Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC), Vol. 12, Iss. 1, pp. 53-71, 2014

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 used a real dataset named The Million Song Dataset (MSD), available on the internet. A particularly useful characteristic of this dataset is the existence of many labels associated to their instances (songs). We conduct the experiments on datasets processed from MSD, and the results show that both density and cardinality characteristics influence the performance of the multi-label learning methods used in this work. To extend our analysis, we also analyze the results obtained in natural datasets, i.e, datasets available on the internet pre-processed for empirical tests in multi-label learning. Our results show that density characteristic influences more to multi-label learning than cardinality characteristic.

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, 7]. 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 datasets's cardinality divided by the number of dataset's labels.

Many large datasets have been collected on the internet. When these datasets present a multi-label problem to be tackled, it is common that the number of labels is very large, with (very) low density values, and, in some cases, large cardinality values. Some studies treat this kind of problems [8, 9]. However, is not yet clear how cardinality and density characteristics influence multi-label learning performance. Some researches in literature indicate that these dataset characteristics — cardinality and density — may cause different behaviors in multi-label learning methods. In [7], 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 [10], the authors proposed a new method called BRkNN, an adaptation of the kNN algorithm for multi-label classification based on Binary Relevance method (BR), and compared this method with LPkNN, another adaptation method of the kNN algorithm based on Label Powerset multi-label method (LP). 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. In [5] the influence of these two characteristics was studied on the performance of two multi-label learners also used in this work — BR and LP, also used in our benchmark. In this work, correlations were observed between cardinality and density and the results obtained with some datasets. However, the domains of that datasets are quite different, what leaded us to question how the domain features influenced the analysis. All of these related studies 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 [11] the Million Song Dataset (MSD) 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 cardinality and density measures to multi-label learning. To allow this study, we pre-processed MSD. In this work, we present this dataset and its data pre-processing step. To induce the multi-label classifiers, we used the Mulan library[1] [12], based on Weka [13]. 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. We also bring to this work the results obtained for the six datasets used in [5] to enrich our analysis, and we also enlarge that results to evaluate similarly to the evaluation using MSD. 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

In classical supervised 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 [13, 14].

On the other hand, the multi-dimensional classification problem consists of finding a function $\mathbf{h}$ that assigns to each instance $\mathbf{x} = (x_{i1}, ..., x_{iM})$ a vector of $|L|$ class values $\mathbf{c} = (c_1, ..., c_|L|)$, i.e, $\mathbf{h} : \mathcal{D}_{X_1} \times ... \times \mathcal{D}_{X_M} \rightarrow \mathcal{D}_{C_1} \times ... \times \mathcal{D}_{C_|L|}$, and so $(x_1, ..., x_M) \mapsto (c_1, ..., c_|L|)$. We assume that C_l is a discrete variable, for all $l = 1, ..., |L|$, with $\mathcal{D}_{C_l}$ denoting its sample space, and $\mathcal{I} = \mathcal{D}_{C_1} \times ... \times \mathcal{D}_{C_|L|}$, the space of joint configurations of the class variables. Analogously, $\mathcal{D}_{X_j}$ is the sample space of the discrete feature variable X_j, for all $j = 1, ..., m$. Multi-dimensional classification is a more difficult problem than the single-class case. The main problem is that there is a large number of possible class label combinations, $|\mathcal{I}|$, and a corresponding sparseness of available data.

A particular case of multi-dimensional classification problem is the class multi-label classification problems, where $\mathcal{D}_{C_l} = \{0, 1\}$. 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].

[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.

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 $Dens$ [7]. 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

To evaluate multi-label learning algorithms, there are three groups of measures to evaluate induced multi-label classifiers: based on instances, based on labels and based on ranking [7, 15]. In this work, we use the first two groups of measure, because multi-label ranking is not the aim of this work. In what follows, we describe each of the used measures.

Measures based on instances: *Hamming Loss* (Ham) evaluates how many times an example-label pair is misclassified, i.e., how many times a label belonging to the example is not predicted or a label not belonging to the example is predicted. $Ham(\mathbf{h}, S)$ is defined by Equation 3, where Δ stands for the symmetric difference between two datasets, N is the number of examples and $|L|$ is the number of labels in the dataset S. The smaller the value of $Ham(\mathbf{h}, S)$, the better the performance of $\mathbf{h}$, and the performance is perfect when $Ham(\mathbf{h}, S) = 0$. Subset Accuracy ($SAcc$), or classification accuracy, is a very strict evaluation measure as it requires the predicted set of labels to be an exact match of the true set of labels. $SAcc(\mathbf{h}, S)$ is defined by Equation 4, where $I(true) = 1$ and $I(false) = 0$. Accuracy (Acc) for a single example $\mathbf{x}_i$ is defined by the Jaccard similarity coefficients between the label sets $\mathbf{h}(\mathbf{x}_i) = Z_i$ and Y_i. Accuracy is micro-averaged across all examples. $Acc(\mathbf{h}, S)$ is defined by Equation 5. F is the harmonic mean between precision and recall. $F(\mathbf{h}, S)$ is defined by Equation 6.

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

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

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

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

Measures based on labels: 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. $F1$ is the harmonic mean between precision and recall. $F1(t_p, t_n, f_p, f_n)$ is given by Eq. 9. In [16] there is an explanation about how to calculate Area Under ROC Curves (AUC).

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

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

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

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

Multi-label machine learning methods can be divided into two categories [7]: 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 base classifiers. In the second category, the machine learning algorithm is adapted to deal with multi-label problems. Several multi-label learning methods were proposed in literature in each category, and many of them are describe in [6, 7].

In this work, we use five methods, commonly used in multi-label learning, named BR, LP, RAKEL, CC and HOMER. BR was chosen due to its low complexity when compared to other multi-label learners [17]. LP method was chosen because it was the first method to deal with correlations among labels, although being challenged by domains with large number of labels. The choice of RAKEL is due to being an extension of LP, attempting to the computational complexity problem of LP [18]. CC and HOMER were appointed as two of the best multi-label learning algorithms in [15].

2.2.1 Binary Relevance — BR

One main approach to solve a multi-label learning problem is decomposing the original problem into various binary problems. The most popular method based on this approach is called *Binary Relevance* — BR —, used in [2]. In BR method, firstly the training dataset S_m is transformed into $|L|$ datasets S_{l_i}, where each dataset corresponds to a label $l_i, i = 1, ..., |L|$. Then, a classifier for each label l_i, named $\mathbf{h}_{l_i}$, is constructed using a supervised learning algorithm for binary problems. A new instance $\mathbf{x}$ is classified by the labels which $\mathbf{h}_{l_i} = 1$ (or $\mathbf{h}_{l_i} = true$).

2.2.2 Label Powerset — LP

The Label Powerset — LP — method, proposed in [19], transforms the original multi-label problem into a multiclass problem. In LP, each set of labels Y_i in S_m is considered a class. So, each $\mathbf{x}_i$ is classified by the new label y'_i, where y'_i is the concatenation of all labels in Y_i. For instance, considering three labels l_1, l_2 and l_3 and a multi-label training dataset S_m, the original instance $\mathbf{T}_1 \in S_m$, labeled with $Y_1 = \{l_1, l_2\}$, after the transformation is labeled with $y'_1 = l_{1,2}$; the instance $\mathbf{T}_2 \in S_m$ labeled with $Y_2 = \{l_1, l_3\}$, after the transformation is labeled with $y'_2 = l_{1,3}$; the instance $\mathbf{T}_3 \in S_m$ labeled with $Y_3 = \{l_1\}$, after the transformation is (still) labeled with $y'_3 = l_1$; and so on. After this process, a multiclass classifier $\mathbf{h}^*$ is induced using the generated dataset.

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. For instance, let us consider that $\mathbf{h}^*$ outputs the following probability distribution for $\mathbf{x}$: $l_{1,2} = 0.7$, $l_{2,3} = 0.2$ 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 $Z = \{l_1, l_2\}$.

2.2.3 RAndom K-labELsets — RAKEL

The RAndom K-labELsets (RAKEL) method constructs an ensemble of multi-label classifiers $\mathbf{h}^*$ [18]. Firstly, RAKEL method constructs m random subsets of labels, called R_i, each of them containing K labels from L. Then, each $R_i, i = 1, ..., m$ is used to induce a multi-label model $\mathbf{h}_i$ using LP multi-label learner method.

Given a new instance $\mathbf{x}$ to be classified, each $\mathbf{h}_i$ provides binary predictions $\mathbf{h}_i(\mathbf{x}, l_j)$ for each label $l_j \in R_i$. Subsequently, RAKEL calculates the mean of these predictions for each label $l_j \in L$ and outputs a final positive decision if it is greater than a 0.5 threshold. For instance, considering $L = \{l_1, l_2, l_3, l_4, l_5, l_6\}$ (and so $|L| = 6$), $m = 7$ and $k = 3$, Figure 1 shows how an instance $\mathbf{x}$ is classified given a multi-label model $\mathbf{h}^*$ constructed using RAKEL method. It should be observed that the default value of parameter m is usually $m = 2 \times |L|$.

