset

4 RESULTS AND ANALYSES

To evaluate the influence of cardinality and density characteristics to multi-label learning, we considered five multi-label learning methods frequently used in literature, briefly described in Section 2.2 — BR, LP, RAkEL, HOMER [6] and CC [15]. As base learning algorithms, we used Naïve Bayes (NB) and J48 [10]. We denote each combination of multi-label learning method and base learning algorithm as BR-NB, BR-J48, CC-J48, CC-NB, HOMER-J48,

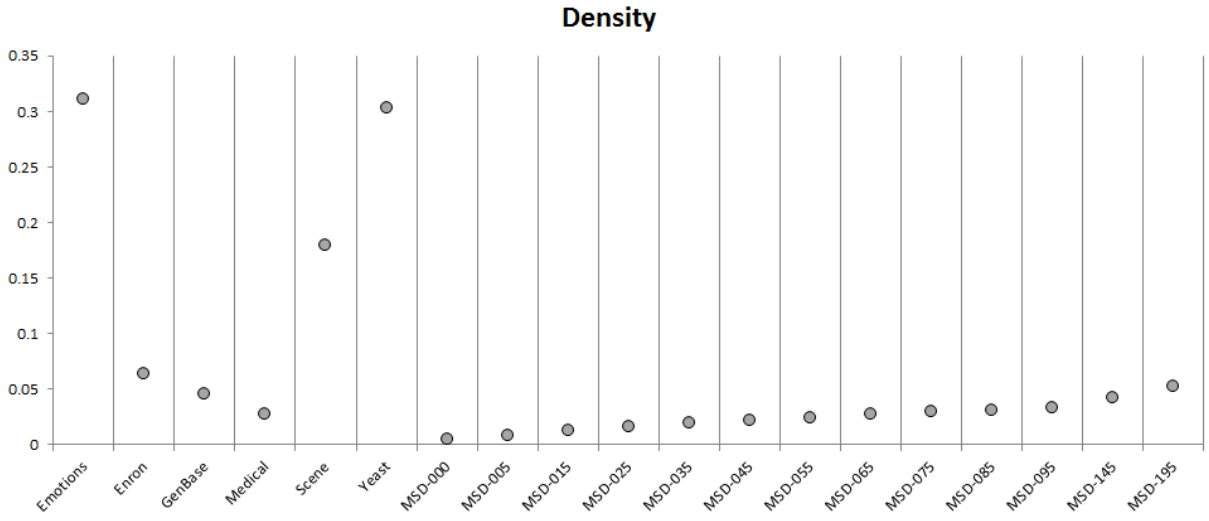


Figure 3: Density Values of Each Dataset

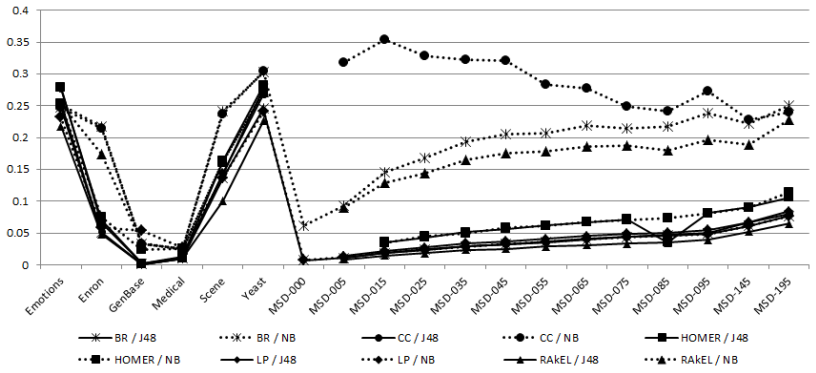
HOMER-NB, LP-J48, LP-NB, RAkEL-J48 and RAkEL-NB. Figures 4 and 5 shows all the results obtained for each triple of (i) dataset, (ii) multi-label learning method and (iii) base learning algorithm. It is important to observe that the methods CC-J48, CC-NB,LP-J48, LP-NB, RAkEL-J48 and RAkEL-NB could not be executed for MSD-000 dataset; and HOMER-J48 and HOMER-NB could not be executed for both MSD-000 and MSD-005 datasets. All of these executions could not be terminated by lack of memory problem.

