A Tutorial on Multilabel Learning

EVA GIBAJA and SEBASTIÁN VENTURA, Department of Computer Science and Numerical Analysis, University of Córdoba, Spain

Multilabel learning has become a relevant learning paradigm in the past years due to the increasing number of fields where it can be applied and also to the emerging number of techniques that are being developed. This article presents an up-to-date tutorial about multilabel learning that introduces the paradigm and describes the main contributions developed. Evaluation measures, fields of application, trending topics, and resources are also presented.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications—*Data mining*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models*; I.2.6 [**Artificial Intelligence**]: Learning—*Concept learning, connectionism and neural nets, induction*; I.7.5 [**Document and Text Processing**]: Document Capture—*Document analysis*; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*Object recognition*; I.5.2 [**Pattern Recognition**]: Design Methodology—*Classifier design and evaluation*; I.5.4 [**Pattern Recognition**]: Applications—*Computer vision, text processing*

General Terms: Algorithms, Experimentation, Theory

Additional Key Words and Phrases: Multilabel learning, ranking, classification, machine learning, data mining

ACM Reference Format:
Eva Gibaja and Sebastián Ventura. 2015. A tutorial on multilabel learning. ACM Comput. Surv. 47, 3, Article 52 (April 2015), 38 pages.
DOI: http://dx.doi.org/10.1145/2716262

1. INTRODUCTION

Classification is one of the main tasks in data mining. Given a set of training patterns consisting of a set of features and an associated class, the aim of classification is to obtain a model that will be able to assign the proper class to an unknown pattern. This formulation of the problem entails the restriction of *only one label per pattern*; nevertheless, there are increasing numbers of classification problems being contemplated today, such as text and sound categorization, semantic scene classification, medical diagnosis, or gene and protein function classification in which a pattern can have several labels simultaneously associated. For instance, in the field of semantic scene classification, a picture containing a landscape with both a beach and a mountain could be associated with *beach* and *mountain* categories simultaneously. This type of problem is called *multilabel* in comparison with classical supervised learning (also called

This work is supported by the Ministry of Science and Technology project TIN-2011-22408.
Authors' addresses: E. Gibaja and S. Ventura, Department of Computer Science and Numerical Analysis, University of Córdoba, Spain. Dr. Ventura also belongs to the Computer Sciences Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia; emails: {egibaja, sventura}@uco.es.

DOI: http://dx.doi.org/10.1145/2716262

single-label [Schapire and Singer 2000]). Solving a problem with multilabel data involves new challenges due to the exponential growth of combinations of labels to take into account and also to the computational cost of building and querying the models. In addition, multilabel data usually present features such as high dimensionality, unbalanced data, and dependences between labels.

Thus, during the past years, the paradigm of *Multilabel Learning* (MLL) has arisen as a kind of supervised learning and has become a very hot topic. Despite the publication of some works compiling the basis of MLL [Tsoumakas and Katakis 2007; de Carvalho and Freitas 2009; Tsoumakas et al. 2010a; Zhang and Zhou 2014], it is not possible to find an up-to-date tutorial on MLL. The aim of this article is to cover, among other issues, the formal definition of the problem, the domains where MLL has been applied, an up-to-date summary of the main proposals presented during latest years, evaluation measures, and resources. This article is organized as follows. In Section 2, the MLL problem is formally defined. After that, in Section 3, some aspects related to the development and evaluation of MLL models are described. The main approaches developed in the literature are presented in Section 4. Next, Section 5 describes findings on empirical comparisons between MLL algorithms. Section 6 describes the main domains where MLL has been applied, and, finally, new trends in MLL (Section 7) and a set of conclusions are presented. The article also includes an Appendix with resources (software, datasets, etc.) for MLL.

2. MULTILABEL LEARNING

2.1. MLL Settings

According to Read [2010] a multilabel problem has the following settings:

(1) The set of labels is predefined, meaningful, and human-interpretable.
(2) The number of labels is limited in scope and not greater than the number of attributes.
(3) Each training example is associated with several labels of the label set.
(4) The number of attributes may be large, but attribute-reduction strategies can be employed in these cases.
(5) The number of examples may be large.

The two last settings are related with the high dimensionality of data, a common feature of many multilabel datasets. It is also worth noting two other features:

(6) Labels may be correlated. As an example, Figure 1 shows in a graph the 10 most frequent labels of a multilabel dataset, the so-called *imdb* [Read 2010] (more information about this dataset is found in the Appendix). This dataset, contains 120,919 movie plot text summaries from the imdb database and has 28 labels corresponding with genres (e.g., *comedy*, *action,* etc.). Node thickness represents prior probability of the label, and edge thickness indicates co-occurrence of the two linked labels. It is observed that *talk show* and *war* labels are not related, whereas there are relationships with different strength between the other ones, for instance, *action* and *crime* are more related than *mystery* and *film noir*. These relationships between labels represent additional knowledge that can be explored to facilitate the learning process.
(7) Data may be unbalanced. This can be seen from two points of view. On one hand, if each particular label is considered, the number of patterns belonging to a certain label may outnumber other labels (interclass). For instance, Figure 1 shows that *comedy* is much more frequent than *film noir*. In addition, the proportion of positive to negative examples for each class may be unbalanced (inner or intraclass). On the other hand, *label skew* may be defined as a relative high number of examples

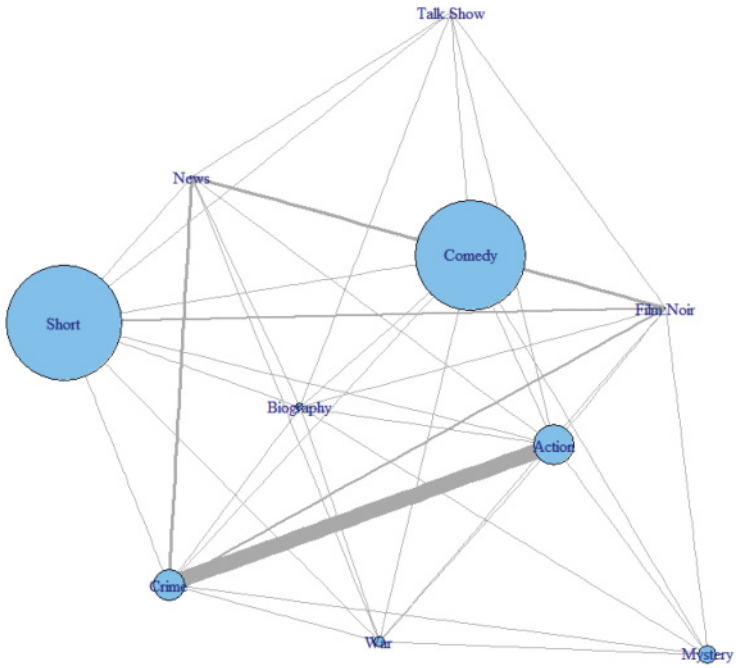

Fig. 1. Co-occurrence graph of labels in the imdb dataset.

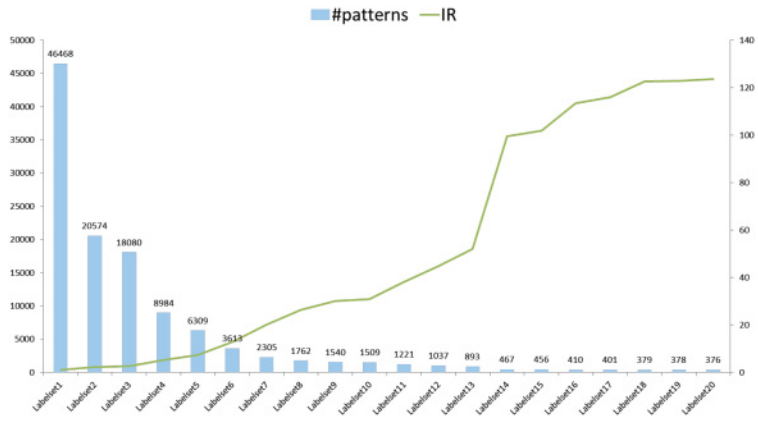

Fig. 2. Examples and Imbalance Ratio (IR) per labelset in the imdb dataset.

associated with the most common labelsets, whereas a relatively high number of examples are associated with infrequent labelsets [Read 2010]. Figure 2 represents the number of examples per labelset and the *Imbalance Ratio* (IR), showing that the dataset of the example is clearly unbalanced. IR has been computed for each labelset as the quotient between the size of the most frequent labelset and the size of the labelset.

2.2. A Formal Definition of MLL

This section is based on the notations defined by Schapire and Singer [2000], Zhang and Zhou [2005], and Brinker et al. [2006]. Let these be the following definitions:

Table I. An Example of a Single-Label vs. a Multilabel Dataset

		Single-label binary	Single-label multiclass	Multilabel output				
Example	Features	$Y \in \mathcal{L} = \{0, 1\}$	$Y \in \mathcal{L} = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \}$	y_1	y_2	y_3	y_4	$Y \subseteq \mathcal{L} = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$
1	$\overline{\mathbf{x_1}}$	1	λ_2	1	1	0	1	$\{\lambda_1, \lambda_2, \lambda_4\}$
2	$\overline{\mathbf{x_2}}$	0	λ_4	0	0	0	1	$\{\lambda_4\}$
3	$\overline{\mathbf{x_3}}$	0	λ_3	0	1	1	1	$\{\lambda_2, \lambda_3, \lambda_4\}$
4	$\overline{\mathbf{x_4}}$	1	λ_1	1	0	1	0	$\{\lambda_1, \lambda_3\}$
5	$\overline{\mathbf{x_5}}$	0	λ_3	0	1	1	0	$\{\lambda_2, \lambda_3\}$
				2	3	3	3	◁Count

—$\mathcal{X}$ is a d-dimensional input space of numerical or categorical features.
—$\mathcal{L} = \{\lambda_1, \lambda_2, \ldots, \lambda_q\}$ an output space of q labels, $q > 1$. Each subset of $\mathcal{L}$ is called *labelset*.
—$(\mathbf{x}, Y)$, where $\mathbf{x} = (x_1, \ldots, x_d) \in \mathcal{X}$, is a d-dimensional instance that has a set of labels associated $Y \subseteq \mathcal{L}$. Label associations can be also represented as a q-dimensional binary vector $\mathbf{y} = (y_1, y_2, \ldots, y_q) = \{0, 1\}^q$ where each element is 1 if the label is relevant and 0 otherwise. Table I shows an example of a multilabel dataset compared with a single-label binary one. As can be observed, in single-label (binary or multiclass) learning $|Y| = 1$.
—$S = \{(\mathbf{x}_i, Y_i) | 1 \leq i \leq m\}$ is a multilabel training set with m examples.

According to Tsoumakas et al. [2010a], MLL includes two main tasks: *Multilabel Classification* (MLC) and *Label Ranking* (LR). MLC consists of defining a function $h_{\text{MLC}} : \mathcal{X} \rightarrow 2^{\mathcal{L}}$. Therefore, given an input instance, a multilabel classifier will return a set of relevant labels, Y, and the complement of this set, $\overline{Y}$, the set of irrelevant labels. So, a bipartition of the set of labels into relevant and irrelevant sets is obtained. Multiclass classification can be seen as a particular case of MLC where $h_{\text{MC}} : \mathcal{X} \rightarrow \mathcal{L}$ whereas in binary classification $h_{\text{B}} : \mathcal{X} \rightarrow \{0, 1\}$.

On the other hand, LR defines a function $f : \mathcal{X} \times \mathcal{L} \rightarrow \mathbb{R}$ that returns an ordering of all the possible labels according to the relevance of labels to a given instance $\mathbf{x}$. Thus, label λ_1 is considered to be ranked higher than λ_2 if $f(\mathbf{x}, \lambda_1) > f(\mathbf{x}, \lambda_2)$. A rank function, $\tau_{\mathbf{x}}$, maps the output real value of the classifier to the position of the label in the ranking, $\{1, 2, \ldots, q\}$. Therefore, if $f(\mathbf{x}, \lambda_1) > f(\mathbf{x}, \lambda_2)$, then $\tau_{\mathbf{x}}(\lambda_1) < \tau_{\mathbf{x}}(\lambda_2)$. The lower the position, the better the position in the ranking is. Finally, a third task in MLL, called *Multilabel Ranking* that can be seen as a generalization of MLC and LR, produces at the same time both a bipartition and a consistent ranking. In other words, if Y is the set of labels associated with an instance, $\mathbf{x}$, and $\lambda_1 \in Y$ and $\lambda_2 \in \overline{Y}$, then a consistent ranking will rank labels in Y higher than labels in $\overline{Y}$, $\tau_{\mathbf{x}}(\lambda_1) < \tau_{\mathbf{x}}(\lambda_2)$. The definition of multilabel classifier from a multilabel ranking model can be derived from the function $f(\mathbf{x}, \lambda) : h(\mathbf{x}) = \{\lambda | f(\mathbf{x}, \lambda) > t(\mathbf{x}), \lambda \in \mathcal{L}\}$, where $t(\mathbf{x})$ is a threshold function. Section 4.4 details this topic.

3. EVALUATION OF MULTILABEL MODELS

This section includes aspects to consider when an MLL method is being evaluated: evaluation metrics, how to prepare the dataset, statistical tests, and complexity.

3.1. Evaluation Metrics

The evaluation of models in MLL needs a special approach because the performance over all labels should be taken into account. In addition, a prediction could be partially correct (some of the labels are correctly predicted), fully wrong (all predictions are wrong), or fully correct (all labels are correctly predicted). In this section, the most

frequent performance metrics for MLL will be summarized and grouped according to two categories: *metrics to evaluate bipartitions* and *metrics to evaluate rankings*.

3.1.1. Metrics to Evaluate Bipartitions. Metrics to evaluate bipartitions can be classified into two groups: *label-based* and *example-based*. The former are calculated for each label and then are averaged across all labels, whereas the latter are calculated for each test example and then averaged across the test set.