		Predictions					
Model	Label Set (R_i)	l_1	l_2	l_3	l_4	l_5	l_6
$\mathbf{h}_1$	$R_1 = \{l_1, l_2, l_6\}$	1	0	-	-	-	1
$\mathbf{h}_2$	$R_2 = \{l_2, l_3, l_4\}$	-	1	1	0	-	-
$\mathbf{h}_3$	$R_3 = \{l_3, l_5, l_6\}$	-	-	0	-	0	1
$\mathbf{h}_4$	$R_4 = \{l_2, l_4, l_5\}$	-	0	-	0	0	-
$\mathbf{h}_5$	$R_5 = \{l_1, l_4, l_5\}$	1	-	-	0	1	-
$\mathbf{h}_6$	$R_6 = \{l_1, l_2, l_3\}$	1	0	1	-	-	-
$\mathbf{h}_7$	$R_7 = \{l_1, l_4, l_6\}$	0	-	-	1	-	0
	Average Votes	3/4	1/4	2/3	1/4	1/3	2/3
	Final Prediction (Binary values)	1	0	1	0	0	1
	Final Prediction (Set of Labels)	$Z = \{l_1, l_3, l_6\}$					

Figure 1: Example of classification of an instance $\mathbf{x}$ by a multi-label classifier $\mathbf{h}^*$ constructed by RAKEL method [18].

2.2.4 Hierarchy Of Multilabel classifiERs — HOMER

Problems with large number of labels can be found in several domains. For instance, the version of the Million Song Dataset (MSD) 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 [20]. Secondly, the computational cost of training a multi-label model may be strongly affected by the number of labels. To exemplify this problem, considering the BR method, the algorithm complexity is linear with respect to $|L|$, and considering LP method, its complexity is even worse. Thirdly, although the complexity of using a multi-label model for prediction is linear with respect to $|L|$ in the best case, this may still be inefficient for applications requiring fast time response. Finally, methods that need to maintain a large number of models in memory may fail to scale up to such domains [7].

HOMER constructs a Hierarchy Of Multilabel classifiERs [21]. The method follows the divide-and-conquer paradigm of algorithm design, transforming a large set os labels L into a tree-shaped hierarchy. The root L_{root} of this tree contains all labels $l_i \in L$, i.e., $L_{root} = L$. Each leaf of this tree contains one, and only one, label from L, and all of the leaves are disjunct, i.e., $L_{leaf_i} = \{l_i\}, i = 1, ..., |L|$. Each internal node L_{node} contains the union of the label sets of its children, i.e., $L_{node} = \cup L_{children}, children \in \text{children}(node)$.

In [21] the authors also present a definition of meta-label: The meta-label of a node L_{node}, μ_{node}, is a disjunction of the labels contained in that node, $\mu_{node} \equiv \vee l_j, l_j \in L_{node}$. A training instance $\mathbf{x}_i$ is annotated with a meta-label μ_{node} if Y_i has at least one label of μ_{node}, i.e., $\mu_{node} \cap Y_i \neq \emptyset$. For each meta-label, a multi-label classifier $\mathbf{h}_{node}$. The task of $\mathbf{h}_{node}$ is the prediction of one or more of the meta-labels of its children. Given a new instance $\mathbf{x}$ to be classified, this instance is firstly presented to $\mathbf{h}_{node}$, which is a multi-label classifier. Remembering that $\mathbf{h}_{node}$ classifies an instance into a set of labels $Z_i \in L$, the instance $\mathbf{x}$ will be conducted to their children and internal nodes L_{node} which meta-labels $\mu_{node} \cap Z_i \neq \emptyset$. This process is followed until the instance is classified by the leaves. Considering, for instance, the sample hierarchy shown in Figure 2.

2.2.5 Classifier Chains — CC

The transformation approach based on decomposing the original problem into various binary problems is used by many other proposed methods, and Classifier Chains (CC) is when of them, proposed by Read [22]. The method BR assumes label independence in the multi-label problem, and is commonly mentioned as the main problem of the method. CC takes advantage of the computational efficiency of BR, and also includes the possibility to use dependency between labels for classification. To achieve this purpose, CC also constructs $|L|$ binary classifiers, as in BR. On the other hand, to turn possible using dependency between labels, CC considers an order of the elements of the label set L, for instance $(l_1, l_2, L_3, ..., l_{|L|})$. Then, the base classifiers are linked along a chain, which forms the multi-label classifier $\mathbf{h}^*$. The chain is formed as follows: The first classifier $\mathbf{h}_1$ is constructed using only the domain attributes $X_i \in X$. Each of the other classifiers $\mathbf{h}_j, j = 2, ..., |L|$ deals with the binary relevance problem associated with all labels $l_1, ..., l_{j-1} \in L$, and so the feature space of each link in the chain is extended with the 0/1 label associations

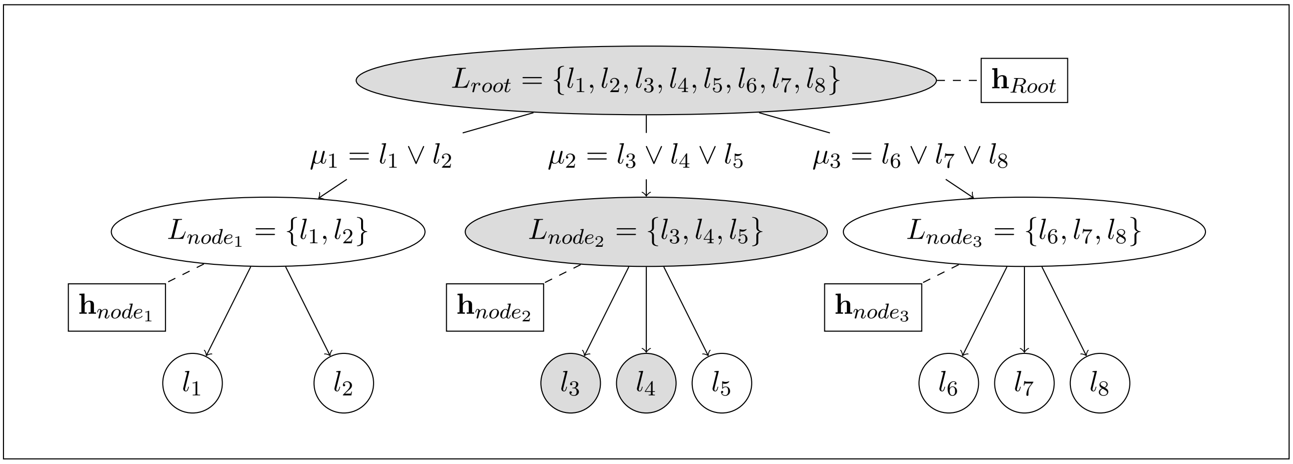

$L_{root} = \{l_1, l_2, l_3, l_4, l_5, l_6, l_7, l_8\}$
$\mathbf{h}_{Root}$
$\mu_1 = l_1 \vee l_2$
$\mu_2 = l_3 \vee l_4 \vee l_5$
$\mu_3 = l_6 \vee l_7 \vee l_8$
$L_{node_1} = \{l_1, l_2\}$
$L_{node_2} = \{l_3, l_4, l_5\}$
$L_{node_3} = \{l_6, l_7, l_8\}$
$\mathbf{h}_{node_1}$
$\mathbf{h}_{node_2}$
$\mathbf{h}_{node_3}$
l_1
l_2
l_3
l_4
l_5
l_6
l_7
l_8

Figure 2: A sample hierarchy constructed by HOMER. Grey filled shapes indicate a possible path of an instance $\mathbf{x}$ given to the multi-label classifier $\mathbf{h}^*$ constructed by HOMER.

of all previous links. In other words, the label l_1 is added to the attribute domain X to induce $\mathbf{h}_2$; labels l_1 and l_2 are added to X to induce $\mathbf{h}_3$; and so on.