We aim to analyze if there is a relation between 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. Because Pearson Correlation is a parametric statistic, we first executed the Anderson-Darling's normality test for all algorithms results. In some results we could reject the normality test, what leaded us to measure Spearman's rank correlation[9] [18].

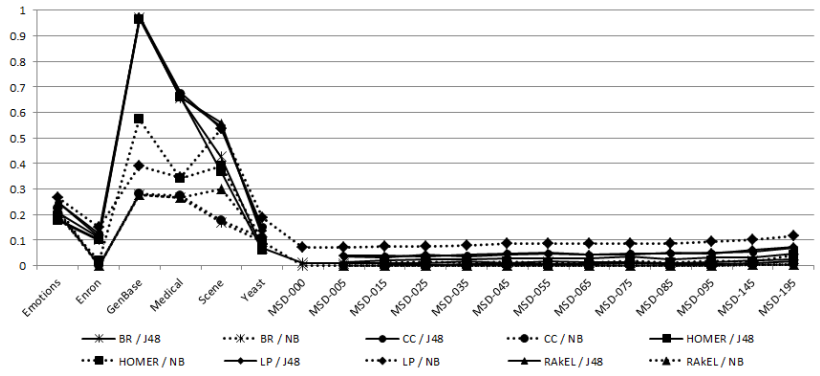
Spearman's rank correlation was calculated between $Card$ and each measure results, and also was calculated between $Dens$ and each measure results. Correlation between the results and $Card$, as well as between the results and $Dens$, was expected. Figures 6 and 7 shows the $|\rho(Card, Mea)|$ and $|\rho(Dens, Mea)|$ values for measures $Mea \in \{Ham, SA, F, Acc, F1_{-}, F1^{-}\ AUC_{-}\}$.

We can observe that, when putting together all results, we could not observe high correlation between Cardinality values and measures' results for all but four situations. These exceptions can be observed in Figure 6(a) for CC-NB method, and in Figure 7(c) for CC-NB, HOMER-NB and LP-NB. On the other hand, for $SAcc$, F and Acc measures, all the correlations between $Card$ and Mea ($\rho(Card, Mea)$) are near 0.7. Regarding to density values, we can observe that for Ham, $SAcc$, F, Acc and $F1_{-}$ measures, which correlation values are shown in Figures 6(a), 6(b), 6(c),6(d) and 7(a), all but three correlations are lower than 0.7. The exception are for Ham measure and CC-NB, HOMER-J48 and RAkEL-NB methods. Also, we noticed that multi-label methods may be more affected by low density values than by high cardinality values. Because LP and RAkEL transform the original multi-label problem into transformed multi-class(es) problem(s), it was expected that these methods would show high correlation considering both $Card$ and $Dens$ values. However, only $Dens$ showed high correlations with Mea measures. Finally, we also observed that, for $F1^{-}$ and AUC_{-} measures, we could not observe any pattern in correlation behaviour.