LABEL-BASED METRICS

Any binary evaluation metric can be used with this type of approach (e.g., precision, recall, accuracy, or F1-score). The idea is to compute a single-label metric for each label based on the number of true positives (tp), true negatives (tn), false positives (fp), and false negatives (fn). Due to the fact of having several labels per pattern, there will be a contingency table for each label, so it is necessary to compute an average value. Two different approaches can be used: *macro* and *micro*. Let B be a binary evaluation measure; then, the macro approach computes one metric for each label, and the values are averaged over all the categories (see Equation (1))). By contrast, the micro approach considers predictions from all instances together (aggregating the values of all the contingency tables) and then calculates the measure across all labels (see Equation (2)):

$$B_{macro} = \frac{1}{q}\sum_{i=1}^{q} B(tp_i,\ fp_i, tn_i,\ fn_i). \tag{1}$$

$$B_{micro} = B\left(\sum_{i=1}^{q} tp_i, \sum_{i=1}^{q} fp_i, \sum_{i=1}^{q} tn_i, \sum_{i=1}^{q} fn_i\right). \tag{2}$$

These two types of averaging are informative, and there is no general agreement about using a macro or micro approach. According to Yang [1999] and Yang and Liu [1999], macro averaged scores give equal weight to every category, regardless of its frequency (per-category averaging) and is more influenced by the performance on rare categories. On the other hand, micro averaged scores give equal weight to every example (per-example averaging) and tend to be dominated by the performance in most common categories. In the same vein, Pestian et al. [2007] pointed out that a macro approach would be better when the system is required to perform consistently across all classes regardless of the frequency of the class (i.e., in problems where distribution of training samples across categories is skewed), whereas the micro approach may be better if the density of the class is important.

INSTANCE/EXAMPLE-BASED METRICS

Let $T = \{(\mathbf{x}_i, Y_i)|1 \leq i \leq t\}$ be a multilabel test set with t instances and Y_i and Z_i the sets of true and predicted labels for an instance. For any predicate, π, $[\![\pi]\!]$ returns 1 if the predicate is true and 0 otherwise; finally, let $\tau^*_{\mathbf{x}}$ be the true ranking.

0/1 subset accuracy [Zhu et al. 2005]. This metric, also called *classification accuracy* or *exact match ratio*, computes the percentage of instances whose predicted labels are exactly the same as their corresponding set of ground-truth labels (see Equation (3)). Because an exact match is needed, it does not distinguish between *completely incorrect* and *partially correct* predictions because it is a very strict evaluation measure. This metric must be maximized:

$$0/1\ \textit{subset accuracy} = \frac{1}{t}\sum_{i=1}^{t}[\![Z_i = Y_i]\!], \tag{3}$$

Hamming Loss [Schapire and Singer 1999] evaluates how many times, on average, an example-label pair is misclassified. This metric takes into account both prediction errors (an incorrect label is predicted) and omission errors (a correct label is not predicted). The lower the value, the better the performance of the classifier. The expression of this metric is given in Equation (4), where Δ stands for the symmetric difference of two sets, and the $1/q$ factor is used to obtain a normalized value in [0, 1].

$$Hamming\ loss = \frac{1}{t}\sum_{i=1}^{t}\frac{1}{q}|Z_i \Delta Y_i|. \tag{4}$$

In MLL, it is also common to use a group of example-based metrics from the *information retrieval* area [Godbole and Sarawagi 2004] that must be maximized: *Recall* (see Equation (5)) is the fraction of predicted correct labels of the actual labels, whereas *precision* (see Equation (6)) is the proportion of labels correctly classified of the predicted positive labels, averagedover all instances:

$$recall = \frac{1}{t}\sum_{i=1}^{t}\frac{|Z_i \cap Y_i|}{|Y_i|}. \tag{5}$$

$$precision = \frac{1}{t}\sum_{i=1}^{t}\frac{|Z_i \cap Y_i|}{|Z_i|}. \tag{6}$$

The *accuracy* (see Equation (7)) is the proportion of label values correctly classified of the total number (predicted and actual) of labels for that instance averaged over all instances:

$$accuracy = \frac{1}{t}\sum_{i=1}^{t}\frac{|Z_i \cap Y_i|}{|Z_i \cup Y_i|}. \tag{7}$$

Finally, the *F1-Score* or *harmonic mean* that combines precision and recall is defined in Equation (8):

$$F1 - score = \frac{1}{t}\sum_{i=1}^{t}\frac{2|Z_i \cap Y_i|}{|Z_i| + |Y_i|}. \tag{8}$$

3.1.2. Metrics to Evaluate Rankings. All the metrics detailed herein can be also considered example-based metrics because they are first computed for each test example, and then they are averaged across the test set.

The *One-error* [Schapire and Singer 2000] measure evaluates how many times the top-ranked label was not in the set of possible labels. The lower the value, the better the metric. It measures the probability of not getting even one of the correct labels. A priori, this metric is not a good metric for MLL because it only takes into account the top-ranked label. Note that for single-label, the one-error is equivalent to ordinal error. The expression of this metric is shown in Equation (9); the arg function returns a label, $\lambda \in \mathcal{L}$:

$$One\text{-}error = \frac{1}{t}\sum_{i=1}^{t}[\![\arg\min_{\lambda \in \mathcal{L}} \tau_i(\lambda) \notin Y_i]\!]. \tag{9}$$

Coverage [Schapire and Singer 2000]. This metric (see Equation (10)) measures the average depth in the ranking in order to cover all the labels associated with an instance. The smaller the value, the better the performance. Whereas one-error only

takes into account the performance for the top-ranked label, the coverage measures the performance for all the possible labels:

$$coverage = \frac{1}{t}\sum_{i=1}^{t}\max_{\lambda\in Y_i}\tau_i(\lambda) - 1 \tag{10}$$

Ranking loss [Schapire and Singer 1999] evaluates the average of pairs of labels that are misordered for the instance. The lower the value of the metric, the better the performance. The goal is to obtain a small number of misorderings so that the labels in Y are ranked above those in $\overline{Y}$. The $|E|$ is called in Crammer and Singer [2003] and Park and Fürnkranz [2008] *error-set-size*.

$$ranking\ loss = \frac{1}{t}\sum_{i=1}^{t}\frac{1}{|Y_i||\overline{Y_i}|}|E| where$$

$$E = \{(\lambda,\lambda')|\tau_i(\lambda) > \tau_i(\lambda'), (\lambda,\lambda') \in Y_i \times \overline{Y_i}\}. \tag{11}$$

Average precision [Schapire and Singer 2000]. Coverage and one-error are not complete metrics for MLL because it is possible to have a good coverage while having high one-error values. The average precision evaluates the average fraction of labels ranked above a particular label, $\lambda \in Y$, which actually are in Y. The performance is perfect when the value is 1. The higher the value, the better the performance:

$$avg.\ precision = \frac{1}{t}\sum_{i=1}^{t}\frac{1}{|Y_i|}\sum_{\lambda\in Y_i}\frac{|\{\lambda' \in Y_i|\tau_i(\lambda') \le \tau_i(\lambda)\}|.}{\tau_i(\lambda)} \tag{12}$$

Margin loss [Loza and Fürnkranz 2010] metric returns the number of positions between the worst-ranked positive and the best-ranked negative classes. This measure is related to the number of wrongly ranked classes and must be minimized:

$$margin\ loss = \frac{1}{t}\sum_{i=1}^{t}\max(0, \max\{\tau(\lambda)|\lambda \in Y_i\} - \min\{\tau(\lambda')|\lambda' \notin Y_i\}). \tag{13}$$

IsError [Crammer and Singer 2003] returns 0 if the ranking is perfect and 1 otherwise, irrespective of how wrong the ranking is. This metric has the same meaning as 0/1 subset accuracy described in Zhu et al. [2005], but is applied to ranking:

$$is\ error = \frac{1}{t}\sum_{i=1}^{t}\left[\!\left[\sum_{\lambda\in\mathcal{L}}|\tau_i^*(\lambda) - \tau_i(\lambda)| \neq 0\right]\!\right]. \tag{14}$$

Ranking error [Park and Fürnkranz 2008] returns the normalized sum of squared position differences for each label in the predicted and true rankings. It is 0 for a ranking that is identical to the true ranking and 1 for a completely reversed ranking:

$$ranking\ error = \frac{1}{t}\sum_{i=1}^{t}\sum_{\lambda\in\mathcal{L}}|\tau_i^*(\lambda) - \tau_i(\lambda)|^2. \tag{15}$$

3.2. Partitioning Datasets

In supervised learning, two of the more frequent evaluation techniques are the *holdout* method, which splits the dataset into a training and a test set, and *cross-validation*, which is used when the training data are limited and splits the dataset into a number of disjoint subsets of the same size. In both techniques, partitions should have

approximately the same data distribution of the original distribution; this is known as *stratified partition*. In MLL, it is not clear how stratification should be carried out. Most works have used the default train/test partitions of the dataset or the random version of holdout and cross-validation methods, and literature about multilabel stratification is sparse. The work of Sechidis et al. [2011] can be referenced in which two stratification methods are proposed. The first one splits the partitions by considering the different combinations of labels. This approach is impractical for datasets in which the number of distinct labelsets is too large. The second proposal is a relaxed interpretation whose aim is to maintain the distribution of positive and negative examples of each label. Sechidis et al.'s experiments concluded that both approximations were better than random sampling. Finally, it is worth noting that some authors have performed train/test splits on large datasets (i.e., delicious and mediamill) since cross-validation becomes too computationally intensive [Read et al. 2011; Rokach et al. 2014].

3.3. Statistical Tests

To compare the performance of several multilabel classifiers, a two-step procedure for classical single-label classification recommended in Demšar [2006] has been frequently used in MLL. It consists of applying a Friedman test with the null hypothesis that all learners have equal performance, and, if the null-hypothesis is rejected, a post-hoc test is carried out. Two main post hoc tests have been applied: In Cheng and Hüllermeier [2009], Cherman et al. [2012], and Madjarov et al. [2012], a Nemenyi test that compared learners in a pairwise way, and, in Ávila et al. [2011], a Bonferroni-Dunn post hoc test. This test compares not in a pairwise way, but with a control algorithm (the one that obtains the lower ranking value in the Friedman test). According to Demšar [2006], if the target is only testing whether a newly proposed method is better than the existing ones, the power of the post hoc test is much greater when all classifiers are compared only to a control classifier and not in a pairwise way. For pairwise comparison of two classifiers, the nonparametric Wilcoxon signed-rank test (a better alternative to the paired t-test when the performance scores are not normally distributed) has been used in Vens et al. [2008] and Yang and Gopal [2012]. The null hypothesis is that both methods are equally effective; the alternative is that one of the methods is better.

3.4. Complexity

According to Tsoumakas et al. [2008], the high dimensionality of the label space may challenge the efficiency of MLL methods in two ways. On the one hand, the computational cost of training a multilabel classifier may be affected by the number of labels. There are simple algorithms (e.g., Binary Relevance) with linear complexity with respect to q, but there are also more advanced methods whose complexity is worse (e.g., Ranking by Pairwise Comparison). Second, the classification stage can also be influenced by the number of classifiers and can be quite time-consuming, especially in classification problems with large numbers of labels. Another important factor to consider related to high dimensionality is the memory requirements. All of these factors have to be taken into account when developing a new MLL, and they make the development of a time- and space-efficiency analysis necessary. In Zhang and Zhou [2014] detail the complexity of MLL methods.

4. MULTILABEL LEARNING METHODS

Today, most authors agree with the taxonomy presented in Tsoumakas et al. [2010a] that differentiates between two main approaches for solving MLL problems: *problem transformation methods* and *algorithm adaptation methods*. The former transforms the multilabel problem into one or more single-label problems that are solved with a

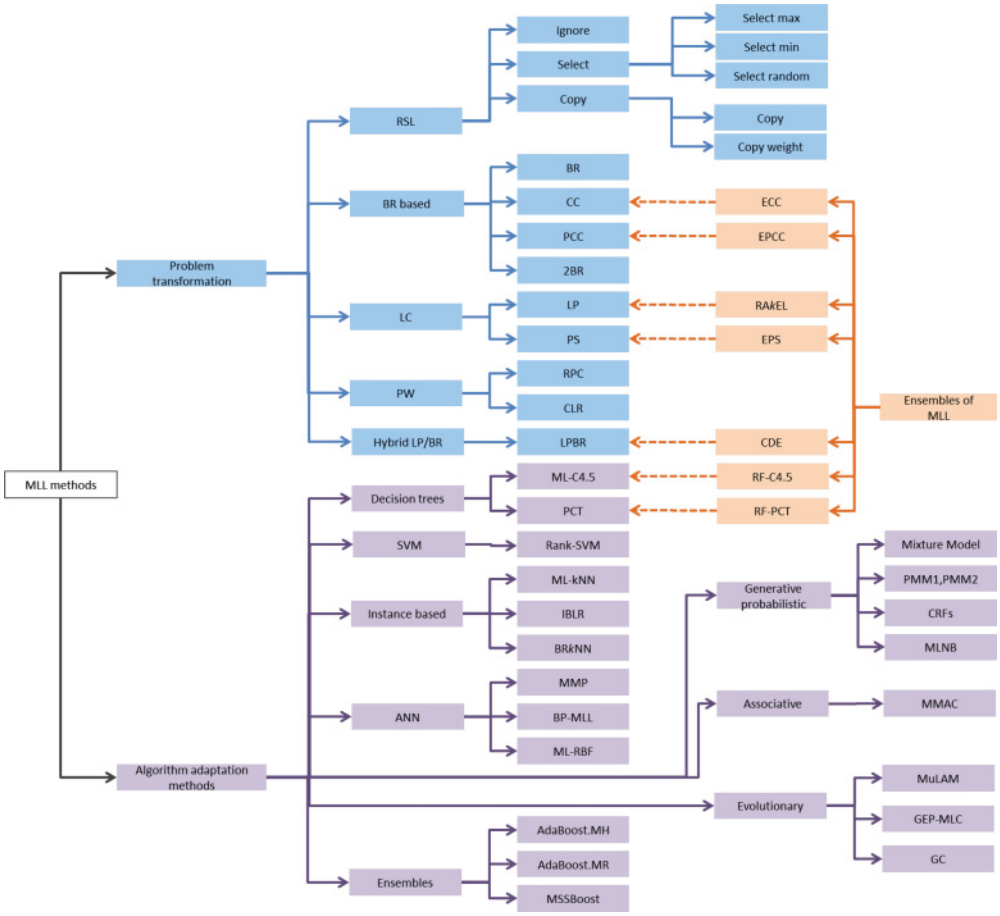

Fig. 3. Taxonomy of MLL methods.