To classify a new instance $\mathbf{x}$, $\mathbf{h}^*$ is used respecting the order of the formed chain $(\mathbf{h}_1, ..., \mathbf{h}_{|L|})$. Each classifier $\mathbf{h}_j$ is responsible for learning and predicting the binary association of label l_j given the feature space, augmented by all prior binary relevance predictions in the chain $(l_1, ..., l_{j-1})$. The classification process begins at $\mathbf{h}_1$ and propagates along the chain: $\mathbf{h}_1$ determines $Pr(l_1|\mathbf{x})$, and every following classifier $\mathbf{h}_2, ..., \mathbf{h}_{|L|}$ predicts $Pr(l_j|\mathbf{x}, l_1, ..., l_{j-1})$. In other words, $\mathbf{h}_1$ firstly classifies $\mathbf{x}$; then $\mathbf{x}' = (x_1, ..., x_M, \mathbf{h}_1(\mathbf{x}))$ is classified by $\mathbf{h}_2$; $\mathbf{x}'' = (x_1, ..., x_M, \mathbf{h}_1(\mathbf{x}), \mathbf{h}_2(\mathbf{x}'))$ is classified by $\mathbf{h}_3$; and so on.

3 DATASETS DESCRIPTION

In this work, we use the Million Song Dataset. Also, we use and expand the results obtained in [5], which consider other multi-label dataset found on the internet. We describe all these datasets in what follows.

3.1 The MSD Dataset

The MSD — The Million Song Dataset[3] [11] — 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[4]. 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 [23]. Other important features are *artist name* (the singer of the music), *title* of the music, *location* (where the music was recorded), *year* when

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

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

the music was recorded, time *duration*, *segments-start*, *bars start*, *similar artists*, *terms* and *mbtags* — MusicBrainz tags, provided by MusicBrainz[5]. 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 [24]. In [11], 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 [11], they generated 90 features from the Million Song Dataset. Figure 3 shows the structure of the MSD dataset.

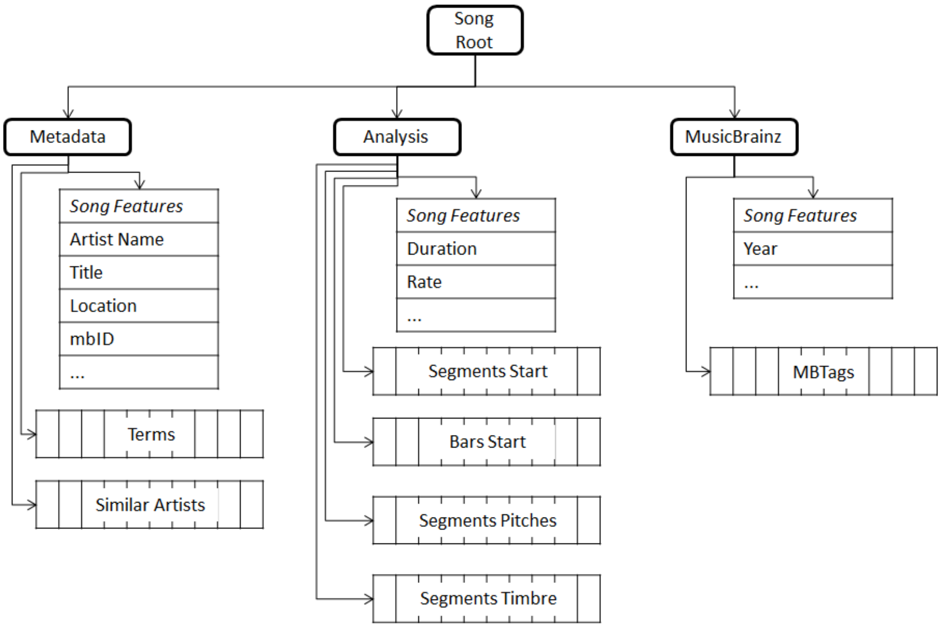


Figure 3: A visualization of the structure of each HDF5 file of the MSD dataset.

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 $\{mean_{segP\_list_1}, ..., mean_{segP\_list_{12}}\}$;
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.

[5] http://musicbrainz.org/

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.

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[6]. 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.

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

3.2 Natural Datasets

We used six natural datasets in our experiments, also used in [5][7]: Emotions, Genbase, Scene, Yeast, Enron e Medical. Table 2 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. It is worth to mention that we extended the experiments described in [5] to appropriately analyze the impact of cardinality and density measures to both natural and MSD-based datasets.

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

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

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

Figures 4 and 5 show, respectively, cardinality and density values of each dataset used in this work. We can observe in Figure 5 that density values of the MSD-based datasets are much lower than density values of the natural datasets.