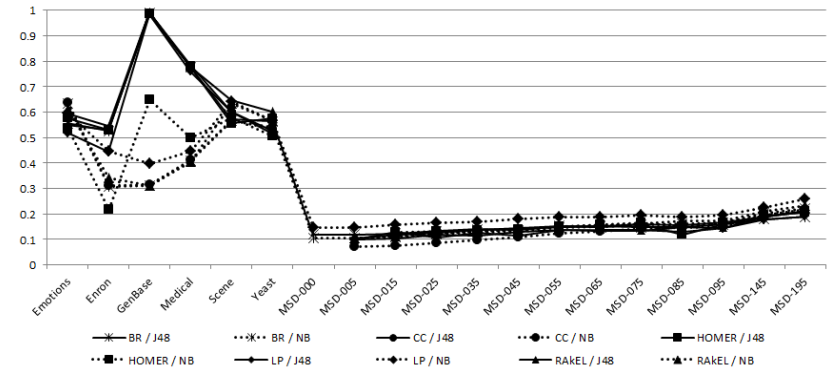
[9]Anderson-Darling's normality test and Spearman's rank correlation was calculated using R software, available at `http://www.r-project.org/`



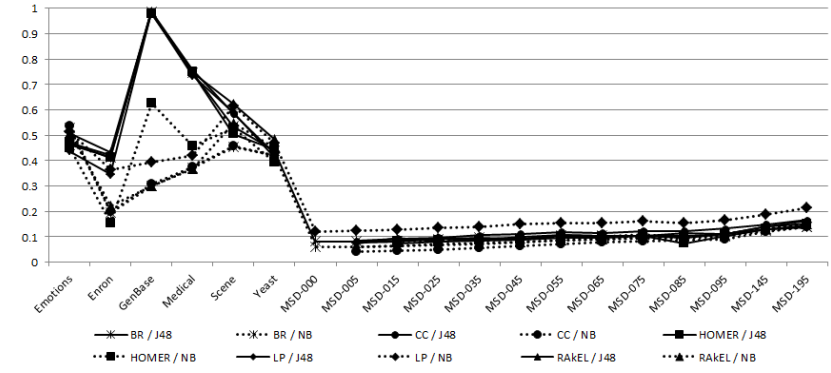
(a) Results — Ham Measure



(b) Results — $SAcc$ Measure

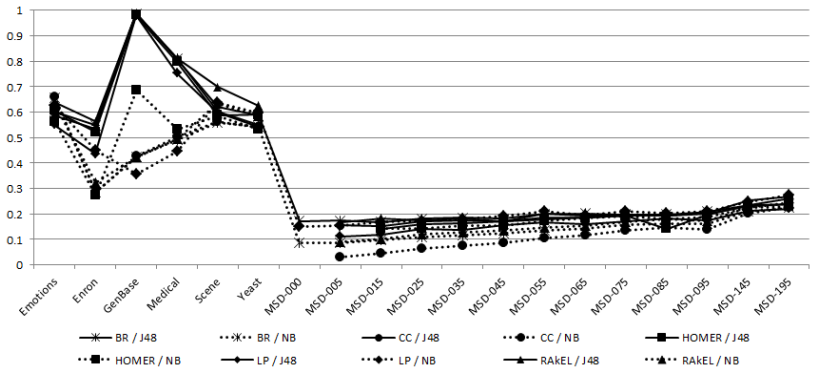


(c) Results — F Measure

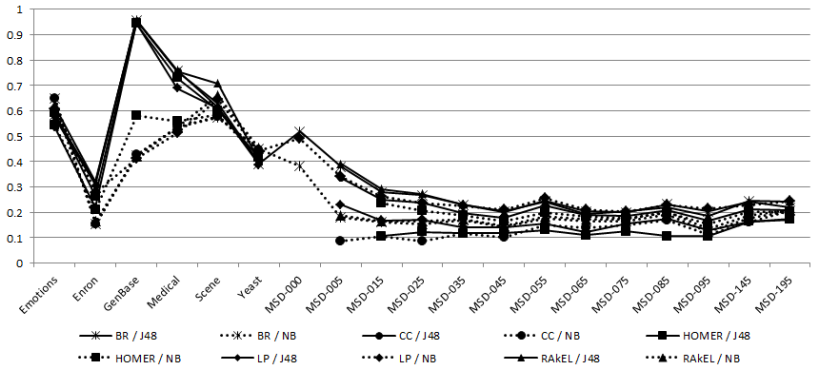


(d) Results — Acc Measure

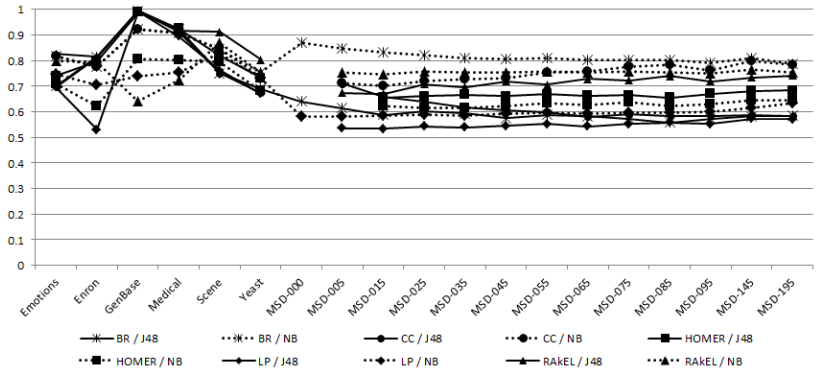