Table II. *Ignore* Transformation of the Dataset in Table I

Example	$Y \subseteq \mathcal{L}$
2	λ_4

single-label classification algorithm, whereas the latter consists of extending a single-label algorithm in order to directly deal with multilabel data. It is worth noting that the problem transformation approach is algorithm-independent. A taxonomy with the proposals described in this article is found in Figure 3.

4.1. Problem Transformation Methods

4.1.1. Ranking Via Single-Label Learning Methods. The approach dubbed *ranking via single-label learning* (SLR) transforms a multilabel dataset into a single-label one and then uses a single-label classifier that is able to produce a score (e.g., probability) for each label in order to obtain a ranking. Thus, the label with the highest probability will be ranked first, and so on. A straight transformation is one called *ignore* (see Table II), which consists of ignoring all multilabel instances. It is a very simple idea, but useless due to its great drawback: the loss of information.

Table III. *Select* Transformation of the Dataset in Table I

(a) Select-max

Example	$Y \subseteq \mathcal{L}$
1	λ_2
2	λ_4
3	λ_2
4	λ_3
5	λ_2

(b) Select-min

Example	$Y \subseteq \mathcal{L}$
1	λ_1
2	λ_4
3	λ_4
4	λ_1
5	λ_3

(c) Select-random

Example	$Y \subseteq \mathcal{L}$
1	λ_1
2	λ_4
3	λ_3
4	λ_1
5	λ_2

Table IV. *Copy* and *Copy-Weight* Transformation of the Dataset in Table I

Example	$Y \subseteq \mathcal{L}$	Weight
1	λ_1	0.33
1	λ_2	0.33
1	λ_4	0.33
2	λ_4	1.00
3	λ_2	0.33
3	λ_3	0.33
3	λ_4	0.33
4	λ_1	0.50
4	λ_3	0.50
5	λ_2	0.50
5	λ_3	0.50

Table V. *BR* Transformation of the Dataset in Table I

Example	λ_1 vs. rest	Example	λ_2 vs. rest	Example	λ_3 vs. rest	Example	λ_4 vs. rest
1	true	1	true	1	false	1	true
2	false	2	false	2	false	2	true
3	false	3	true	3	true	3	true
4	true	4	false	4	true	4	false
5	false	5	true	5	true	5	false

Another simple transformation is selecting one of the labels of those instances with more than one label; this method is called *select,* and it also produces some information loss. Depending on the method used to select the label of the instance, it can be distinguished among the *select-max* (the most frequent label), *select-min* (the less frequent label), and *select-random* (a random selection) methods (see Table III).

The latest simple transformation method is called *copy* (see Table IV). It consists of transforming every multilabel instance into several instances, one per label. It is also possible to weight examples by $\frac{1}{|Y|}$, in which case it is called *copy-weight*. This last method does not produce information loss, but it increases the number of patterns and may complicate modeling decision boundaries [Read 2010].

4.1.2. Binary Relevance Methods. The *Binary Relevance* (BR) method generates one binary dataset for each label in which positive patterns are those predicting the label, and the rest are considered to be negative patterns (an example is shown in Table V). Once an unknown pattern is presented to the model, the output will be the set of positive classes predicted. This approach is similar to the *One-Versus-All* (OVA) approach employed to solve multiclass problems with binary classifiers, with the difference that in multiclass problems an instance has only one label associated. In Zhou et al. [2012a], three problems of BR are described. The first one is the fact that BR assumes labels are independent, so it ignores correlations and interdependences between labels and this is not always true. According to Read et al. [2011], due to this information loss, BR predictive performance can decrease; in Tsoumakas et al. [2009], it is highlighted that BR may fail to predict label combinations or rankings of labels. The second one is the problem of sample imbalance that may occur after the BR transformation. It

Table VI. *LP* Transformation of the Dataset in Table I

Example	$Y \subseteq \mathcal{L}$
1	$\lambda_{1,2,4}$
2	λ_4
3	$\lambda_{2,3,4}$
4	$\lambda_{1,3}$
5	$\lambda_{2,3}$

leads to induction of binary classifiers from datasets where negative examples tend to outnumber positive ones. The last problem is related to the high dimensionality of labels that may increase the sample imbalance and can also increase the number of classifiers to be trained. Despite these drawbacks, BR is simple and reversible (the original dataset can be recovered). In Read et al. [2011], the main advantages of BR are highlighted. First, it has low computational complexity compared with other methods, and BR scales linearly with the number of labels. Second, since labels are independent, they can be added and removed without affecting the rest of the model. This makes it applicable to an evolving or dynamic scenario and offers the opportunity of parallel implementation.

4.1.3. Label Powerset Methods. The *Label Powerset* (LP) approach, also called *Label Combination* (LC) in Read et al. [2011], generates a new class for each possible combination of labels and then solves the problem as a single-label multiclass one (an example can be seen in Table VI). When a new unknown instance is presented, LP outputs a class, which is actually a set of labels in the original dataset. This approach is effective and simple and is also able to model label correlations in the training data. Nevertheless, after the transformation, it is possible to have limited training examples for many new classes (the less frequent combinations), producing sample imbalance. In addition, this approach only takes into account the distinct labelsets in the training set, so it cannot predict unseen labelsets that may also lead to a tendency to overfit the training data [Read et al. [2011]. Finally, another problem is the potentially large number of classes that must be dealt with. This number is upper bounded by $min(m, 2^q)$; thus, in the worst-case scenario, the complexity is exponential with the number of labels. This is the reason why LP typically works well if the original labelset is small but quickly deteriorates for larger labelsets [Cheng and Hüllermeier 2009].

The *RAndom k-labELsets* method (RA*k*EL) [Tsoumakas and Vlahavas 2007; Tsoumakas et al. 2010b] constructs an ensemble of LP classifiers. Each classifier is trained with a random subset of k labels. Thus, RA*k*EL, as LP, is able to deal with label correlations but avoids the problems of LP related to the computational cost and class imbalance when the number of labels is high. In addition, it is able to predict unseen labelsets. During the classification, when an unknown instance is presented, the response of the classifiers is averaged per label; after that, a threshold is used to assign the labelset. It is worth highlighting that although RA*k*EL is independent of the multilabel algorithm, its authors recommend using methods heavily influenced by the specific set of labels such as LP or Pruned Sets (see below) while they recommend against BR and ML-kNN (this last one is heavily influenced by the feature space). Regarding the single-label base classifier, authors performed a study concluding that, in general, C4.5 and Support Vector Machines (SVMs) perform better than naive Bayes (NB). Experiments showed the improvement of RA*k*EL over BR and LP. Nevertheless, its build time increases by a factor of approximately 10 each time k is doubled [Read et al. 2008].

Pruned Problem Transformation (PPT) or *Pruned Sets* (PS) [Read et al. 2008] extend LP transformation while trying to avoid its problems related to complexity and unbalanced data by pruning examples with less frequent labelsets (under a user-defined

Table VII. *RPC* Transformation of the Dataset in Table I

Example	λ_1 *vs.* λ_2
3	false
4	true
5	false

Example	λ_1 *vs.* λ_3
1	true
3	false
5	false

Example	λ_1 *vs.* λ_4
2	false
3	false
4	true

Example	λ_2 *vs.* λ_3
1	true
4	false

Example	λ_2 *vs.* λ_4
2	false
5	true

Example	λ_3 *vs.* λ_4
1	false
2	false
4	true
5	true

Table VIII. Datasets Added to *RPC* to Obtain *CLR* Transformation of the Dataset in Table I

Example	$\lambda_1 vs. \lambda_0$
1	true
2	false
3	false
4	true
5	false

Example	$\lambda_2 vs. \lambda_0$
1	true
2	false
3	true
4	false
5	true

Example	$\lambda_3 vs. \lambda_0$
1	false
2	false
3	true
4	true
5	true

Example	$\lambda_4 vs. \lambda_0$
1	true
2	true
3	true
4	false
5	false

threshold). This reduces complexity by focusing on the most important combinations of labels. To compensate for such information loss, it reintroduces the pruned example, considering as output disjoint subsets of the pruned labelsets that do exist more times than the threshold. The *Ensemble of Pruned Sets* (EPS) algorithm [Read et al. 2008] constructs a number of PS by sampling the training sets (i.e., bootstrap). Given a new instance, the final response is obtained by a voting schema and a threshold that allow EPS to form new combinations of labels. The experiments showed that PS performed best in an ensemble scheme, whereas EPS outperformed LP and RAkEL and proved to be particularly competitive in terms of efficiency.

4.1.4. Pairwise Methods (PW). The *Ranking by Pairwise Comparison* (RPC) [Hüllermeier et al. 2008] approach transforms a dataset with q classes into $q(q-1)/2$ binary datasets, one per each pair of labels, and a binary classifier is built for each dataset. Each dataset, $\lambda_i vs. \lambda_j$, contains the patterns labeled with at least one of the two labels, but not both, being a true pattern if λ_i is true and false otherwise (an example is shown in Table VII). This approach is similar to the *One-Versus-One* (OVO) method for multiclass problems. Given a new instance, all models are invoked, and a ranking is obtained by the counting votes for each label. The main drawback is the space complexity and the need to query all the generated (q^2) binary models at runtime. According to Read et al. [2011], this quadratic complexity in terms of the number of labels makes RPC very sensitive to large q and usually intractable for large problems.

It is also worth mentioning *Calibrated Label Ranking* (CLR) [Brinker et al. 2006; Fürnkranz et al. 2008] that extends RPC by means of an additional virtual or calibration label, λ_0. Thus, the final ranking will include the virtual label that can be interpreted as a split point for relevant and nonrelevant labels obtaining a consistent ranking and bipartition. The transformation is built by adding to the RPC transformation (in Table VII) a new dataset for each label, λ_i, corresponding to the pair $\lambda_i vs. \lambda_0$. Each new dataset uses all the examples; so, when the label λ_i is true, the virtual label is considered false and vice versa (as in BR transformation). An example of the datasets added to the RPC transformation to consider the calibration label is shown in Table VIII. Experiments carried out in the fields of text categorization and gene analysis concluded that CLR outperformed the BR approach [Fürnkranz et al. 2008]. Nevertheless, the space complexity of the model is similar to RPC but needs to query q^2+q binary models. Alternatives to decrease the complexity of the voting process have been proposed in Madjarov et al. [2011] and Loza et al. [2009].

4.1.5. Transformations for Identifying Label Dependences. The *Classifier Chains* (CC) model [Read et al. [2011] generates q binary classifiers, but they are linked in such a way that the feature space of each link in the chain is extended with the label associations of all

Table IX. *CC* Transformation of the Dataset in Table I. Chain $\lambda_1, \lambda_2, \lambda_3, \lambda_4$

Example	λ_1 *vs.rest*	Example$\cup\lambda_1$	λ_2 *vs.rest*	Example$\cup\lambda_1 \cup \lambda_2$	λ_3 *vs.rest*	Example$\cup\lambda_1 \cup \lambda_2 \cup \lambda_3$	λ_4 *vs.rest*
1	true	1 true	true	1 true true	false	1 true true false	true
2	false	2 false	false	2 false false	false	2 false false false	true
3	false	3 false	true	3 false true	true	3 false true true	true
4	true	4 true	false	4 true false	true	4 true false true	false
5	false	5 false	true	5 false true	true	5 false true true	false

Table X. Frequency Counts of Labels λ_i and λ_j

	λ_j	$\neg\lambda_j$	total
λ_i	a	b	a + b
$\neg\lambda_i$	c	d	c + d
total	a + c	b + d	a + b + c + d

previous links (see Table IX). Thus, CC overcomes the label independence assumption of BR and also overcomes the worst-case computational complexity of LP (exponential with the number of labels). On the one hand, when labels are independent, CC will tend to function similarly to BR, whereas, on the other hand, given the presence of label correlations—despite not being optimal—it will tend to function like LP. Because the order of the chain itself can influence the performance, its authors proposed using an *Ensemble of Classifier Chains* (ECC) that trains a set of CC classifiers with a random chain ordering and a random subset of training patterns. The sum of votes for each label is computed and normalized, with the output of the classifier being those labels that exceed a threshold. A Bayes optimal way of forming classifier chains based in probability theory, dubbed *Probabilistic Classifier Chains* (PCC), was described in Dembczyński et al. [2010]. It tests all possible chain orderings and predicts $\arg\max_{Y\subseteq\mathcal{L}} P(Y|\mathbf{x})$. Despite obtaining better accuracy than CC, because PCC has to look at each of 2^q possible combinations at the prediction stage, the applicability of the algorithm is only advisable for datasets with a small to moderate number of labels ($q \leq 15$). *Ensemble of Probabilistic Classifier Chains* (EPCC) is the application of the method used to build the ensemble ECC to PCC.