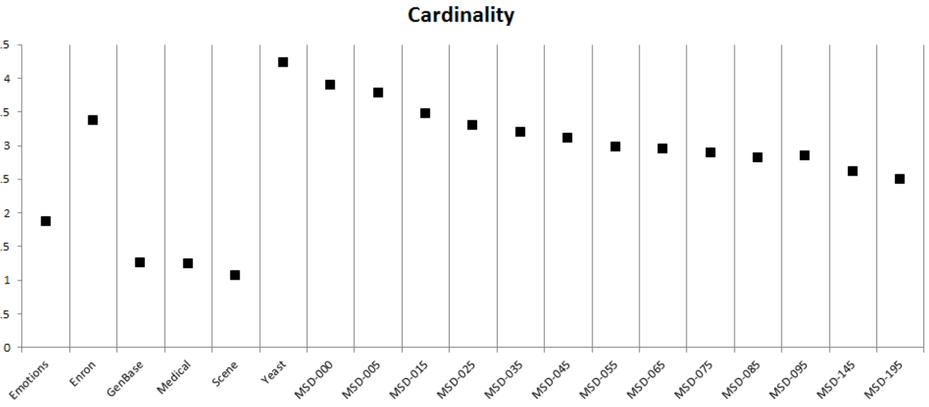


Figure 4: Cardinality Values of Each Dataset

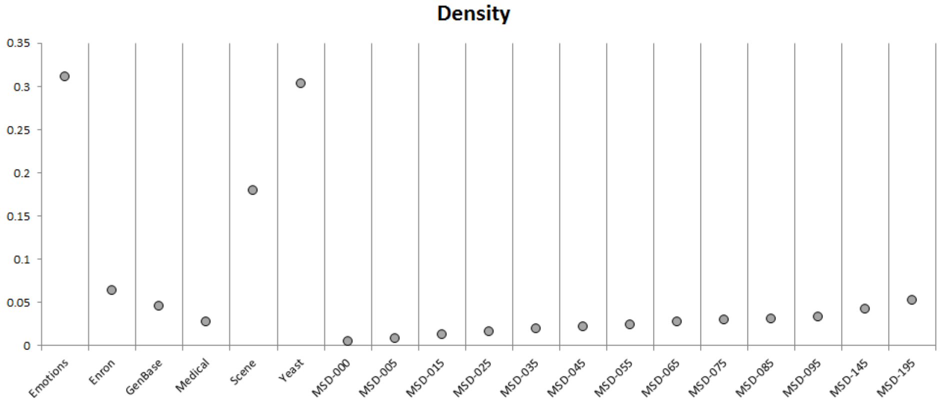


Figure 5: Density Values of Each Dataset

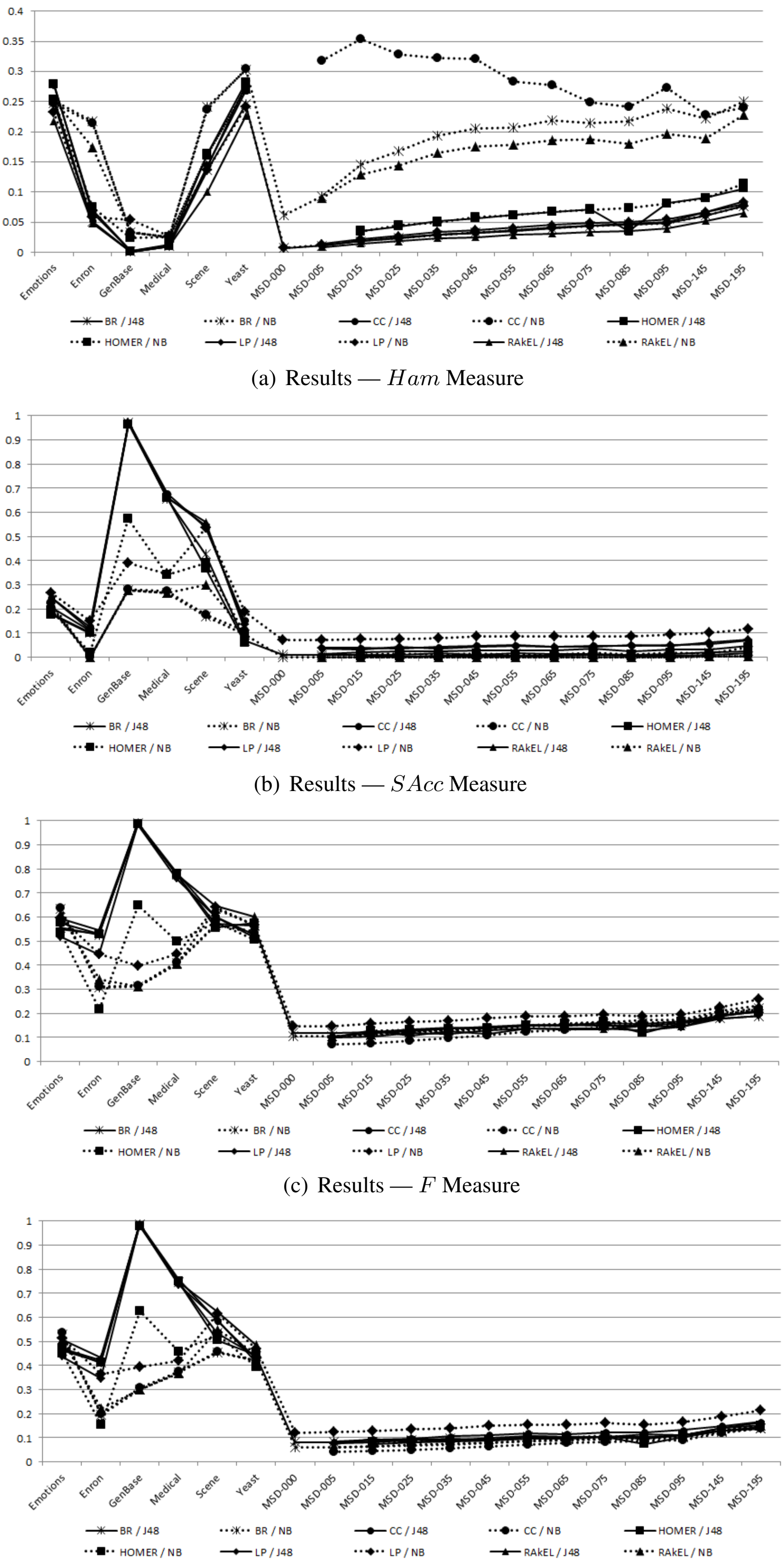
4 EXPERIMENTS, RESULTS AND ANALYSIS

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 [7] and CC [22]. To induce the multi-label classifiers, we used the Mulan library[8] [12], based on Weka [13]. To induce the base classifiers, we used Naïve Bayes and J48 algorithms, also implemented in Weka, due to its low time consumption for induction of the classifiers and lack of requirement for parameters adjustment. We denote each combination of multi-label learning method and base learning algorithm as BR-NB, BR-J48, CC-J48, CC-NB, HOMER-J48, HOMER-NB, LP-J48, LP-NB, RAKEL-J48 and RAKEL-NB. Figures 6 and 7 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 MSD-000 and MSD-005 datasets. All of these executions could not be terminated by lack of memory problem.

Figures 6 and 7 shows that the multi-label learning methods presents low performance for all measures when using MSD-based datasets. We believe that this is because many features were extracted from the original MSD dataset, but also a very large number of labels. However, yet is necessary to evaluate the influence of $Card$ and $Dens$ on multi-label learning methods, mainly in these cases, which brought us to this study.

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

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



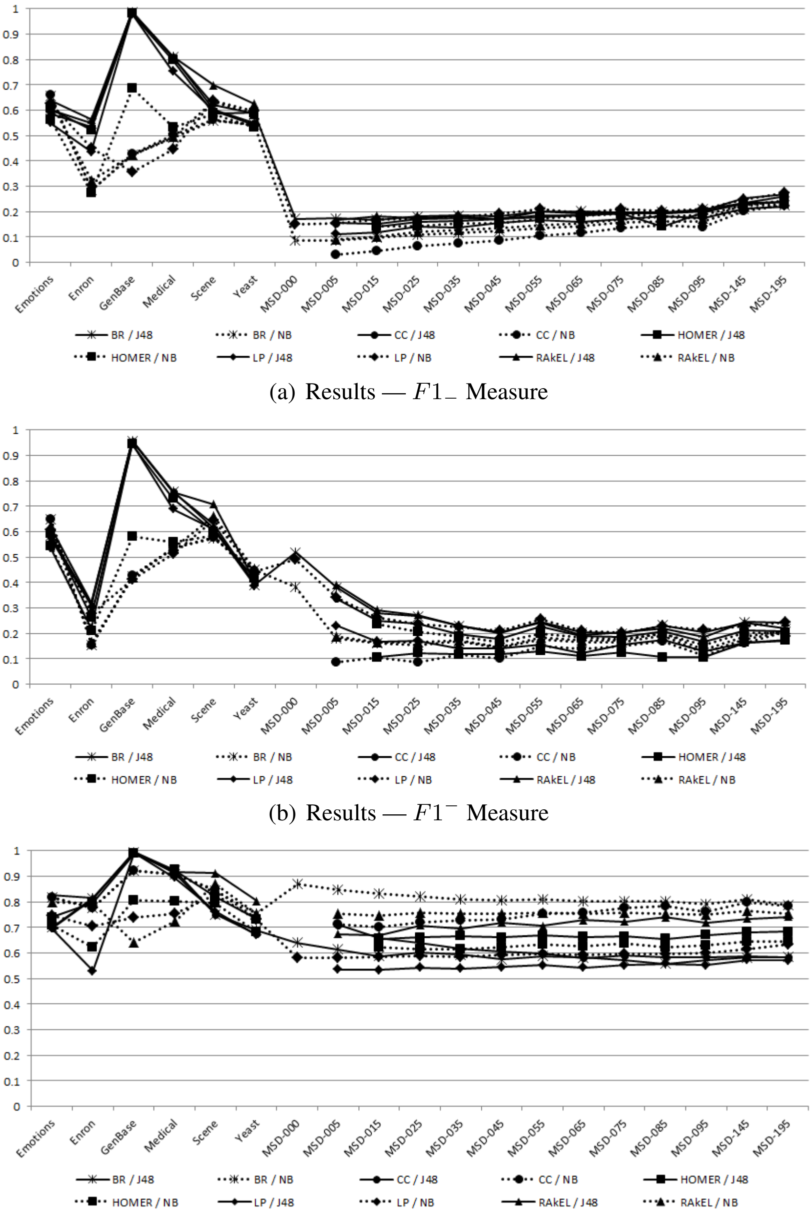
(a) Results — Ham Measure

(b) Results — $SAcc$ Measure

(c) Results — F Measure

(d) Results — Acc Measure

Figure 6: Results for instance-based measures $Mea \in \{Ham, SAcc, F, Acc\}$, all datasets and all multi-label learning methods.



(a) Results — $F1_-$ Measure

(b) Results — $F1^-$ Measure

(c) Results — AUC_- Measure

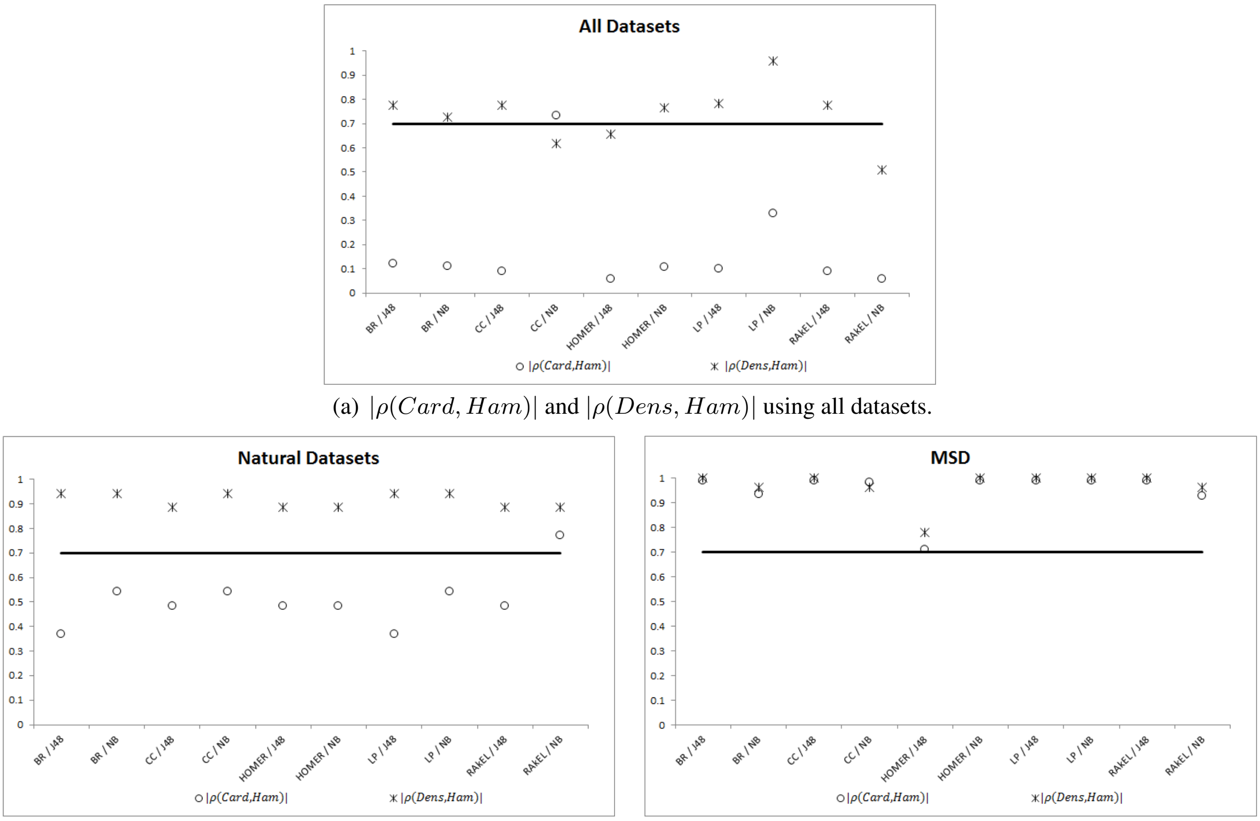
Figure 7: Results for label-based measures $Mea \in \{F1_-, F1^- \; AUC_-\}$, all datasets and all multi-label learning methods.