Figure 4: Results for instance-based measures $Mea \in \{Ham, SA, F, Acc\}$, all datasets and all multi-label learning methods.



(a) Results — $F1_{-}$ Measure

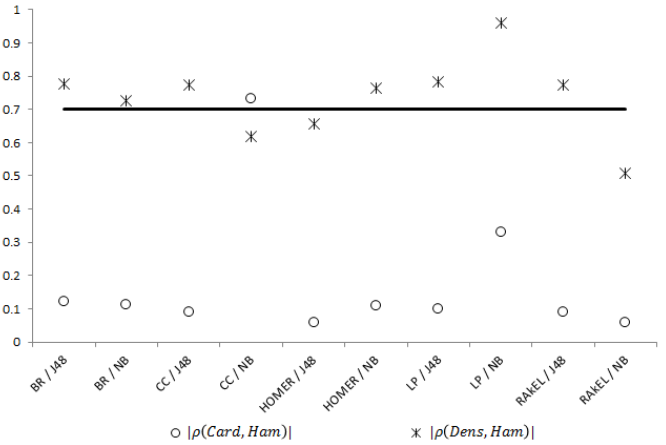


(b) Results — $F1^{-}$ Measure

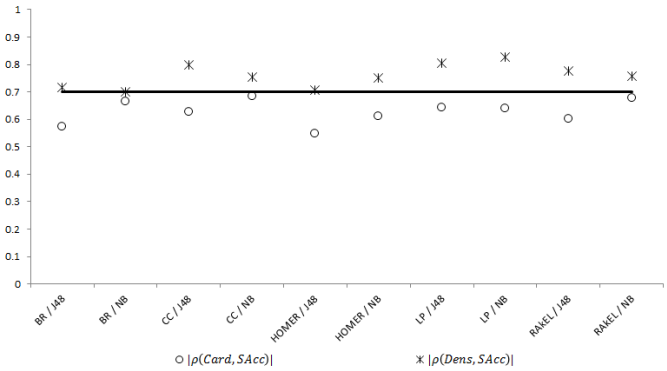


(c) Results — AUC_{-} Measure

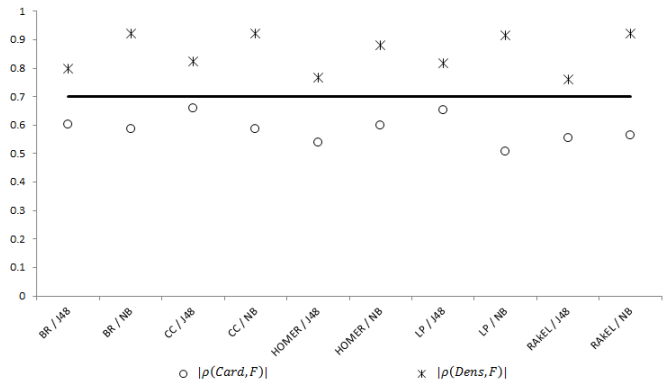
Figure 5: Results for label-based measures $Mea \in \{F1_{-}, F1^{-}\ AUC_{-}\}$, all datasets and all multi-label learning methods.