To overcome the problems of the independence assumption of the BR model, some authors have proposed *2BR*, also called *MetaBR* (MBR) [Read et al. [2011], which basically consists of applying BR twice. During the first step, a BR classifier is learned; the second BR step implements a meta-learning stage. There will be a binary meta-learner for each label to be learned. The input will be the output of all the BR classifiers in the first step plus the desired output. It follows the philosophy of *stacking* proposed by Wolpert [1992] and maintains the linear time complexity with respect to the number of labels in the dataset. In Tsoumakas et al. [2009], during meta-learning, predictions of the base-level models only participated on those labels whose absolute value of the ϕ coefficient was greater or equal to a certain threshold $t, 0 \leq t \leq 1$. Given two labels, λ_i and λ_j and the frequency counts of their co-occurrences (see Table X), the ϕ coefficient is obtained by Equation (16). Experiments showed that the pruning substantially improved computational cost while maintaining or improving predictive performance. Other stacking approaches are described in Godbole and Sarawagi [2004], Pachet and Roy [2009], and Antenreiter et al. [2009]:

$$\phi = \frac{ad - bc}{\sqrt{(a+b)(c+d)(a+c)(b+d).}} \tag{16}$$

In Tenenboim et al. [2010], LPBR is proposed to try to find the optimal tradeoff between the simplicity of BR and the complexity of LP and to find a balance between theindependence assumption of BR and the few examples for many labels of LP

datasets. It manages this target by combinations of rounds of LP (within the groups of dependent labels) and BR (applied to the independent labels). The approach where the dependence between labels is computed by using the χ^2 score (see Equation (17)) is called *ChiDep*. First, all label pairs are sorted according to the χ^2 score, and a first round with BR is applied. In each round, the most dependent label pair is considered. Depending on whether the labels of the pair belong to other previous groups of labels, either a new group with these two labels can be formed, they can be added to a previous group, or two previous groups can be joined to include this pair of labels. Then, LP will be applied to the groups of dependent labels (groups with a limited, potentially small number of labels) and BR to the independent labels:

$$\chi^2 = \frac{(ad - bc)^2(a + b + c + d)}{(a + b)(c + d)(b + d)(a + c)}. \tag{17}$$

To improve performance, a method to develop a *ChiDep Ensemble* (CDE) is proposed. A large number of random labelset partitions are generated (i.e., 10,000), and a score based on the χ^2 score of all label pairs in each partition is computed. The top high-scoring partitions are selected as members of the ensemble. Experiments showed a predictive performance higher than BR and LP; nevertheless, both ChiDep's and CDE's train times were relatively long and approximate to that of RA*k*EL and 2BR.

4.2. Algorithm Adaptation Methods

Almost all classical paradigms in classical or single-label classification have been revisited in order to be adapted to multilabel data.

4.2.1. Decision Trees. Decision tree methods have been mainly used in the field of genomics due to the interpretability of its outputs and in hierarchical [Blockeel et al. 2006; Vens et al. 2008] and ensemble settings [Madjarov et al. 2012]. It is worth citing ML-C4.5, the adaptation of the popular C4.5 [Quinlan 1993] carried out by Clare and King [2001]. Multiple labels in the tree's leaves were allowed, and the entropy definition was adapted to take into account how much information was needed to describe to what classes a certain pattern belonged (see Equation (18)). So, given q classes, not only the probability of membership $p(\lambda)$ is considered, but also the probability of not membership, $1 - p(\lambda)$. These probabilities are measured in terms of relative frequency. When the algorithm checks whether it is better to prune a branch and replace it with a leaf, the most frequent set of classes in the branch is found (rather than the best single class) and how many items are in this set of classes is determined:

$$entropy_{ML}(S) = \sum_{i=1}^{q} P(\lambda_i)\log(P(\lambda_i)) + (1 - P(\lambda_i))\log(1 - P(\lambda_i)). \tag{18}$$

Predictive Clustering Trees (PCT) [Blockeel et al. 1998] consider a decision tree as a hierarchy of clusters in which data are partitioned in a top-down strategy by minimizing the variance. The leaves represent the clusters and are labeled with the cluster's prototype (prediction). Unlike standard decision trees, the variance and the prototype functions are treated as parameters. Particularly in MLL, the variance function is computed as the sum of the Gini indices [Breiman et al. 1984] of the variables from the target tuple, and the prototype function returns a vector with probabilities for each label [Madjarov et al. 2012]. The approach obtained competitive performance as base classifier on random forest ensembles (see Section 5). PCTs have also been used in hierarchical multilabel learning [Vens et al. 2008] (see Section 7) .

4.2.2. Support Vector Machines. Single-label Support Vector Machines (SVMs) have been widely used in MLL by applying an OVA approach [Gonçalves and Quaresma 2004;

Boutell et al. 2004]. The algorithm adaptation approach has also been used. In Elisseeff and Weston [2001], the authors proposed an SVM ranking-based algorithm called *Rank-SVM* that improved performance over BR with SVMs. A set of q linear classifiers, $\{h_j(\mathbf{x}) = \langle w_j, \mathbf{x}\rangle + b_j = w_j^T \cdot \mathbf{x} + b_j | 1 \le j \le q\}$, each with weight vector, w_j, and bias, b_j, are defined. They are optimized to minimize the empirical ranking loss with quadratic programming in its dual form and kernel trick to manage nonlinearity. The multilabel margin defined on the whole training set (Equation (19)) considers its capability to properly rank every relevant-irrelevant label pair for each training example, $(\mathbf{x}_i, Y_i)$, in the training set, S. The boundary for each pair of relevant-irrelevant labels corresponds to the hyperplane $\langle w_j - w_k, \mathbf{x}_i\rangle + b_j - b_k$. Improvements to this method have been presented in Jiang et al. [2008] and Xu [2012]:

$$\min_{(\mathbf{x}_i, Y_i) \in S} \min_{(y_j, y_k) \in Y_i \times \overline{Y_i}} \frac{\langle w_j - w_k, \mathbf{x}_i\rangle + b_j - b_k}{\| w_j - w_k \|}. \tag{19}$$

4.2.3. Instance-based Algorithms. As far as we know, the first multilabel lazy learning algorithm is *multilabel k-nearest neighbor* (ML-kNN) proposed by Zhang and Zhou [2005]. Given an unknown instance, $\mathbf{x}$, the algorithm first determines $N = \{(\mathbf{x}_i, Y_i) | 1 \le i \le k\}$, the set of k nearest neighbors and obtains a membership counting vector, $\mathbf{c} = (c_1, \ldots c_q) | c_j = \sum_{(\mathbf{x}_i, Y_i) \in N} [\![\lambda_j \in Y_i]\!]$, that stores, for each label, the number of examples in the neighborhood of $\mathbf{x}$. Then, based on prior and posterior probabilities for the frequency of each label within the k nearest neighbors, it identifies the set of labels to be associated with the unseen instance by using the *Maximum A Posteriori* (MAP) principle:

$$y_j = \begin{cases} 1 & if\ P(c_j | y_j = 1) P(y_j = 1) \ge P(c_j | y_j = 0) P(y_j = 0). \\ 0 & otherwise \end{cases}$$

Cheng and Hüllermeier [2009] proposed *Instance-Based Learning by Logistic Regression* (IBLR), an approach that combines Instance-Based Learning (IBL) and logistic regression. The key idea is to consider labels of neighboring instances as features of unseen samples and to reduce IBL to logistic regression. This approach is able to capture interdependencies between labels; these interdependencies are reflected by the sign and magnitude of the regression coefficients, thus improving upon ML-kNN. Experiments showed IBLR outperformed the predictive accuracy of LP, MLkNN, and BR with kNN as base classifier. Finally, in Spyromitros et al. [2008], BRkNN was described as equivalent to using BR with kNN as the base classifier, but much faster because, instead of computing q times the k nearest neighbors, it searches the k nearest neighbors only once.

4.2.4. Neural Networks. Crammer and Singer [2003] proposed the *Multilabel Multiclass Perceptron* (MMP) algorithm. Just as in BR, one perceptron is used for each label, and the prediction is calculated via the inner products. Nevertheless, instead of learning the relevance of each class independently, MMP is incrementally trained to produce a real-valued relevance score that ranks relevant labels above the irrelevant ones. So, the performance of the whole ensemble is considered to update each individual perceptron. Studies have demonstrated it is efficient, competitive, and suitable for solving large-scale multilabel problems [Loza and Fürnkranz 2007].

Later, Zhang and Zhou [2006] developed *Backpropagation for Multilabel Learning* (BP-MLL), an adaptation of the traditional multilayer feed-forward neural network to multilabel data. The net was trained with gradient descendent and error backpropagation with an error function closely related to the ranking loss that took into

account the multilabel data (see Equation (20)):

$$E = \sum_{i=1}^{m} \frac{1}{|Y_i||\overline{Y_i}|} \sum_{(j,k)\in Y_i \times \overline{Y_i}} exp\big(-\big(o_j^i - o_k^i\big)\big). \tag{20}$$

where $o_j^i - o_k^i$ measures the difference between the outputs of the network on one label belonging to the i-th pattern and one label not belonging to it. The network architecture has three layers. The input layer consists of d units, each one corresponding to one dimension in the input space. The output layer has q units whose output will be used for label ranking (i.e., the labels belonging to an instance should be ranked higher than those not belonging to it). The hidden layer is fully connected with the input and output layers using weights. Experimental results showed competitive performance in genomics and text categorization domains with a computational cost derived according to neural networks methods.

Finally, Zhang [2009] presented *Multilabel Radial Basis Function* (ML-RBF), an approach inspired by the well-known RBF method. The input corresponds to a d-dimensional feature vector. It consists of two layers of neurons: In the first layer, each hidden neuron (basis function) is associated with a prototype vector, whereas each output neuron corresponds to a possible label. The network is trained by means of a two-stage procedure. First, basis functions in the hidden layer are learned by performing k-means clustering on instances of each possible class (other clustering algorithms could also be used). So, the centroids of the clustered groups will constitute the proto- type vectors of the first-layer basis functions (q sets of prototype vectors). After that, the weights of the second layer are optimized through minimizing a sum-of-squares error function. It is worth noting that each output neuron is connected with all basis functions corresponding to the prototype vectors of all possible classes. Therefore, the correlations between labels are addressed both in training and test.

4.2.5. Generative and Probabilistic Models. Many of the approaches for multilabel document classification mainly rely on discriminative modeling techniques; nevertheless, some generative models have also been devised. In McCallum [1999], a probabilistic generative model for text classification was presented. It assumes that associated with each individual label is a word distribution $P(w|\lambda)$ for all words in the vocabulary $\mathcal{V} = \{w_1, \ldots, w_d\}$. So, a document is generated by a mixture of these word distributions with mixture weights $\gamma^Y = (\gamma^Y_{\lambda_1}, \ldots, \gamma^Y_{\lambda_q})$, with $Y \subseteq \mathcal{L}$. Given a document $\mathbf{x}$, it is associated with a labelset Z, according to Equation (21).

$$Z = \arg\max_{Y\subseteq\mathcal{L}} P(Y) \prod_{w\in\mathbf{x}} \sum_{\lambda\in Y} \gamma^Y_\lambda P(w|\lambda), \tag{21}$$

where $P(Y)$ is directly estimated from the training set by frequency counting and $P(w|\lambda)$ and γ^Y_λ, the mixture weight of label λ in mixture weight distribution γ^Y, are estimated with Expectation Maximization (EM) [Dempster et al. 1977]. Later, Ueda and Saito [2002a] presented PMM1 and PMM2, two probabilistic generative *Parametric Mixture Models*. The basic assumption under PMMs is that multilabeled text has a mixture of characteristic words appearing in single-labeled text that belong to each category of the multicategories. Because the described generative models are based on text frequencies in documents, they are specific for text domains. The use of *Conditional Random Fields* (CRFs) [Lafferty et al. 2001] has been proposed in Ghamrawi and McCallum [2005] with two multilabel graphical models for classification that parameterize label co-occurrences, and Shotton et al. [2009] also used CRFs to incorporate different low-level image features. Finally, in Zhang et al. [2009], a method called *Multilabel Naive Bayes* (MLNB) was presented. It adapted the NB classifier to deal with multilabel instances.

Using the Bayesian rule and adopting the assumption of class conditional independence among features (as classic naive Bayes), given a test instance $\mathbf{x}$, the MAP estimate is computed as in the following equation. The density of the features variables conditioned on the class values follows a Gaussian distribution $g(x_k, \mu_k^{jb}, \sigma_k^{jb})$, $1 \leq k \leq d$:

$$y_j = \begin{cases} 1 & if\ P(y_j = 1)\exp(\phi_1) \geq P(y_j = 0)\exp(\phi_0) \\ 0 & otherwise \end{cases} \quad where\ \phi_b = -\sum_{k=1}^{d} \frac{(x_k - \mu_k^{jb})^2}{2\sigma_k^{jb2}} - \sum_{k=1}^{d} \ln \sigma_k^b$$

4.2.6. Associative Classification. Associative classification integrates association rule mining and classification. *Multiclass, multilabel associative classification* (MMAC) [Thabtah et al. 2004] first scans the training data to discover and generate an initial set of classification rules by association rule mining. Next, an iterative process learns rules from the remaining unclassified instances until no further frequent items are left. The generated rules have only one consequent (one label). Finally, the rule sets derived are merged to form a multilabel classifier. Rules with the same antecedent but different consequent are merged by ranking labels according to their frequency in the training patterns satisfying the antecedent. Other associative approaches are found in Thabtah and Cowling [2007] and Rak et al. [2008].

4.2.7. Evolutionary Approaches. Bio-inspired approaches have also been used to solve multilabel problems. To the best of our knowledge, the first one, called *Multilabel Ant-Miner* (MuLAM), was proposed by Chan and Freitas [2006]. It was an extension of the ant colony-based Ant-Miner algorithm [Parpinelli et al. 2002]. The rule representation allowed more than one predicted class in the rule consequent, and each ant was able to discover a set of rules: at least one rule and at most a rule for each class. In Ávila et al. [2011], GEP-MLC, an evolutionary approach to find discriminant functions was proposed. Later, the same authors proposed GC [Ávila et al. 2010], another evolutionary approach to build classification rules using a model more interpretable than discriminant functions. Both proposals obtained results competitive with the state-of-the-art in MLL.