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] [25].

Spearman's rank correlation assesses how well the relationship between two variables X and Y and can be described using a monotonic function. If there are no repeated data values, a perfect Spearman correlation of +1 or -1 occurs when each of the variables is a perfect monotone function of the other. For a sample of size N from X and Y, the N scores X_i, Y_i are converted to ranks r_{X_i}, r_{Y_i}, and $\rho(X,Y)$ is computed as shown in Equation 10, where $d_i = r_{X_i} - r_{Y_i}$, is the difference between ranks.

$$\rho(X,Y) = 1 - \frac{6\sum d_i^2}{N(N^2-1)} \qquad (10)$$

Spearman's rank correlation was firstly calculated between $Card$ and each measure results, and also was calculated between $Dens$ and each measure results using all datasets. Correlation between the results and $Card$, as well as between the results and $Dens$, was expected. However, as we explain later in this section, we observed that many situations where correlations were observed in [5] could not be observed in our results. We also calculated Spearman's rank correlation between $Card$ and each measure results, and also was calculated between $Dens$ and each measure results using (i) only natural datasets; and (ii) only MSD-based datasets. We extended the experiments shown in [5] to also consider multi-label methods and base learning algorithms that were not considered before to appropriately analyze the behavior of natural and MSD-based datasets. Figures 8 to 14 shows the $|\rho(Card, Mea)|$ and $|\rho(Dens, Mea)|$ values for measures $Mea \in \{Ham, SAcc, F, Acc, F1_-, F1^- \; AUC_-\}$ using (a) all datasets, (b) using only natural datasets, and (c) using only MSD-based datasets.



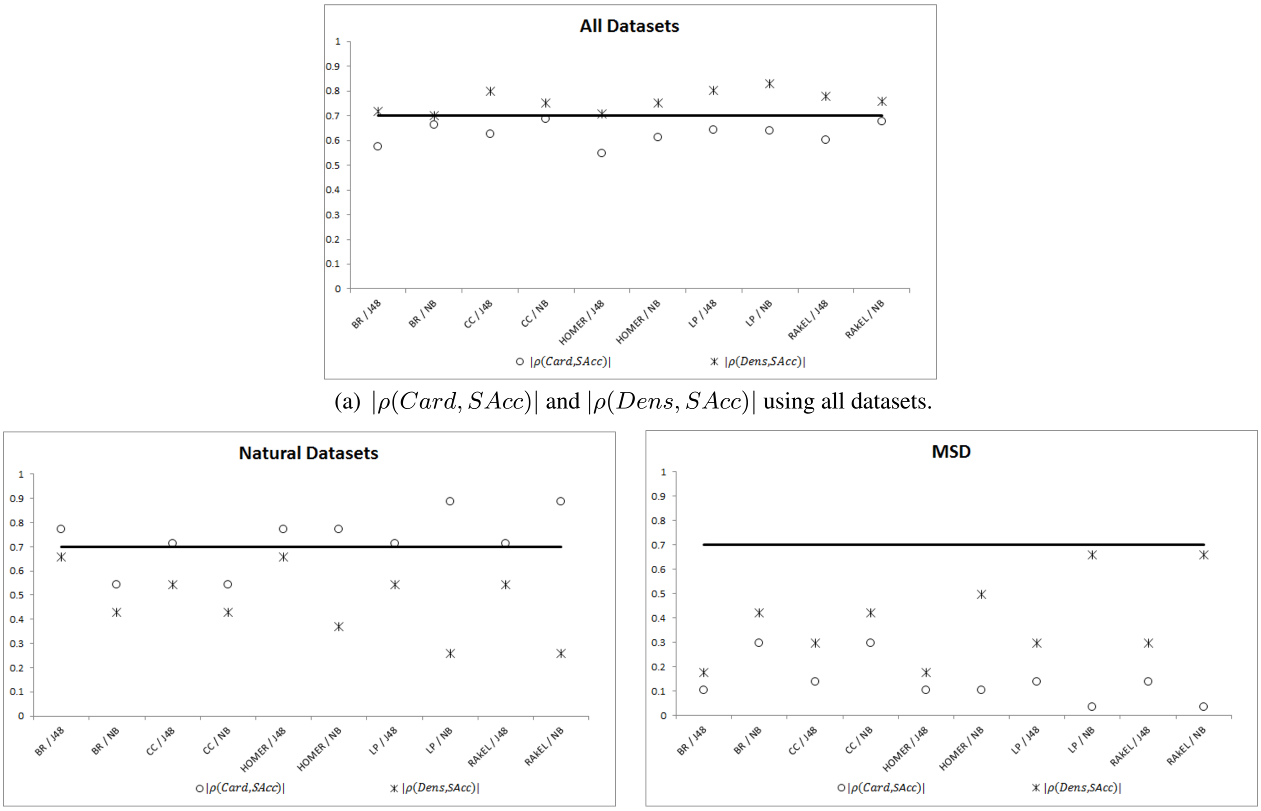
(a) $|\rho(Card, Ham)|$ and $|\rho(Dens, Ham)|$ using all datasets.

(b) $|\rho(Card, Ham)|$ and $|\rho(Dens, Ham)|$ using only natural datasets.

(c) $|\rho(Card, Ham)|$ and $|\rho(Dens, Ham)|$ using only MSD-based datasets.

Figure 8: $|\rho(Card, Ham)|$ and $|\rho(Dens, Ham)|$ values for all test scenarios.