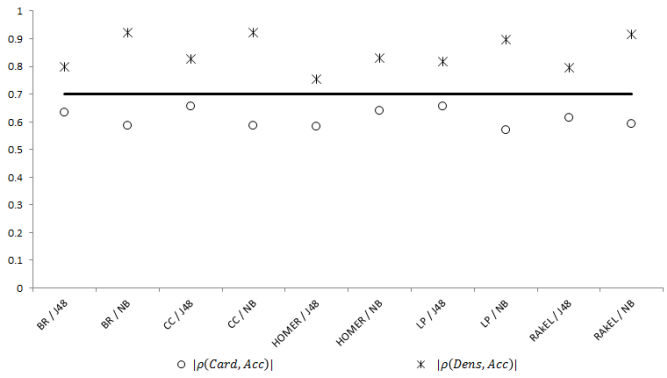
(a) Ham Measure



(b) $SAcc$ Measure

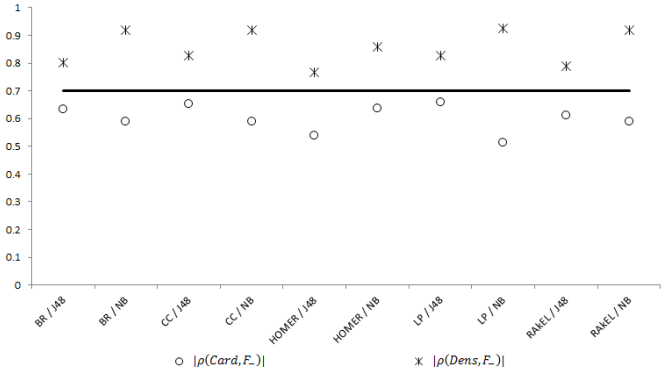


(c) F Measure

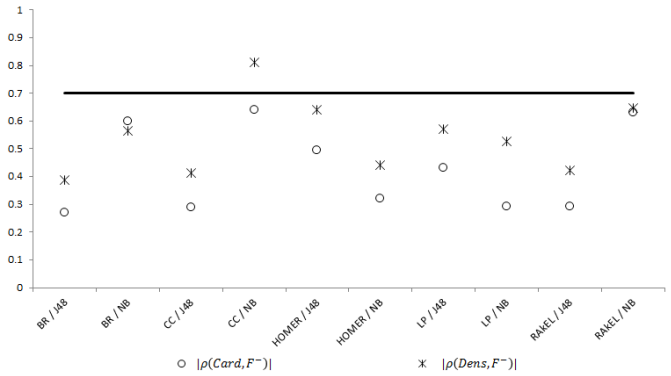


(d) Acc Measure

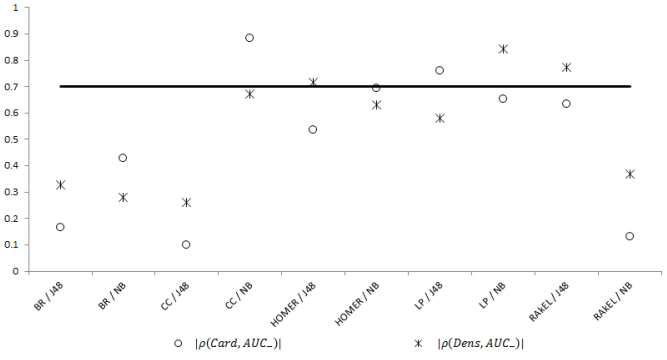
Figure 6: $|\rho(Card, Mea)|$ and $|\rho(Dens, Mea)|$ values for each instance-based measures $Mea \in \{Ham, SA, F, Acc\}$.