4.2.8. Ensembles. Schapire and Singer [1999, 2000] proposed a set of boosting algorithms for text categorization adapted to the multilabel case and based in the popular AdaBoost [Freund and Schapire 1997], namely *AdaBoost.MH* and *AdaBoost.MR*. The aim of AdaBoost.MH is to minimize Hamming loss and maintain a set of weights not only over the training examples, but also over labels. During training, weights related to labels and difficult to classify examples are increased. In practice, this algorithm carries out a reduction of the multilabel problem by mapping each example $(\mathbf{x_i}, Y_i)$ to q binary examples $\{(\mathbf{x_i}, \lambda), Y_i(\lambda)\}$ for all $\lambda \in \mathcal{L}$. Here, $(\mathbf{x_i}, \lambda)$ denotes the concatenation of $\mathbf{x_i}$ and label λ, and $Y_i(\lambda) = +1$ if $\lambda \in Y_i$ and -1 otherwise. Then, binary AdaBoost is applied to the transformed binary-labeled examples iteratively with one-level decision trees as base learners. The output of each weak learner is a hypothesis $h : \mathcal{X} \times \mathcal{L} \to \mathbb{R}$, with the sign of the output being interpreted as a prediction on the relevance of the label and the magnitude being interpreted as a measure on the confidence of the prediction. Given an ensemble of T base classifiers, the final output is $f(\mathbf{x}, \lambda) = \sum_{i=1}^{T} \alpha_t h_t(\mathbf{x}, \lambda)$. A discussion on the values of α_t can be found in Schapire and Singer [2000]. AdaBoost.MR operates in a similar way. However, because its aim is to minimize the ranking loss, it has to take into account all labels misorderings; thus, the set of weights is maintained for each instance and pair of labels. Some works based on these popular algorithms can be found in Sebastiani et al. [2000], Nardiello et al. [2003], and De Comité et al. [2003].

Several ensemble-based approaches that work at feature and data level have been presented. An example is a boosting-type algorithm called *Model-Shared Subspace Boosting* (MSSBoost) [Yan et al. 2007], in which each model was learned from random feature subspace and bootstrap data sampling. It exploited the label space redundancy by allowing base models to be shared across multiple labels.

4.3. Ensembles of MLL Methods

Madjarov et al. [2012] consider that ensemble methods whose base classifiers are multilabel as a special group that is different from both problem transformation and algorithm adaptation because they are developed on top of these two approaches. Therefore, RA*k*EL, EPS, ECC, EPCC, and CDE (described in Section 4.1) are ensembles of MLL methods. Other approaches worth citing are *Random Forest of Predictive Clustering Tress* (RF-PCT) [Kocev et al. 2007; Kocev 2012] and *Random Forest of ML-C4.5* (RF-C4.5) [Madjarov et al. 2012], two ensemble methods that used PCT and ML-C4.5 trees (described in Section 4.2.1) as base classifiers, respectively. The diversity of base classifiers was obtained by using bagging and changing the feature set during learning (i.e., a random subset of features was considered to select the best split attribute). Ensembles of MLL methods are different from ensembles described in Section 4.2.8 because, one way or another, they decompose the problem into a set of binary single-label methods whereas ensembles of MLL methods have as base classifier a multilabel learner.

4.4. Thresholding Strategies

Many of the described algorithms output a score (e.g., probability, ensemble votes, etc.) or a ranking, but obtaining a bipartition is still needed. As mentioned in Section 2.2, a bipartition of labels can be obtained from a ranking by means of a threshold.

The simplest approach is to use a predefined threshold, t; given a new instance, a label will be considered relevant if its score is greater than t. This value can be user-defined (i.e., 0.5) or previously tuned by a validation set. This approach is called *One Threshold* (OT) in Ioannou et al. [2010]. Another simple approach is *RCut* [Yang 2001], a *ranking-based* strategy that assigns the t top-ranked categories, where $1 \leq t \leq q$ can be either specified by the user (a common value is the label cardinality of the dataset [Tang et al. 2009]) or tuned by means of a validation set. Note that when $t = 1$, the output is single-label. According to Montejo-Ráez and Ureña López [2006], it is not a good approach because classes that were refused by the binary classifier can be selected; and, equally, some classes that were found positive by the classifier may be discarded. Yang [2001] also noted that it tends to over fit and to be unstable across datasets. More thresholding approaches can be found in Tang et al. [2009].

5. EXPERIMENTAL COMPARISONS OF MLL METHODS

Despite the number of proposals for MLL, the development of exhaustive experimental comparative studies to get a better understanding of different MLL algorithms is still an open topic. It is worth citing an extensive comparison with significance statistical test involving 12 MLL methods, 16 evaluation measures, and 11 benchmark datasets with different scales and from different domains (i.e., biology, multimedia, and text) [Madjarov et al. 2012]. The best overall methods were RF-PCT and the Hierarchy of Multilable Classifiers (HOMER; see Section 7), followed by BR and CC. Regarding the base classifiers, SVMs and decision trees (i.e., J48 and PCT) were used. Because SVMs are able to exploit the information of all the features, they performed better on datasets with a large number of features but small number of examples, whereas tree-based methods performed better with larger datasets. ML-kNN performed poorly across all evaluation measures.

In Chekina et al. [2011], a meta-learning approach based on datasets' metafeatures was used to recommend the best MLL algorithm to be used over a certain domain.[1] The study involved 11 algorithms, 12 datasets, and 18 measures. HOMER, BR, ECC, and EPS obtained the best predictive performance results. In addition, certain metafeatures were found to be relevant for recommending an algorithm (e.g., number of labels, label cardinality of the training set, average examples per class, number of unconditionally dependent label pairs, etc.). The results of those studies shed some light on which algorithm to select or which algorithms must be taken into account when developing a new one. The final decision will depend on the problem being faced and the requirements that must be satisfied: efficiency, flexibility, predictive performance, interpretability of the model, etc.

6. APPLICATIONS OF MLL

Although the term multilabel was not yet used, the first works in MLL were related to text categorization [Yang 1999; McCallum 1999; Schapire and Singer 2000]. Later, many other applications, mainly those related to bioinformatics and classification of multimedia, arose. Recently, MLL has been applied to an increasing number of new applications. A summary of the main fields of applications of MLL is described here.

6.1. Text Categorization

The problem of text categorization basically consists of assigning a set of predefined categories to documents in order to make certain tasks faster and cheaper. Since a document can belong simultaneously to more than one category, it can be considered a multilabel problem. Document classification has been applied to many domains. For example, in Loza and Fürnkranz [2008] and Loza and Fürnkranz [2010], the authors studied the problem of assigning documents of the EUR-Lex database of legal documents (treaties, legislation, case-law, legislative proposals, etc.) of the European Union to a few of 4,000 possible labels. Due to the fact that the number of web documents is increasing daily, another application domain is the automatic categorization of web documents [Rubin et al. 2012; Ueda and Saito 2002a]. Automatic document categorization has also been applied in the fields of news [Schapire and Singer 2000], research papers [Nguyen et al. 2005], and economics [Vogrincic and Bosnic 2011]. Other fields of interest where the manual categorization of documents to facilitate information retrieval needs a lot of time and electronic and human resources include:

—*Document indexing:* The task of assigning a document a set of keywords from a controlled vocabulary (thesaurus or ontology) in order to describe its content. In Lauser and Hotho [2003], MLL was applied to an extensive document base maintained by the Food and Agriculture Organization (FAO) of the United Nations (UN).
—*Tag suggestion:* The automated process of suggesting useful and informative tags or keywords to an emerging object based on historical information. Examples are found in Song et al. [2011] and Katakis et al. [2008].
—*Medical coding:* The process of transforming information contained in patient medical records into standard predefined codes (an example is the ICD-9-CM). Because each document can be assigned to one or more codes, the problem can be considered from an MLL perspective [Yan et al. 2010].
—*IR from narrative clinical text*: In Spat et al. [2008], MLL was used to classify clinical texts into a set of categories referring to medical fields (*surgery*, *radiology*, etc.).

[1]Only one algorithm was recommended, but meta-learning for recommendation of algorithms may also be a multilabel problem.

—*Economic activities classification* [Ciarelli et al. 2009]: The task of finding a correspondence between a contract that contains a description of the business activities of one company and a set of standard categories.
—*Patent classification*: Because a patent document may be associated with several categories, the problem can be considered multilabel [Cong and Tong 2008].
—*E-mail filtering*: Yearwood et al. [2010] applied MLL to obtain phishing profiles from e-mails where profiles were generated based on the predictions of the classifier.
—*Classifying news sentences into multiple emotion categories* may be used to design intelligent interfaces. In Bhowmick et al. [2010], a set of news sentences were categorized according to the emotions triggered in readers (*disgust*, *happyness*, etc.).
—*Aeronautics reports*: The Aviation Safety Reporting System database (ASRS) to detect anomalies contains reports submitted by flight crews regarding events that took place during a flight. Because a report may belong to multiple classes, in Oza et al. [2009], the problem has been solved using MLL.
—*Query categorization:* In applications such as Internet portals and search engines, it is useful to categorize user search queries into a set of relevant classes to deliver to users content and ads that are relevant to their interests [Tang et al. 2009].

6.2. Multimedia

With the exponential growth of digital multimedia resources (e.g., images, videos, sounds) whose manual annotation requires great effort, MLL has become a powerful tool.

—*Automatic image and video annotation*. An image can have several tags simultaneously associated with it, making this problem a multilabel one [Wang et al. 2008; Tahir et al. 2009]. This application is also called *semantic scene classification*. Video annotation consists of assigning several semantic concepts to a video [Dimou et al. 2009; Wang et al. 2010].
—*Face verification* consists of determining whether different images correspond to the same person. To tackle this target, in Kumar et al. [2009] a set of MLL classifiers return the presence of certain visual features or traits in the picture.
—*Object recognition* is the automatic detection, recognition, and segmentation of object classes in pictures. Therefore, given a picture, the system is able to automatically find semantic regions, each labeled with an object class [Shotton et al. 2009].
—*Detection of emotions in music* is an MLL problem in which songs are classified simultaneously in several categories [Trohidis et al. 2008; Ma et al. 2009]. It has applications such as music recommendation systems or music therapy.
—*Speech emotion classification*. This problem consists of inferring affective states (e.g., emotions, mental states, attitudes, etc.) from nonverbal expressions in speech. These affective states can occur simultaneously. It has applications in fields such as human-computer and human-robot interfaces and public speaking skills assessment. It has been solved with MLL in Sobol-Shikler and Robinson [2010].
—*Music metadata extraction* is an MLL problem consisting of automatically extracting perceptive information such as genre, mood, or main instruments from acoustic signals [Pachet and Roy 2009].

6.3. Biology

In the field of biology, MLL has been successfully applied to the following problems:

—*Gene function prediction* consists of assigning functions for unknown genes; each gene may be associated with not one but a set of functional classes. This problem has been tackled with MLL in Elisseeff and Weston [2001], Barutcuoglu et al. [2006], Skabar et al. [2006], and Zhang and Zhou [2006].

—*Protein function prediction*. Because proteins often have multiple functions, MLL has been applied in Diplaris et al. [2005] and [Chan and Freitas 2006].
—*Protein subcellular multilocation*. The localizations of a protein in a cell are important functional attributes. Because proteins may simultaneously exist at, or move between, two or more different subcellular locations, MLL has been applied in Yang and Lu [2006] and Chou et al. [2011].
—*Predicting proteins 3D structures*. Proteins fold into a specific 3D structure that determines their functions in the cell. In Duwairi and Kassawneh [2008], a multilabel classifier is developed where there are two or more class labels to be predicted, and a hierarchical classifier is able to predict the structural classes and folds of proteins simultaneously, as in the natural hierarchy of proteins itself.

6.4. Chemical Analysis

The main applications of MLL in the field of chemical data analysis are detailed next.

—*Drug discovery*. In Kawai and Takahashi [2009], MLL has been used to identify drugs that have two or more different biological actions.
—*Vision-Based Metal Spectral Analysis*. A spectrum could contain emissions from multiple elements. Thus, in Ukwatta and Samarabandu [2009], spectral images are processed in real-time in order to detect contaminants in machine lubricants.
—*Adverse Drug Reactions* (ADRs). MLL has been applied to predict reactions given a set of drugs, and to identify the most likely drugs responsible for given reactions [Mammadov et al. 2007].

6.5. Social Network Mining

The application of MLL to classification problems in social networks has become a new area of interest. The following applications can be highlighted:

—*Collective behavior learning*. This consists of inferring the behavior or preferences of unobserved individuals. Here, behavior can include actions as joining a group, connecting with someone, clicking on an ad, becoming interested in certain topics, etc. This problem is dealt with in Tang and Liu [2009] using MLL.
—*Social networking advertising*. In Krohn-Grimberghe et al. [2012], MLL is used for predicting a list of ranked items in order to make recommendations.
—*Automatic annotation*. In Peters et al. [2010], the task of assigning labels for images when users and images are connected through multiple relations (e.g., authorship, friendship, etc.) is addressed.

6.6. Other Applications

Finally, other fields of applications where MLL has been applied are enumerated.

—*Tagging of Learning Objects* (LO). A learning object can be defined as a minimal content unit that intends to teach something and can be reused on different platforms. Since an LO can be multiply tagged, MLL has been used in López et al. [2012].
—*Direct Marketing*. As opposed to mass marketing, which advertises indiscriminately, direct marketing is a process that identifies potential buyers of a certain product in order to make them the target of promotions. Because one customer can be the target of several products, MLL has been applied in Zhang et al. [2006].
—*Medical Diagnosis*. In clinical data, a case has many symptoms, and they may be associated with more than one syndrome; hence, medical diagnosis can be solved by MLL techniques [Shao et al. 2010]. In addition, in Huang et al. [2008], MLL was proposed to simultaneously segment several significant tissue regions in digitized uterine cervix images. Finally, in Abbas et al. [2013], MLL was applied to the problem

of classifying dermoscopy images of skin lesions (skin lesions often contain several pattern lesions).