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

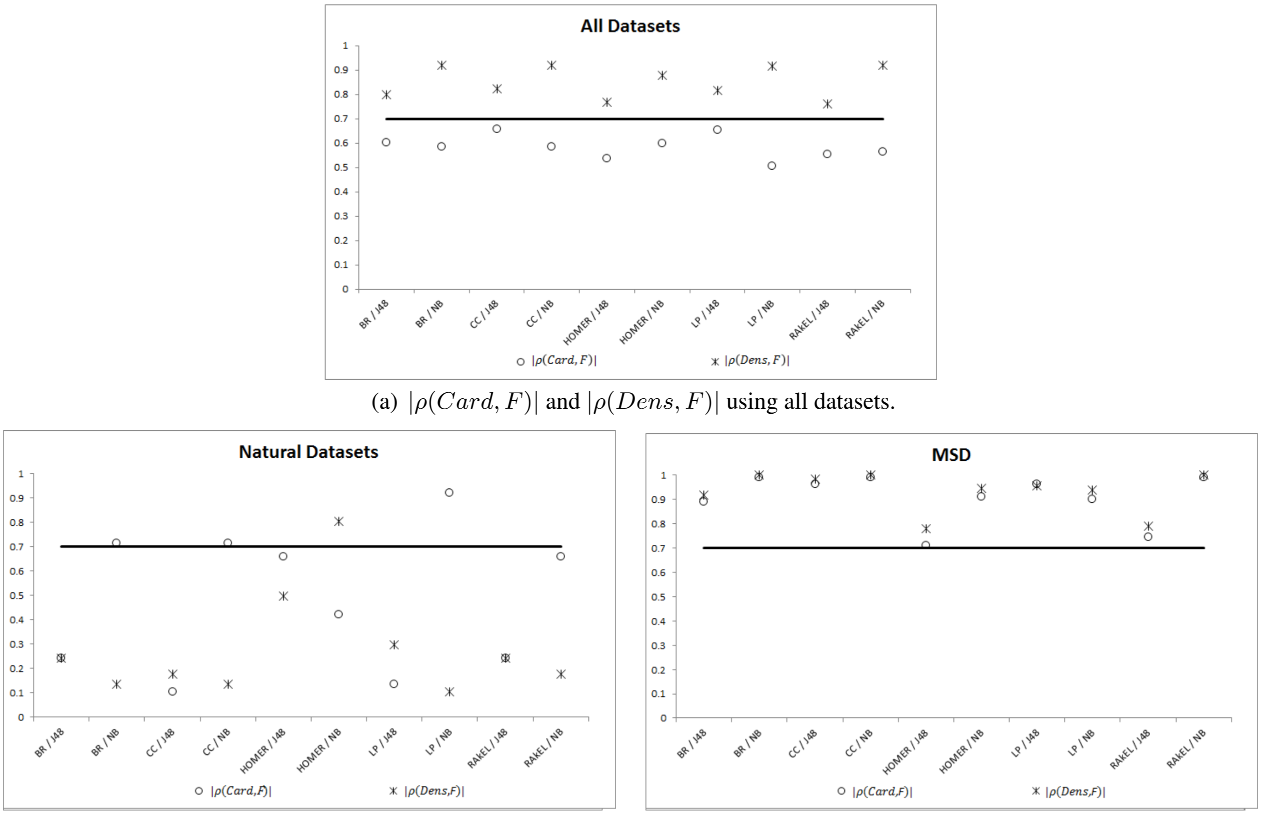


(a) $|\rho(Card, SAcc)|$ and $|\rho(Dens, SAcc)|$ using all datasets.

(b) $|\rho(Card, SAcc)|$ and $|\rho(Dens, SAcc)|$ using only natural datasets.

(c) $|\rho(Card, SAcc)|$ and $|\rho(Dens, SAcc)|$ using only MSD-based datasets.

Figure 9: $|\rho(Card, SAcc)|$ and $|\rho(Dens, SAcc)|$ values for all test scenarios.

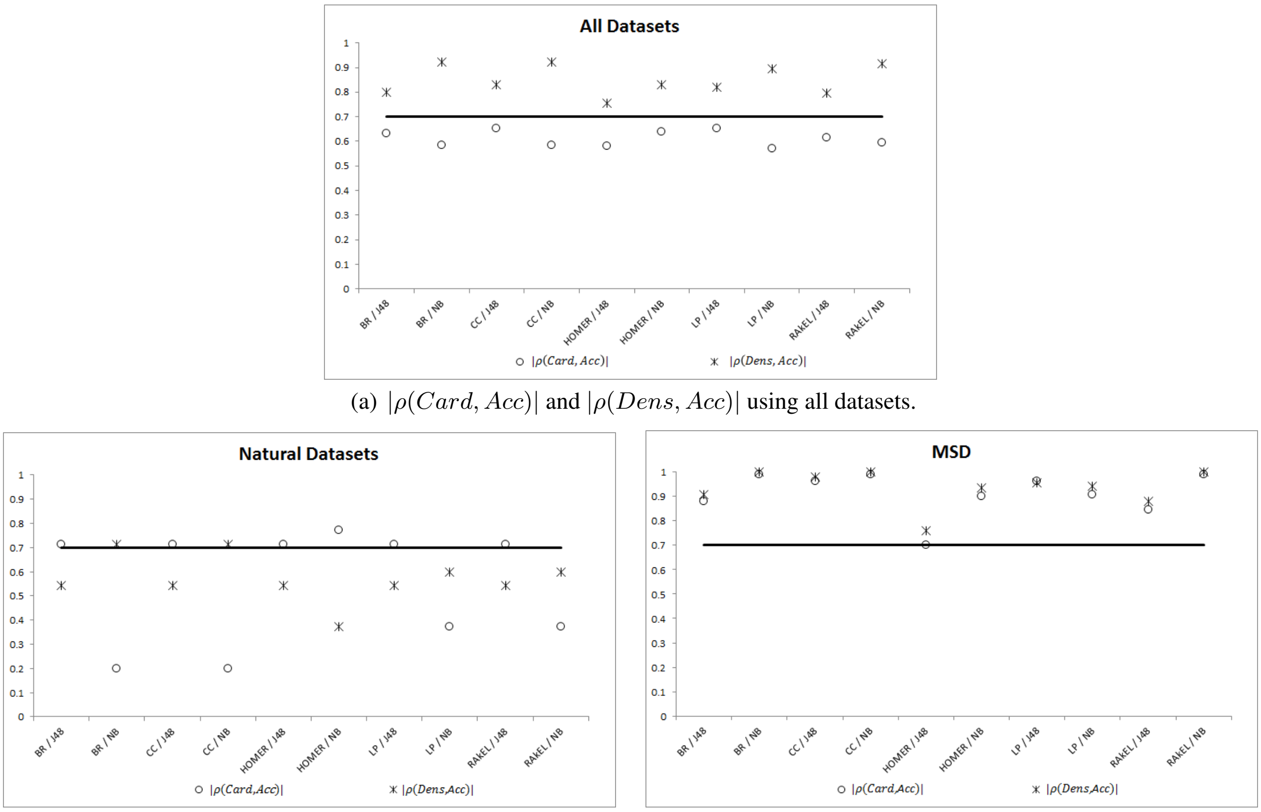


(a) $|\rho(Card, F)|$ and $|\rho(Dens, F)|$ using all datasets.

(b) $|\rho(Card, F)|$ and $|\rho(Dens, F)|$ using only natural datasets.

(c) $|\rho(Card, F)|$ and $|\rho(Dens, F)|$ using only MSD-based datasets.

Figure 10: $|\rho(Card, F)|$ and $|\rho(Dens, F)|$ values for all test scenarios.

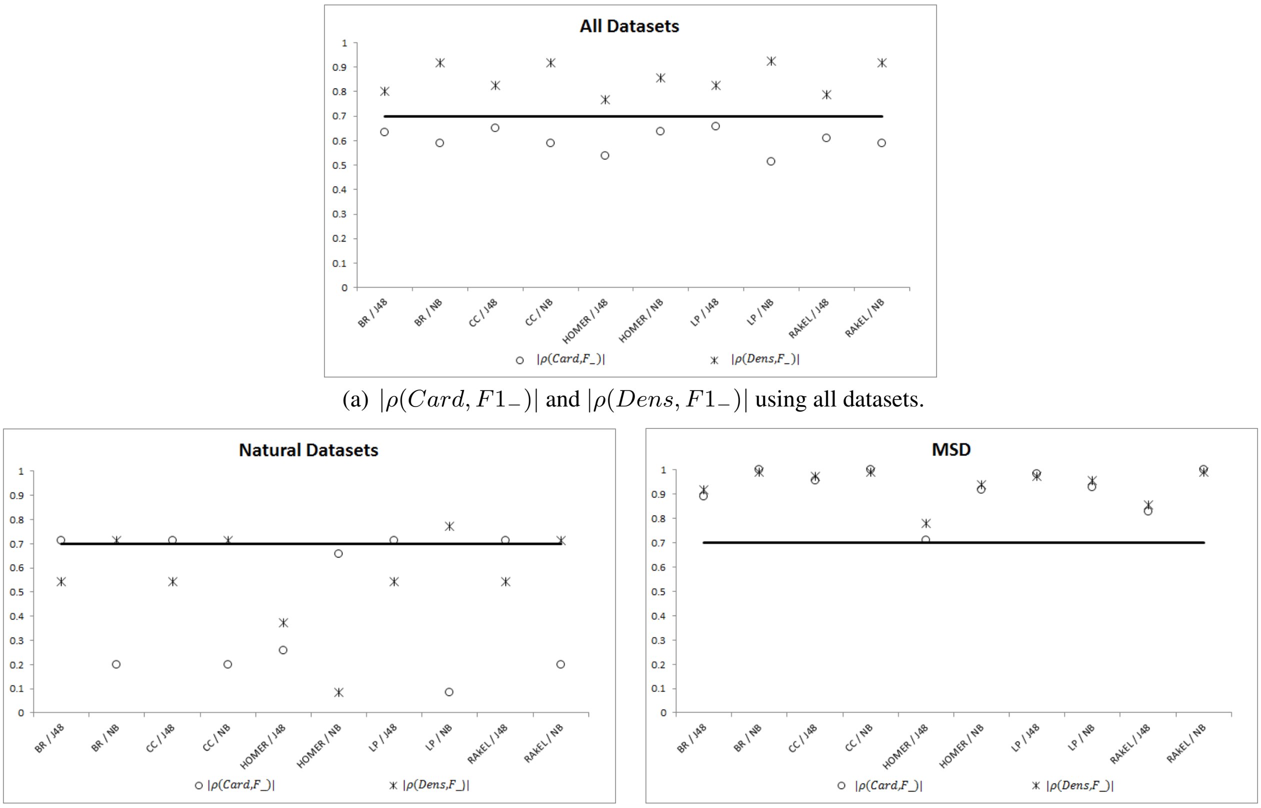


(a) $|\rho(Card, Acc)|$ and $|\rho(Dens, Acc)|$ using all datasets.