(a) $F1_-$ Measure



(b) $F1^-$ Measure



(c) AUC_- Measure

Figure 7: $|\rho(Card, Mea)|$ and $|\rho(Dens, Mea)|$ values for label-based measures $Mea \in \{F1_-, F1^- \; AUC_-\}$.

5 CONCLUSIONS AND FUTURE WORK

Cardinality and density are multi-label datasets' characteristics related to the degree of difficulty to learn a multi-label classifier, *i.e.*, lower the density and higher the cardinality, more difficult the multi-label learning process. In [5], we started our investigation on how much cardinality and density could impact the multi-label learning methods' results. In that work, we used only six natural datasets, available in the internet. However, all of them have different domain features. In this work, we describe the million song dataset, the pre-process phase for multi-label learning, and the generated datasets, with the same domain features, but different cardinality and density values. Also, we considered the results of the six datasets used before, to compose our analyzes. We could observe in this work that density dataset characteristic shows more influence in multi-label learning than cardinality characteristic. So, exploring how to increase density values without changing the learning problems could be an interesting approach.

Also, it is important to notice that real multi-label datasets may present low density values and high number of labels. It should be observed that HOMER is a method developed to scale up multi-label learning according to number of labels; however, HOMER could not be executed for the the datasets with highest number of labels, what indicates that investigation of more scalable algorithms is interesting.

Acknowledgements

We would like to thank to PIBIC/CNPq/UFF for the scientific initiation scholarship granted to the research; FAPERJ for the financial support (E-26/110.552/2012); Prof. Alexandre Plastino (UFF) for his valuable observations; and Prof. Ana Paula Sobral (UFF) for helping us on the evaluation process of this paper.

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] 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.

[5] P. P. da Gama, F. C. Bernardini and B. Zadrozny. "RB: A new method for constructing multi-label classifiers based on random selection and bagging". *Learning and Nonlinear Models*, vol. 11, no. 1, pp. 26–47, 2013.

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

[7] E. Spyromitros, G. Tsoumakas and I. Vlahavas. "An Empirical Study of Lazy Multilabel Classification Algorithms". In *Proc. 5th Hellenic Conf. on Artificial Intelligence: Theories, Models and Applications – SETN'08*, pp. 401–406, 2008.

[8] T. Bertin-Mahieux, D. P. Ellis, B. Whitman and P. Lamere. "The Million Song Dataset". In *Proc. 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

[9] 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.

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

[11] T. Fawcett. "ROC graphs: Notes and practical considerations for researchers." *Machine Learning*, vol. 31, pp. 1–38, 2004.

[12] 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.

[13] G. Tsoumakas and I. Vlahavas. "Random k-labelsets: An ensemble method for multilabel classification". In *Proc. European Conference on Machine Learning*, pp. 406–417, 2007.

[14] N. Chawla, Japkowicz, N. and A. Kotcz. "Editorial: special issue on learning from imbalanced data sets". *SIGKDD Explorations*, vol. 6, pp. 1–6, 2004.

[15] J. Read, B. Pfahringer, G. Holmes and E. Frank. "Classifier Chains for Multi-label Classification". In *Proc 13th European Conference on Principles and Practice of Knowledge Discovery in Databases and 20th European Conference on Machine Learning*, 2009.

[16] C. Stephanidis. *The Universal Access Handbook*. CRC Press, 2010.

[17] M.-A. Krogel, S. Rawles, F. Železný, P. A. Flach, N. Lavrač and S. Wrobel. "Comparative Evaluation of Approaches to Propositionalization". In *Proc. 13th Intern. Conf. on ILP, LNCS*, volume 2835, pp. 197–214. Springer, 2003.

[18] C. T. Ekstrøm. *The R Primer*. CRC Press, 2011.