7. TRENDING CHALLENGES

As has been shown, MLL is a trending learning paradigm, and new unsolved issues continue to arise:

—*Dimensionality Reduction*. One common feature of multilabel data is the high number of attributes and labels that may influence the efficiency and effectiveness of the algorithms. The *dimensionality reduction of the input space* tries to reduce the number of attributes under two distinct points of view: feature selection (removing irrelevant or redundant features) and feature extraction (compressing dependent variables into a smaller number of predictors). The former has been used in Yang and Pedersen [1997], Trohidis et al. [2008], Chen et al. [2007], and Zhang et al. [2009], whereas the latter has been used in Yu et al. [2005] and Zhang and Zhou [2010]. On the other hand, strategies for the *reduction of the label space* have also been applied. For example, HOMER [Tsoumakas et al. 2008], which generates a tree of multilabel classifiers and whose complexity depends on q, has been used. Other proposals are *Compressed Sensing* (CS) [Hsu et al. 2009] and *Principal Label Space Transformation* (PLST) [Tai and Lin 2010].

—*Label Dependence*. In MLL problems, when the number of labels is high or even moderate, complexity may become exponential due to the possible combinations of labels. To cope with this issue, correlations between labels could be explored. This issue has been tackled in Dembczyński et al. [2012], where two types of label dependency in multilabeled data were identified: conditional (dependent on a particular instance) and unconditional (independent of a certain instance). Particular approaches are found in Qi et al. [2007], Zhu et al. [2005], Ji et al. [2010], and Zhang and Zhang [2010].

—*Active learning*. There are many applications where data is unlabeled, or labeling data is expensive or impractical. The aim of *Active Learning* (AL) is to iteratively rank unlabeled examples in terms of how useful they would be in order to propose only the top-ranked ones to human annotators. In the framework of MLL, AL becomes complex, mainly in text and image classification domains, because of the need to assign several labels. A straightforward way is to apply BR to solve q binary problems independently [Brinker 2006]. Other approaches take into account label relationships to reduce human effort during the labeling process [Qi et al. 2009]. In Esuli and Sebastiani [2009], a unique ranking of examples that combines the outputs of the individual binary classifiers is proposed.

—*Multi-instance multilabel learning* (MIML). *Multi-Instance Learning* (MIL) is a supervised learning setting where the task is to find a function $h_{MIL} : 2^{\mathcal{X}} \rightarrow \{0, 1\}$ from patterns (X, y) consisting of a set of vectors $X \subseteq \mathcal{X}$ (a bag of instances). Each bag (pattern) is labeled as positive or negative, $y \in \{0, 1\}$. While multilabel concerns the ambiguity in the output space, multi-instance concerns the ambiguity of the input space. In *Multi-Instance Multilabel Learning* (MIML), a training example is described by multiple instances and associates not one, but a set of labels. Formally, the task of MIML is learning a function $h_{MIML} : 2^{\mathcal{X}} \rightarrow 2^{\mathcal{L}}$ from patterns (X, Y) where $X \subseteq \mathcal{X}$ and $Y \subseteq \mathcal{L}$. Some proposals can be found in Zhou and Zhang [2006] and Zhang and Zhou [2007]. Because in many MLL problems, labels can be related only to different parts of the object; thus, developing learners on a MIML setting may reduce noise and improve performance [Zhou et al. 2012b]. He et al. [2012] identified two crucial issues for dealing with MLL problems by using a MIML setting. The first one is modeling the connections between instances and labels of a sample, and the

second one is exploiting relationships between labels. These two problems are not usually tackled together because the models may become complex and difficult to solve.

—*Multiview learning*. The MI, MLL, and MIML settings only deal with situations where data come from a single feature set (single-view). However, there are real-world applications in which one object has different representations in the form of multiple views (multiview). Multiview and MLL have been successfully integrated in Fang and Zhang [2012]. In addition, multiview active learning and semi-supervised learning has been effectively integrated in a single-label setting [Wang and Zhou 2008] (e.g., recommending examples in which different views predict different labels). This leads us to suspect that the combination of these frameworks could be useful in reducing the annotation effort in an MLL setting.

—*Multitask learning* (MTL). In this setting, several tasks that share a common representation are learned together. This setting is slightly different from MLL (e.g., the input space can be the same, but not the set of instances [Evgeniou and Pontil 2004]) but connections between these two approaches have not been deeply studied. MTL has been integrated with multiview learning [Zhang and Huan 2012], leading us to think that relationships between MLL and multiview learning may offer interesting research opportunities.

—*Hierarchical multilabel classification* (HMC), in contrast to *flat classification*, is another important challenge. In this kind of problem, examples can be associated with multiple labels, labels are organized in a hierarchical structure, and the classifier should take relationships among categories into account. The hierarchy may have the structure of a tree or a Directed Acyclic Graph (DAG), in which a child category may have more than one parent category. Some approaches are found in Barutcuoglu et al. [2006] and Vens et al. [2008].

8. CONCLUDING REMARKS

This article has presented an up-to-date tutorial with a description of the MLL framework and the main areas of application. The main proposals developed have been discussed, including new and challenging issues. The article has also described methodological aspects for the evaluation of the models: performance metrics, partitioning datasets, and significance tests. Finally, the main resources (datasets, repositories, bibliography, and software) for MLL learning have been summarized in the Appendix. MLL has been applied and demonstrated to be useful, time-saving, and effort-saving in numerous fields such as text, image, and video annotation; detection of emotions in music; medical diagnosis; and gene and protein function prediction. Its domains of application are increasing (e.g., speech emotion recognition or social network mining). Moreover, in MLL, researchers have found a challenging field of application, since datasets with a great number of instances, features, and labels are widely available. In addition, MLL faces challenges because of relationships between labels, high dimensionality of data, efficiency, and even its integration with other learning settings such as multi-instance, multiview, and the like. All of these factors make MLL a trending research area within the machine learning and data mining disciplines.

APPENDIX

A. RESOURCES

The aim of this Apendix is to provide a useful list of benchmark datasets, metrics for characterizing multilabel data, and software for MLL. In addition, Table XI summarizes other interesting resources for MLL such as bibliographic compilations, websites, tutorials, workshops, books, special issues, and PhD theses.

Table XI. MLL Resources

Reviews	
Multi-label classification: an overview	[Tsoumakas and katakis 2007]
Mining Multi-label Data	[Tsoumakas et al. 2010a]
A Tutorial on Multi-label Classification Techniques	[de Carvalho and Freitas 2009]
A Review on Multi-Label Learning Algorithms	[Zhang and Zhau 2014]
Bibliographic Compilations	
Multilabel Classification - 413 articles	[Gibaja and Ventura 2012]
MLKD - 161 articles	[MLKD 2012]
Tutorials	
Multi-label classification (TAMIDA 2010)	[Larrañage 2010]
Learning from multi-label data (ECML/PKDD 2009)	[Tsoumakas et al. 2009]
Advances in Multi-label Classification	[Read 2011]
Workshops	
1st International Workshop on Learning from Multi-Label Data (MLD'09) ECML/PKDD 2009	[MLD 2009]
2nd International Workshop on Learning from Multi-Label Data (MLD'10) ICML 2010	[MLD 2010]
Extreme Classification: Multi-Class & Multi-Label Learning with Millions of Categories NIPS 2013	[NIP 2013]
The First International Workshop on Learning with Weak Supervision (LAWS'12) ACML 2012	[LAW 2012]
Special Issues and Books	
Machine Learning. Special Issue on Learning from Multi-Label Data	[ML2 2012]
Multi-Label Dimensionality Reduction	[Sun et al. 2013]
PhD Theses	
Machine learning and data mining for yeast functional genomics	[Clare 2003]
Large Margin Multiclass Learning: Models and Algorithms	[Aiolli 2004]
Multilabel Classification over Category Taxonomies	[Cai 2008]
Scalable Multi-label Classification	[Read 2010]
Learning with Limited Supervision by Input and Output Coding	[Zhang 2012]
Ensembles for predicting structured outputs	[Kocev 2012]
Modelos de aprendizaje basados en programación genética para Clasificación Multietiqueta	[Avila 2013]

A.1. Benchmark Datasets

A.1.1. Metrics about Datasets. Before carrying out experiments over multilabel data, it is important to measure some characteristics of the dataset that can influence the performance of the developed proposals. Let $S = \{(\mathbf{x}_i, Y_i)|1 \leq i \leq m\}$ be a multi-label dataset of m instances.

The *label cardinality* (LCard) and *label density* (LDen) [Tsoumakas and Katakis 2007] metrics measure how multilabeled a dataset is. Cardinality is the average number of labels per pattern (see Equation (22)). Density is the cardinality divided by the total number of labels, and it is used to compare datasets with different numbers of labels (see Equation (23)):

$$LCard(S) = \frac{1}{m}\sum_{i=1}^{m}|Y_i| \tag{22}$$

$$LDen(S) = \frac{1}{m}\sum_{i=1}^{m}\frac{|Y_i|}{q} = \frac{LCard(S).}{q} \tag{23}$$

In Tsoumakas et al. [2010b] and Zhang and Zhang [2010], the *Distinct Labelsets* (DL) is described as the number of different label combinations in the dataset:

$$DL(S) = |Y \subseteq \mathcal{L}|\exists(\mathbf{x}, Y) \in S|. \tag{24}$$

In Read [2008] and Zhang and Zhang [2010], the *Proportion of Distinct Labelsets* (PDL) is defined as the number of distinct label subsets relative to the total number of examples:

$$PDL(S) = \frac{DL(S)}{m}. \tag{25}$$

The *diversity* [Tsoumakas et al. 2010b] is defined as the percentage of the bound of labels sets that the distinct represents (that is really in the dataset). In Read [2010], another two measures are introduced that provide information about the uniformity of the labeling scheme. The first one, the *Proportion of Unique Label Combinations*, PUniq, is the proportion of labelsets that are unique across the total number of examples (see Equation 26), and the second one, PMax, represents the proportion of examples associated with the most frequently occurring labelsets. It is computed as in Equation (27) where *count* is the frequency of Y as a labelset in S:

$$PUniq(S) = \frac{|Y \subseteq \mathcal{L}|!\exists \mathbf{x} : (\mathbf{x}, Y) \in S|}{m}. \tag{26}$$

$$PMax(S) = max_{Y \subseteq \mathcal{L}} \frac{count(Y, S)}{m}. \tag{27}$$

According to Read [2010], high PUniq indicates irregular labeling, and, when PMax is also high, the data present *label skew* (defined in Section 2.1). A complete summary of meta-features used to characterize datasets can be found in Chekina et al. [2011].

A.1.2. Benchmark Datasets. In this section, the main datasets that have been used in MLL are presented. Table XII summarizes their main characteristics.[2] Items in the table are ordered according a somewhat rough overall complexity measure used by Read [2010] and Madjarov et al. [2012], consisting in the product of $labels \times instances \times features$. The double line separates large datasets.

—Flags [Gonçalves et al. 2013]. This small dataset contains details of various nations and their flags. The multilabel classification task is to predict the seven colors that appear on the flags (e.g., *red*, *green*, *yellow*, etc.). It has 194 instances and 19 attributes about area, population, presence of triangles, religion of the country, and the like. In Gonçalves et al. [2013], it was used to test an evolutionary algorithm that optimizes labeling ordering in CC transformation.
—Emotions [Trohidis et al. 2008]. Also called Music [Read 2010], is a small dataset to classify music into the emotions that it evokes according to the Tellegen-Watson-Clark model of mood: *amazed-surprised*, *happy-pleased*, *relaxing-clam*, *quiet-still*, *sad-lonely*, and *angry-aggressive*. It consists of 593 songs, 6 labels, and 72 features falling into two categories: rhythmic and timbre. It has been intensively used in research for detecting emotions in music and as a benchmark dataset [Trohidis et al. 2008; Ma et al. 2009].
—Birds [Briggs et al. 2013]. The goal of this dataset is to predict the set of bird species that are present given a 10-second audio clip. The full dataset consists of 645 audio recordings and 19 species of birds that may be simultaneously vocalizing. It was used in a conference competition [Briggs et al. 2013].
—Yeast [Elisseeff and Weston 2001] contains 103 numeric features about micro-array expressions and phylogenetic profiles for 2,417 yeast genes. Each gene is annotated with a subset of 14 functional categories (e.g., *metabolism*, *energy*, etc.) from the

[2]They can be downloaded from http://www.uco.es/grupos/kdis/kdiswiki/index.php/Resources.