(b) $|\rho(Card, Acc)|$ and $|\rho(Dens, Acc)|$ using only natural datasets.

(c) $|\rho(Card, Acc)|$ and $|\rho(Dens, Acc)|$ using only MSD-based datasets.

Figure 11: $|\rho(Card, Acc)|$ and $|\rho(Dens, Acc)|$ values for all test scenarios.

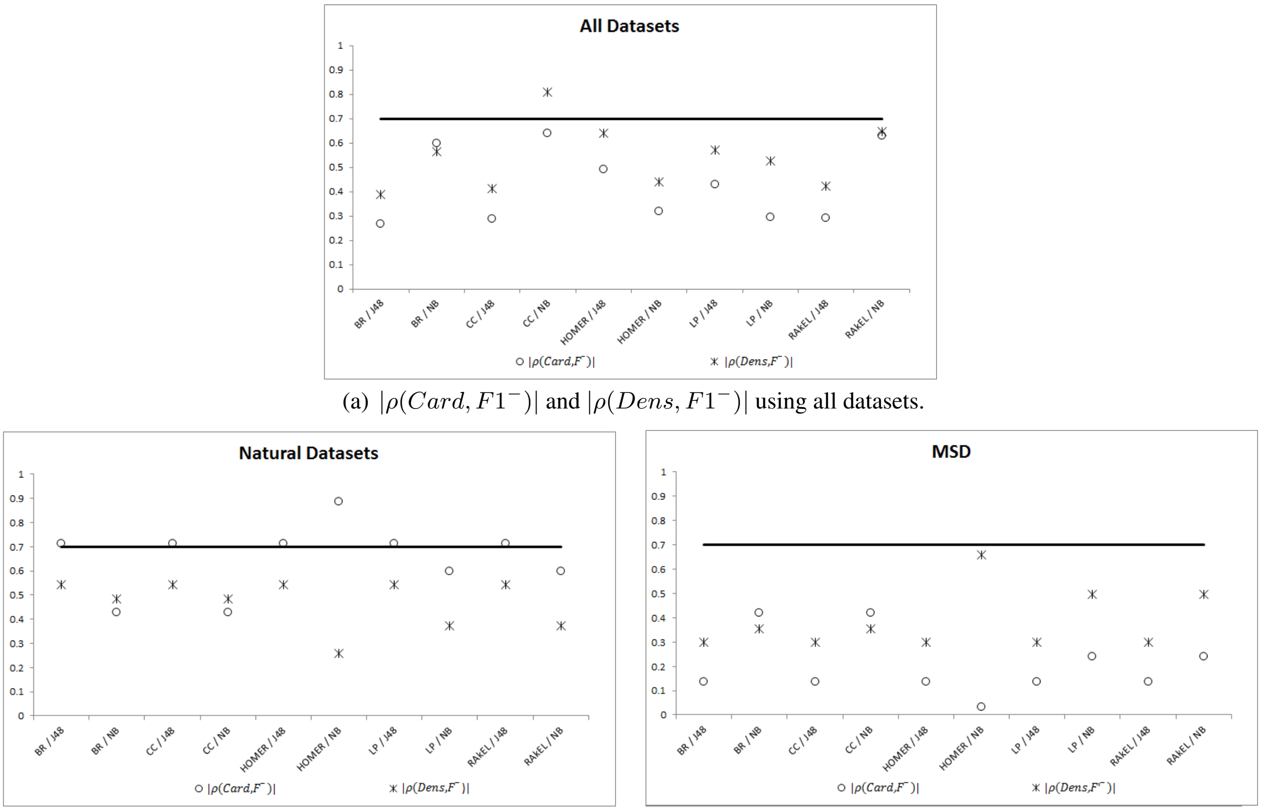


(a) $|\rho(Card, F1_{-})|$ and $|\rho(Dens, F1_{-})|$ using all datasets.

(b) $|\rho(Card, F1_{-})|$ and $|\rho(Dens, F1_{-})|$ using only natural datasets.

(c) $|\rho(Card, F1_{-})|$ and $|\rho(Dens, F1_{-})|$ using only MSD-based datasets.

Figure 12: $|\rho(Card, F1_{-})|$ and $|\rho(Dens, F1_{-})|$ values for all test scenarios.

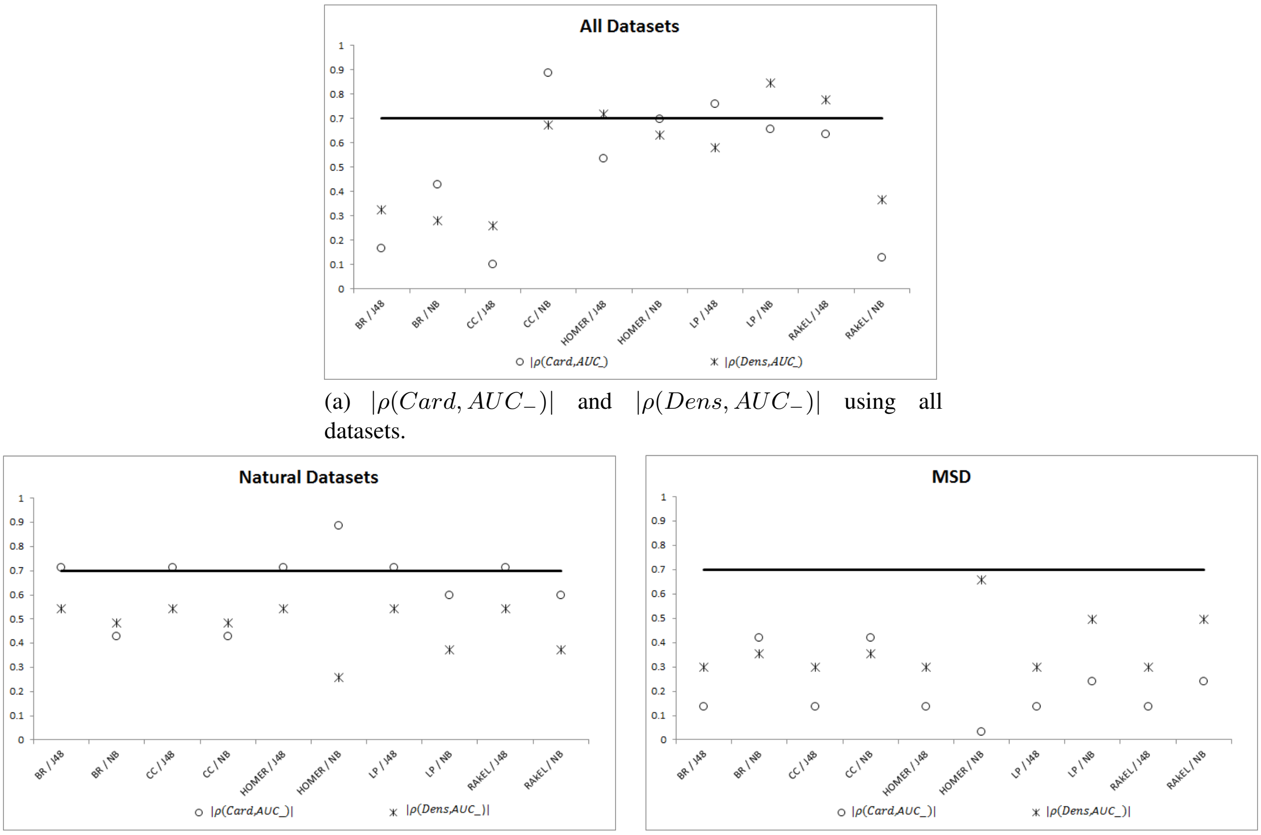


(a) $|\rho(Card, F1^{-})|$ and $|\rho(Dens, F1^{-})|$ using all datasets.

(b) $|\rho(Card, F1^{-})|$ and $|\rho(Dens, F1^{-})|$ using only natural datasets.

(c) $|\rho(Card, F1^{-})|$ and $|\rho(Dens, F1^{-})|$ using only MSD-based datasets.

Figure 13: $|\rho(Card, F1^{-})|$ and $|\rho(Dens, F1^{-})|$ values for all test scenarios.



(a) $|\rho(Card, AUC_{-})|$ and $|\rho(Dens, AUC_{-})|$ using all datasets.

(b) $|\rho(Card, AUC_{-})|$ and $|\rho(Dens, AUC_{-})|$ using only natural datasets.

(c) $|\rho(Card, AUC_{-})|$ and $|\rho(Dens, AUC_{-})|$ using only MSD-based datasets.