Table XII. ML Datasets

Dataset	Domain	Inst.	Feat.	q	Card.	Dens.	Dist.	Download
Flags	images(toy)	194	9c+10n	7	3.392	0.485	54	[Tsoumakas et al. 2011] [Bache and Lichman 2013]
Emotions	music	593	72n	6	1.869	0.311	27	[Tsoumakas et al. 2011]
Birds	audio	645	2c+258n	19	1.059	0.053	133	[Tsoumakas et al.2011]
Yeast	biology	2,417	103n	14	4.237	0.303	198	[Tsoumakas et al. 2011]
Scene	images	2,407	294n	6	1.074	0.179	15	[Tsoumakas et al. 2011]
Plant	biology	948	440n	12	1.078	0.089	32	˜[Xu 2013b]
CAL500	music	502	68n	174	26.044	0.150	502	[Tsoumakas et al. 2011]
Human	biology	3,108	440n	14	1.185	0.084	85	˜[Xu 2013b]
Genbase	biology	662	1,186b	27	1.252	0.046	32	[Tsoumakas et al. 2011]
Medical	text	978	1,449b	45	1.245	0.028	94	[Tsoumakas et al. 2011]
Slashdot	text	3,782	1 ,079nb	22	1.18	0.053	156	[Read 2012]
Enron	text	1,702	1001b	53	3.378	0.064	753	[Tsoumakas et al. 2011] [Read 2012]
LangLog	text	1,460	1,004nb	75	1.180	0.015	304	[Read 2012]
Tmc2007-500	text	28,596	500b	22	2.219	0.100	1172	[Tsoumakas et al. 2011]
20ng	text	19,299	1,006nb	20	1.028	0.051	55	[Read 2012]
Mediamill	video	43,907	120n	101	4.376	0.043	6555	[Tsoumakas et al. 2011]
Corel5k	images	5,000	499b	374	3.522	0.009	3175	[Tsoumakas et al. 2011]
Corel16k(10samples)	images	13,811	500b	161	2.867	0.018	4937	[Tsoumakas et al. 2011]
Bibtex	text	7,395	1,836b	159	2.402	0.015	2856	[Tsoumakas et al. 2011]
Yahoo(Health)	text(web)	9,205	30,605nb	32	1.644	0.051	335	[Ueda and Saito 2002b]
Yahoo(Arts)	text(web)	7,484	23,146nb	26	1.653	0.063	599	[Ueda and Saito 2002b]
Yahoo(Business)	text(web)	11,214	21,924nb	30	1.598	0.053	233	[Ueda and Saito 2002b]
Yahoo(Reference)	text(web)	8,027	39,679nb	33	1.174	0.035	275	[Ueda and Saito 2002b]
Yahoo(Science)	text(web)	6,428	37 ,187nb	40	1.449	0.036	457	[Ueda and Saito 2002b]
IMDB	text	120,919	1001nb	28	2.00	0.071	4503	[Read 2012]
Yahoo(Education)	text(web)	12,030	27 ,534nb	33	1.463	0.044	511	[Ueda and Saito 2002b]
Yahoo(Entertainment)	text(web)	12,730	3 ,2001nb	21	1.413	0.067	337	[Ueda and Saito 2002b]
Yahoo(Recreation)	text(web)	12,828	3 ,0324nb	22	1.428	0.064	530	[Ueda and Saito 2002b]
Yahoo(Computers)	text(web)	12,444	34 ,096nb	33	1.507	0.045	428	[Ueda and Saito 2002b]
Delicious	text(web)	16,105	500b	983	19.02	0.019	15806	[Tsoumakas et al. 2011]
Yahoo(Social)	text(web)	12,111	52 ,350nb	39	1.279	0.032	361	[Ueda and Saito 2002b]
Yahoo(Society)	text(web)	14,512	31 ,802nb	27	1.670	0.061	1054	[Ueda and Saito 2002b]
EUR-Lex(subjectmatters)	text	19,348	5 ,000n	201	2.213	0.011	2504	[Tsoumakas et al. 2011]
Rcv1v2(subset4)	text	6 ,000	47,229n	101	2.484	0.025	816	[Tsoumakas et al. 2011]
Rcv1v2(subset5)	text	6 ,000	47,235n	101	2.642	0.026	946	[Tsoumakas et al. 2011]
Rcv1v2(subset1)	text	6 ,000	47,236n	101	2.880	0.029	1028	[Tsoumakas et al. 2011]
Rcv1v2(subset2)	text	6 ,000	47,236n	101	2.634	0.026	954	[Tsoumakas et al. 2011]
Rcv1v2(subset3)	text	6 ,000	47,236n	101	2.614	0.026	939	[Tsoumakas et al. 2011]
Tmc2007	text	28 ,596	49,060b	22	2.158	0.098	1341	[Tsoumakas et al. 2011]
Bookmarks	text	87 ,856	2,150b	208	2.028	0.010	18716	[Tsoumakas et al. 2011]
EUR-Lex(directorycodes)	text	19 ,348	5 ,000n	412	1.292	0.003	1615	[Tsoumakas et al. 2011] [Loza and Furnkranz 2013]
EUR-Lex(eurovocdescript.)	text	19 ,348	5 ,000n	3 ,993	5.31	0.001	16467	[Tsoumakas et al. 2011] [Loza and Furnkranz 2013]

In the Feat. Column n, b, nb, and c refer to numeric, binary, numeric with binary values, and categorical attributes, respectively.

top level of the functional catalog FunCat. It has been used in protein and gene function classification [Clare and King 2001; Diplaris et al. 2005] and intensively as benchmark. It is also a benchmark for HMC [Blockeel et al. 2006; Barutcuoglu et al. 2006].

—Scene [Boutell et al. 2004] has 2,407 images annotated with up to six concepts (e.g., *beach*, *mountain*, etc.). Each one is described with 294 visual numeric features corresponding to spatial color moments in the LUV space. It is relatively small but widely used as a benchmark and for semantic scene classification [Boutell et al. 2004].

—PLANT and HUMAN [Xu 2013a] are two datasets used to predict the subcellular locations of proteins according to their sequences. Some proteins can simultaneously exist at, or move between, two or more different location sites (e.g., *nucleus*, *golgi apparatus*, *mitochondrion*, etc.). These datasets contain 948 and 3,108 protein sequences for plant and human species, respectively, with 440 numeric features (20 amino acid, 20 pseudo-amino acid, and 400 dipeptide compositions). There are 12 positions or labels for plants and 14 for humans.
—COMPUTER AUDITION LAB 500 (CAL500) [Turnbull et al. 2008] is a dataset composed of 502 popular Western songs, represented by 68 acoustic features, each of which has been manually annotated by at least three human annotators who employed a vocabulary of 174 tags. These tags span six semantic categories: instrumentation, vocal characteristics, genres, emotions, acoustic quality of the song, and usage terms (e.g., "I would like listen to this song while *driving*"). It has been used as a benchmark and in semantic annotation and retrieval of music [Turnbull et al. 2008].
—GENBASE [Diplaris et al. 2005] is a dataset for protein function classification. Each of the 662 instances is a protein chain represented using a motif sequence vocabulary of fixed size. Thus, each sequence is encoded as a binary array where each bit is 1 if the corresponding motif is present and 0 otherwise. Each label identifies the functional family of the sequence. It has been mainly used as a benchmark dataset.
—MEDICAL [Pestian et al. 2007] is based on the data made available during the Computational Medicine Center's 2007 Medical Natural Language Processing Challenge. It consists of 978 clinical free-text radiology reports labeled with ICD-9-CM disease codes. The dataset has 45 codes and 1,149 binary attributes representing if a certain term is or is not in the report. It has been used as a benchmark dataset.
—SLASHDOT [Read 2010] is a collection of 3,782 texts mined from Slashdot[3] and labeled with 22 subject categories (e.g., *enternaiment*, *interviews*, *games*, etc.). Each document has 1,079 binary features representing the presence of a term. It has been used as a benchmark and in the field of multilabel data streams [Read et al. 2010].
—ENRON [Read et al. 2008] is a subset of the Enron e-mail text corpus. It is based on a collection of e-mails exchanged between the Enron Corporation employees, which were made available during a legal investigation. It contains 1,702 e-mails that were categorized into 53 topic categories (e.g., *company strategy*, *humor,* and *legal advice*). It has been mainly used as a benchmark dataset.
—LANGUAGELOG (LANGLOG) [Read 2010] is a dataset with 1,460 instances and 1,004 binary features. It was compiled from the Language Log Forum,[4] which discusses various topics relating to language. It has 75 topics (e.g., *prepositions*, *punctuation*, *relative_clauses*, etc.) and has been used as benchmark.
—REUTERS [Lewis et al. 2005]. The Reuters-RCV1 dataset is a well-known benchmark for text classification methods [Sebastiani 2002; Rubin et al. 2012]. Lewis et al. [2005] made some corrections to the RCV1 dataset resulting a new dataset called RCV1-v2. It has five subsets, each one with 6,000 news articles assigned into one or more of 101 topics. In Read [2010], the attribute space was reduced to 500. Reuters-21578 [Bache and Lichman 2013] is other Reuters dataset commonly used in MLL [Godbole and Sarawagi 2004] and consisting of a set of 21,578 news stories that appeared on the Reuters news wire in 1987. Is has also been used in HMC [Cesa-Bianchi et al. 2006].
—20 NEWSGROUP (20NG) [Lang 2008] is a compilation of around 19,300 posts to 20 different newsgroups. The names of the groups are the 20 possible labels (e.g., *rec.motorcycles*, *sci.electronics*, etc.), and binary attributes represent the presence

[3]http://slashdot.org.
[4]http://languagelog.ldc.upenn.edu/nll.

or absence of a word in the post. Around 1,000 posts are available for each of group. It has been mainly used as a benchmark dataset [Read 2010].

—MEDIAMILL [Snoek et al. 2006] is a multimedia dataset for generic video indexing that was extracted from the TRECVID 2005/2006 benchmark.[5] This benchmark dataset contains 85 hours of international broadcast news data categorized into 101 semantic concepts (e.g., *car*, *golf*, *bird*, etc.), and each video instance is represented as a numeric vector of 120 features including visual and textual information. It has been also used in the field of multilabel data streams [Read et al. 2010].

—ECCV2002 OR COREL5K [Duygulu et al. 2002] is based on 5,000 Corel images, 4,500 of which are used for training and the remaining 500 for testing. Images were segmented and then only regions larger than a threshold were clustered into 499 blobs using k-means, which are the features used to describe the image. JMLR2003 OR COREL16K [Barnard et al. 2003] is derived from ECCV2002 by eliminating infrequent labels. It is a popular benchmark for image annotation and retrieval [Nasierding and Kouzani 2010] and has been also used in MIML [Zhou and Zhang 2006].

—BIBTEX [Katakis et al. 2008] is based on the data of the ECML/PKDD 2008 discovery challenge. This benchmark dataset contains 7,395 bibtex entries from the BibSonomy[6] social bookmark and publication sharing system, annotated with a subset of the tags assigned by users (e.g., *statistics*, *quantum*, *data mining* etc.). The title and abstract of entries were used to construct features using the boolean bag-of-words model.

—YAHOO! [Ueda and Saito 2002a] is a dataset to categorize webpages and consists of 11 of the 14 top-level categories of the Yahoo! directory. About 30–45% of the pages were multilabeled over the 11 text classification problems. Attributes represent the presence of a word in the document. It has been used as a benchmark and in the field of text categorization [Ueda and Saito 2002a; Rubin et al. 2012].

—IMDB [Read 2010] contains 120,919 movie plot text summaries from the Internet Movie Database[7] labeled with one or more genres (e.g., *comedy*, *western*, *documentary*, etc). The dataset has 1,001 binary attributes that follow the binary bag-of-words model. It has been used as a benchmark and also in the field of multilabel data streams [Read et al. 2012].

—DELICIOUS [Tsoumakas et al. 2008] contains textual data from the Delicious[8] website along with their tags (e.g., *academia*, *airline*, *algorithm*, etc.). It has 16,105 instances. In Read [2010], it is highlighted that this dataset is a modified tagging problem where the label space was not predefined prior to labeling, and the size of the label space (983 labels) is greater than the size of the input space (500 attributes). Attributes are binary and represent the presence of a word in the document. It has been used in works focused on its great number of labels [Zhang et al. 2010].

—EUR-LEX [Loza and Fürnkranz 2008]. The EUR-Lex text collection contains 19,348 documents on European Union law (e.g., treaties, legislation, legislative proposals, etc.), which are indexed according to several orthogonal categorization schemes to allow for multiple search facilities. The most important categorization is provided by the EUROVOC descriptors, which form a topic hierarchy with almost 4,000 categories regarding different aspects of European law. Attributes are numeric and represent the *tf*idf* measure for each term in the document. It has been used in the domain of document classification [Rubin et al. 2012] and, due to its dimensionality, to test MLL algorithms designed for large datasets [Loza and Fürnkranz 2008].

[5]http://www-nlpir.nist.gov/projects/trecvid/.
[6]http://www.bibsonomy.org/.
[7]http://www.imdb.com/.
[8]https://delicious.com/.

Table XIII. Examples of a Multilabel Dataset with Three Features and Four Labels in Mulan and Meka Formats

(a) Mulan Format

ARFF FILE
@relation MultiLabelExample
@attribute feature1 numeric
@attribute feature2 numeric
@attribute feature3 numeric
@attribute label1 {0, 1}
@attribute label2 {0, 1}
@attribute label3 {0, 1}
@attribute label4 {0, 1}
@data
4.1,2.9,3.7,0,0,1,1
XML FILE
<labels xmlns="http://mulan.sourceforge.net/labels">
<label name="label1"></label>
<label name="label2"></label>
<label name="label3"></label>
<label name="label4"></label>
</labels>

(b) Meka Format

ARFF FILE
@relation 'Example_Dataset: -C 4 -split-percentage 50'
@attribute label1 {0, 1}
@attribute label2 {0, 1}
@attribute label3 {0, 1}
@attribute label4 {0, 1}
@attribute feature1 numeric
@attribute feature2 numeric
@attribute feature3 numeric
@data
0,0,1,1,4.1,2.9,3.7

—TMC2007 [Srivastava and Zane-Ulman 2005]. The SIAM Text Mining Competition (TMC) 2007 dataset is a subset of the Aviation Safety Reporting System (ASRS) dataset. This benchmark contains 28,596 aviation safety free-text form reports that flight crews submit after completion of each flight regarding problems that took place during a flight. The goal is to label the documents with respect to what types of problems they describe. The dataset has 49,060 binary attributes corresponding to the presence of terms in the collection. The safety reports are provided with 22 labels, each of them representing a problem type. In Tsoumakas et al. [2010b], a χ^2 feature ranking method was used separately for each label, and the top 500 features based on their maximum rank over all labels were selected. Text representation follows the boolean bag-of-words model.

—BOOKMARKS [Katakis et al. 2008] is based on the data of the ECML/PKDD 2008 discovery challenge and contains metadata for bookmark entries from the Bibsonomy system (e.g., the URL of the webpage, a URL hash, a description of the webpage, etc). It has 208 labels and 2,150 binary attributes that represent terms in the document. It was used in an extensive experimental comparison of MLL algorithms [Madjarov et al. 2012], and some of experimenters were not able to finish with this dataset.