Figure 14: $|\rho(Card, AUC_{-})|$ and $|\rho(Dens, AUC_{-})|$ values for all test scenarios.

In what follows, we analyse the influence of $Card$ and $Dens$ on (i) MSD and natural datasets together, (ii) only natural datasets, and (iii) only MSD-based datasets.

4.1 Impact of cardinality dataset measure considering natural datasets

In [5], the authors observed that, for BR and LP methods using three base learning classifiers, the absolute value of correlation between $Card$ and $SAcc$ measure was higher than 0.7 ($|\rho(Card, SAcc)| \geq 0.7$) in all cases. It is worth to observe that they used Pearson's Correlation, which may lead to inconsistences when comparing to our work. In this work, as mentioned before, we used Spearman's rank correlation. Figure 9(b) shows that $|\rho(Card, SAcc)| \geq 0.7$ is observed when: (i) using BR and CC methods, and using J48 but not NB learning algorithm; and (ii) using HOMER, LP and RAkEL and using both J48 and NB learning algorithms. These results indicate that $Card$ measure influences $SAcc$ values in multi-label learning methods. In [5], the authors also observed, for BR and LP methods, using three base learning classifiers, that $|\rho(Card, AUC_{-})| \geq 0.7$ in one case for BR and in two cases for LP. In Figure 14(b) we can observe that $|\rho(Card, AUC_{-})| \geq 0.7$ is observed when using: (i) J48 base learning algorithm and all multi-label learning methods; (ii) NB algorithm and only HOMER multi-label learning method. These results indicate that $Card$ measure influences AUC_{-} values only when using J48 base learning algorithm.

Continuing the analyses of impact of $Card$ measures using only the natural datasets, we can observe that:

- Ham measure is not influenced by $Card$ ($|\rho(Card, Ham)| < 0.7$ in all cases — Figure 8(b));
- F measure is influenced by $Card$ when using BR, CC and LP when using NB ($|\rho(Card, F)| \geq 0.7$ in these cases — Figure 10(b));
- Acc and $F1^{-}$ measure are influenced by $Card$ when using J48 base learning algorithm for all multi-label learning methods, and NB algorithm for only HOMER multi-label learning method, as occurring with AUC_{-} ($|\rho(Card, Acc)| \geq 0.7$ in these cases — Figures 11(b) and 13(b), respectively);
- $F1_{-}$ measure is influenced by $Card$ when using J48 base learning algorithm for all but HOMER multi-label learning methods ($|\rho(Card, Acc)| \geq 0.7$ in these cases — Figure 12(b)).

4.2 Impact of density dataset measure considering natural datasets

In [5], only Ham measure exhibited high correlation with $Dens$ for all measures and all base classifiers. In our results, high correlation can be observed between Ham and $Dens$, as can be observed in Figure 8(b). Measures $SAcc$, $F1^{-}$ and AUC_{-} are not influenced by $Dens$, because high correlation between $Dens$ and each of these measures could not be observed for any multi-label method and base-learning algorithm, as can be observed in Figures 9(b), 13(b) and 14(b). High correlation with $Dens$ was punctually observed for:

- F measure when using HOMER and NB — Figure 10(b);
- Acc measure when using BR and CC with NB as base learning algorithm — Figure 11(b)); and
- $F1_{-}$ measure when using BR, CC, LP and RAkEL with NB as base learning algorithm — Figure 12(b)).

4.3 Impact of cardinality and density measures considering MSD-based datasets

For MSD-based datasets, it is interesting to notice that when high (or low) absolute correlation value between $Dens$ and each classifier evaluation measure is observed, the same is observed between $Card$ and each classifier evaluation measure. We can observe that the measures influenced by cardinality and density dataset measures are Ham, F, $Acc$and $F1_{-}$ — all these measures are highly correlated to $Card$ and to $Dens$ dataset measure, as can be seen in Figures 8(c), 10(c), 11(c) and 12(c). We also can observe that the measures not influenced by cardinality and density dataset measure are $SAcc$, $F1^{-}$ and AUC_{-} — all these measures are weally correlated to $Card$ and $Dens$ dataset measure, as can be seen in Figures 9(c), 13(c), 14(c).

4.4 Impact of cardinality and density measures considering all datasets

When analyzing the results considering all datasets, we could not observe high correlation between Cardinality values and measures results for all but four situations. These exceptions can be observed in Figure 8(a) for CC-NB method, and in Figure 14(a) for CC-NB, HOMER-NB and LP-NB. However, for $SAcc$, F and Acc measures, beside all correlations between $Card$ and Mea ($\rho(Card, Mea)$) are lower than 0.7, they are near to 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 8(a), 9(a), 10(a),11(a) and 12(a), all correlations, but three, are greater than 0.7. The exception are for Ham measure and CC-NB, HOMER-J48 and RAKEL-NB methods. This observation indicates that $Dens$ highly influences the results in these measures.

We also 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 the mentioned measures. Finally, we also observed that, for $F1^{-}$ and AUC_{-} measures, we could not observe any pattern in correlation behaviour.

5 CONCLUSIONS AND FUTURE WORK

Cardinality and density are characteristics of multi-label datasets 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], the authors started an investigation on how much cardinality and density could impact the results of multi-label learning methods. They used only six natural datasets, available on 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 natural datasets used before, extending the experiments to be comparable to the results obtained with MSD-based datasets. All of the obtained results compose our analyzes. We observed in our results that, when analyzing the impact of $Card$ and $Dens$ on multi-label learners only using the natural datasets, with distinct domains, $Card$ influences results of $SAcc$, F, $F1_{-}$ and AUC_{-} measures; and $Dens$ influences results of Ham, F, Acc and $F1_{-}$ measures. When analyzing the impact of $Card$ and $Dens$ only using the MSD based datasets, both $Card$ and $Dens$ influences the results of Ham, F, Acc and $F1_{-}$ measures. On the other hand, when we put all these datasets together to analyze the $Card$ and $Dens$ influence, we do not verify the same influence patterns, although some influences are worth to notice. $Card$ barely influenced the results on the used measures, but $Dens$ highly influenced the results of Ham, $SAcc$, F, Acc and $F1_{-}$ measures when using all the datasets. This bring us evidences, as expected, that density characteristic should be carefully treated when using multi-label datasets with low density values. In this way, exploring how to increase density values without changing the learning problems could be an interesting approach.

Another important observation in our experiments is that, on one hand, the MSD-based datasets have large number of features and large number of labels, and on the other hand the natural datasets have low number of features and low number of labels when compared to the MSD-based datasets. Also, the results obtained with the multi-label learning methods using the natural datasets were better than the results obtained using the MSD-based datasets. The performance of the multi-label learning methods using the MSD-based datasets may have been low due to the characteristics of these datasets. So, it is not yet clear what characteristics of the datasets really lead to the influence of low $Dens$ values on measures results. To this end, we are conducting experiments to evaluate the influence of $Card$ and $Dens$ values using artificially generated multi-label datasets [26].

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; Prof. Ana Paula Sobral (UFF) for helping us on the evaluation process of this paper; and the anonymous referees for their valuable observations.

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] M.-L. Zhang and Z.-H. Zhou. “A Review on Multi-Label Learning Algorithms”. *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.

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

[8] H.-F. Yu, P. Jain, P. Kar and I. Dhillon. “Large-scale Multi-label Learning with Missing Labels”. In *Proc. The 31st International Conference on Machine Learning*, pp. 593–601, 2014.

[9] W. Bi and J. Kwok. “Efficient Multi-label Classification with Many Labels”. In *Proc. 30th International Conference on Machine Learning*, pp. 405–413, 2013.

[10] 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.

[11] 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.

[12] 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.

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

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

[15] G. Madjarov, D. Kocev, D. Gjorgjevikj and S. Džeroski. “An extensive experimental comparison of methods for multi-label learning”. *Pattern Recognition*, vol. 45, no. 9, pp. 3084–3104, 2012.

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

[17] E. Alvares-Cherman, J. Metz and M. C. Monard. “Incorporating label dependency into the binary relevance framework for multi-label classification”. *Expert Systems with Applications*, vol. 39, no. 2, pp. 1647–1655, 2012.

[18] G. Tsoumakas, I. Katakis and L. Vlahavas. “Random k-Labelsets for Multilabel Classification”. *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, 2011.

[19] 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.

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

[21] G. Tsoumakas, I. Katakis and I. Vlahavas. “Effective and efficient multilabel classification in domains with large number of labels.” In *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD’08)*, pp. 30–44, 2008.

[22] 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.

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

[24] 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.

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

[26] J. T. Tomás, N. Spolaôr, E. A. Cherman and M. C. Monard. “A Framework to Generate Synthetic Multi-label Datasets”. *Electronic Notes in Theoretical Computer Science*, vol. 302, pp. 155–176, 2014.