A.1.3. MLL Dataset Formats. Many of the MLL benchmark datasets are available in Mulan or Meka frameworks, both based in the *arff* format [Hall et al. 2009]. Mulan [Tsoumakas et al. 2011] datasets use two files. The first one is an *arff* file where labels should be nominal attributes with 0 or 1 values. The second one is an XML file where labels are defined to allow hierarchical relationships to be represented among them. An example of a Mulan dataset with three features and four labels is presented in Table XIII. Meka [Read 2012] datasets are also in *arff* and use one attribute for each target or label. The dataset options (like the -C option for the number of labels, q) can be included either in the *@relation* tag of the *arff* file or in the command line. Meka allows also the train/test split percentage to be stored in the *@relation* name, where a colon (:) is used to separate the dataset name and the option. An example of a Meka dataset with three features and four labels is presented in Table XIII.

A.2. Software

This section is focused in software that implements MLL baseline methods. Table XIV summarizes the features of the main APIs and software packages. Mulan

Table XIV. Main Features of Software Packages for MLL

	Mulan	Meka	Lamda Gr.	Zhang's site
RSL	Copy, ignore, select	RT		
BR-based	BR	BR, BRq		
LC	LP, PS, EPS	LP(LC), PS, EPS		
PW	CLR			
Dependences	CC, ECC	CC, ECC, PCC, MCC(Montecarlo)	ML-LOC	Bayesian Networks
Lazy	BRkNN, IBLR, MLkNN		MLkNN	MLkNN
ANN	BPMLL, MMP		BPMLL	BPMLL, ML-RBF
SVMs				RankSVM
Meta-learners	RAkEL, HOMER, LPBR(Subset Learner), AdaBoost.MH, 2BR	BaggingML, 2BR(MBR), EnsembleML, RandomSubspaceML		
Other	HMC	Majority labelset, Conditional Dependence Network (CDN), Unsupervised (EM)	InsDiff, MIML, WELL, Multi-modal MIML	Naive BayesML, LIFT, MIML
Utils	Dimensionality Reduction, ConverterLibSVM, ConverterCLUS, Stratification, Statistics of Data, Thresholding	Wrapper Mulan	Dimens. Reduction	
Multi-target		CC, CR, Nearest Set Replacement (NSR), EnsembleMT, BaggingMT		
GUI	No	Yes	No	No
Languaje	Java	Java	Matlab	Matlab
License	GNU GPL	GNU GPL	Free for academic purpose	Free for academic purpose

[Tsoumakas et al. 2011] is an open-source Java API for multilabel classification that is built on top of Weka [Hall et al. 2009] and implements many transformation methods like BR, LP, copy methods, EPS, or CLR and multilabel algorithms like ML-kNN, RAkEL, HOMER, or BP-MLL. It also provides support for the evaluation of the models, has a number of multilabel datasets, and offers methods for basic feature selection. Another API is Meka [Read 2012], a multilabel extension for the Weka framework that also provides an open-source Java implementation of the PS and CC methods. Written in Java, it is also compatible with Mulan and provides support for development, running, and evaluation of multilabel and multitarget (multiclass outputs instead binary outputs) classifiers.

On the LAMDA research group's website (see Table XI) several Matlab packages are provided with the implementation of BP-MLL or ML-kNN, among others. From the Min-Ling Zhang's website (see Table XI), the code of several multilabel models, such as ML-kNN, Rank-SVM, BP-MLL, or ML-RBF, can be also downloaded. LibSVM [Chang and Lin 2011] is a library for SVMs that allows the handling of multilabel data. Specifically, the LP and BR transformation methods are implemented.

Some data mining software suites include multilabel capabilities. Thus, Orange [Laboratory 2013] includes a multitarget add-on with methods such as BR, CC, or ECC, and Scikit-learn [Pedregosa et al. 2011] provides the RPC and BR approaches. All the material (including the repository of manipulated datasets and software for dataset characteristic extraction) used by Chekina et al. [2011] for dataset characteristics extraction is also available.[9] Finally, the PCT framework is also available.[10]

[9]http://www.ise.bgu.ac.il/faculty/liorr/lena/meta.html.
[10]http://dtai.cs.kuleuven.be/clus/.

REFERENCES

Qaisar Abbas, M. E. Celebi, Carmen Serrano, Irene Fondón, and Guangzhi Ma. 2013. Pattern classification of dermoscopy images: A perceptually uniform model. *Pattern Recognition* 46, 1 (2013), 86–97.

Fabio Aiolli. 2004. *Large Margin Multiclass Learning: Models and Algorithms*. PhD Dissertation. Università degli Studi di Pisa.

Martin Antenreiter, Ronald Ortner, and Peter Auer. 2009. Combining classifiers for improved multilabel image classification. In *Proceedings of the 1st Workshop on Learning from Multilabel Data (MLD) Held in Conjunction with ECML/PKDD*. 16–27.

J. L. Ávila. 2013. *Modelos de aprendizaje basados en programación genética para clasificación multietiqueta*. PhD Dissertation. University of Córdoba.

Jose L. Ávila, Eva. Gibaja, and Sebastián Ventura. 2010. Evolving multi-label classification rules with gene expression programming: A preliminary study. In *Hybrid Artificial Intelligence Systems*. Lecture Notes in Computer Science, Vol. 6077. Springer, Berlin, Chapter 2, 9–16.

Jose L. Ávila, Eva Gibaja, Amelia Zafra, and Se Ventura. 2011. A gene expression programming algorithm for multi-label classification. *Journal of Multiple-Valued Logic and Soft Computing* 17, 2–3 (2011), 183–206.

K. Bache and M. Lichman. 2013. UCI Machine Learning Repository. Retrieved from http://archive.ics.uci.edu/ml.

Kobus Barnard, Pinar Duygulu, Nando de Freitas, David Forsyth, David Blei, and Michael I. Jordan. 2003. Matching words and pictures. *Journal of Machine Learning Research* 3 (2003), 1107–1135.

Zafer Barutcuoglu, Robert E. Schapire, and Olga G. Troyanskaya. 2006. Hierarchical multi-label prediction of gene function. *Bioinformatics* (Oxford, England) 22, 7 (April 2006), 830–836.

Plaban K. Bhowmick, Anupam Basu, Pabitra Mitra, and Abhisek Prasad. 2010. Sentence level news emotion analysis in fuzzy multi-label classification framework. *Research in Computer Science, Special Issue: Natural Language Processing and Its Applications* 46 (2010), 143–154.

Hendrik Blockeel, Luc De Raedt, and Jan Ramon. 1998. Top-down induction of clustering trees. In *Proceedings of the 15th International Conference on Machine Learning (ICML'98)*. Morgan Kaufmann, San Francisco, CA, 55–63.

Hendrik Blockeel, Leander Schietgat, Jan Struyf, Sašo Dzěroski, and Amanda Clare. 2006. Decision trees for hierarchical multilabel classification: A case study in functional genomics. In *Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD (Lecture Notes in Computer Science)*, Vol. 4213. 18–29.

Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. 2004. Learning multi-label scene classification. *Pattern Recognition* 37, 9 (Sept. 2004), 1757–1771.

Leo Breiman, Jerome Friedman, R. A. Olshen, and Charles J. Stone. 1984. *Classification and Regression Trees*. Wadsworth.

Forrest Briggs, Raviv Raich, Konstantinos Eftaxias, Zhong Lei, and Yonghong Huang. 2013. The ninth annual MLSP competition: Overview. In *Proceedings of the 2013 IEEE International Workshop on Machine Learning for Signal Processing*.

Klaus Brinker. 2006. On active learning in multi-label classification. In *From Data and Information Analysis to Knowledge Engineering*, Myra Spiliopoulou, Rudolf Kruse, Christian Borgelt, Andreas Nürnberger, and Wolfgang Gaul (Eds.). Springer, Berlin, 206–213.

Klaus Brinker, Johannes Fürnkranz, and Eyke Hüllermeier. 2006. A unified model for multilabel classification and ranking. In *Proceedings of the 17th European Conference on Artificial Intelligence*. IOS Press, Amsterdam, The Netherlands, 489–493.

Lijuan Cai. 2008. *Multilabel Classification over Category Taxonomies*. PhD Dissertation. Brown University.

Nicolò Cesa-Bianchi, Claudio Gentile, and Luca Zaniboni. 2006. Hierarchical classification: Combining bayes with SVM. In *Proceedings of the 23rd International Conference on Machine Learning (ICML'06)*. 177–184.

Allen Chan and Alex A. Freitas. 2006. A new ant colony algorithm for multi-label classification with applications in bioinfomatics. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO'06)*. ACM, New York, NY, 27–34.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2, 3 (2011), 27:1–27:27.

Lena Chekina, Lior Rokach, and Bracha Shapira. 2011. Meta-learning for selecting a multi-label classification algorithm. In *Proceedings of the 11th International Conference on Data Mining Workshops (ICDMW'11)*. 220–227.

Weizhu Chen, Jun Yan, Benyu Zhang, Zheng Chen, and Qiang Yang. 2007. Document transformation for multi-label feature selection in text categorization, In *Proceedings of the IEEE International Conference on Data Mining*. 451–456.

Weiwei Cheng and Eyke Hüllermeier. 2009. Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning* 76(2–3) (Sept. 2009), 211–225.

Everton Alvares Cherman, Jean Metz, and Maria Carolina Monard. 2012. Incorporating label dependency into the binary relevance framework for multi-label classification. *Expert Systems Applications* 39, 2 (2012), 1647–1655.

Kuo-Chen Chou, Zhi-Cheng Wu, and Xuan Xiao. 2011. iLoc-Euk: A multi-label classifier for predicting the subcellular localization of singleplex and multiplex eukaryotic proteins. *PLoS ONE* 6, 3 (2011), e18258.

Patrick Marques Ciarelli, Elias Oliveira, Claudine Badue, and Alberto Ferreira De Souza. 2009. Multi-label text categorization using a probabilistic neural network. *International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)* 1 (2009), 133–144.

Amanda Clare. 2003. *Machine Learning and Data Mining for Yeast Functional Genomics*. PhD Dissertation. University of Wales, Aberystwyth.

Amanda Clare and Ross D. King. 2001. Knowledge discovery in multi-label phenotype data. In *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD'01) (Lecture Notes in Computer Science)*, Vol. 2168. 42–53.

He Cong and Loh H. Tong. 2008. Grouping of TRIZ inventive principles to facilitate automatic patent classification. *Expert Systems with Applications* 34, 1 (2008), 788–795.

Koby Crammer and Yoram Singer. 2003. A family of additive online algorithms for category ranking. *Journal of Machine Learning Research* 3 (March 2003), 1025–1058.

André de Carvalho and Alex Freitas. 2009. A tutorial on multi-label classification techniques. In *Foundations of Computational Intelligence Volume 5*. Studies in Computational Intelligence, Vol. 205. Springer, Berlin, 177–195.

Francesco De Comité, Rémi Gilleron, and Marc Tommasi. 2003. Learning multi-label alternating decision trees from texts and data. In *Proceedings of the 3rd International Conference on Machine Learning and Data Mining in Pattern Recognition (MLDM'03)*. Springer-Verlag, Berlin, 35–49.

Krzysztof Dembczyński, Weiwei Cheng, and Eyke Hüllermeier. 2010. Bayes optimal multilabel classification via probabilistic classifier chains. In *Proceedings of th 27th International Conference on Machine Learning (ICML'10)*. 279–286.

Krzysztof Dembczyński, Willem Waegeman, Weiwei Cheng, and Eyke Hüllermeier. 2012. On label dependence and loss minimization in multi-label classification. *Machine Learning* 88, 1 (2012), 5–45.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistics Society -B* 39(1) (1977), 1–38.

Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7 (2006), 1–30.

Anastasios Dimou, Grigorios Tsoumakas, Vasileios Mezaris, Ioannis Kompatsiaris, and Ioannis Vlahavas. 2009. An empirical study of multi-label learning methods for video annotation. In *Proceedings of the International Workshop on Content-Based Multimedia Indexing (CBMI'09)*. IEEE Computer Society, Los Alamitos, CA, 19–24.

Sotiris Diplaris, Grigorios Tsoumakas, Pericles Mitkas, and Ioannis Vlahavas. 2005. Protein classification with multiple algorithms, In *Proceedings of the 10th Panhellenic Conference on Informatics (PCI'05). Advances in Informatics* 448–456.

R. Duwairi and A. Kassawneh. 2008. A framework for predicting proteins 3D structures. In *Proceedings of the IEEE/ACS International Conference on Computer Systems and Application*. Washington, DC, 37–44.

Pinar Duygulu, Kobus Barnard, Nando de Freitas, and David Forsyth. 2002. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of the 7th European Conference on Computer Vision (ECCV'02) (Lecture Notes in Computer Science)*, Vol. 2353. 97–112.

Andre Elisseeff and Jason Weston. 2001. A kernel method for multi-labelled classification. In *Advances in Neural Information Processing Systems (NIPS)*, Vol. 14. 681–687.

Andrea Esuli and Fabrizio Sebastiani. 2009. Active learning strategies for multi-label text classification. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*. Lecture Notes in Computer Science, Vol. 5478. Springer-Verlag, 102–113.

Theodoros Evgeniou and Massimiliano Pontil. 2004. Regularized multi–task learning. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*. 109–117.

Zheng Fang and Zhongfei (Mark) Zhang. 2012. Simultaneously combining multi-view multi-label learning with maximum margin classification. In *ICDM*. IEEE Computer Society.

Yoav Freund and Robert E. Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer System Sciences* 55, 1 (1997), 119–139.

Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. 2008. Multilabel classification via calibrated label ranking. *Machine Learning* 73(2) (2008), 133–153.

Nadia Ghamrawi and Andrew McCallum. 2005. Collective multi-label classification. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM'05)*. 195–200.

Eva Gibaja and Sebastián Ventura. 2012. Multilabel Classification Library. Retrieved from http://www.citeulike.org/group/4310.

Shantanu Godbole and Sunita Sarawagi. 2004. Discriminative methods for multi-labeled classification. In *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 22–30.

Eduardo Corrêa Gonçalves, Alexandre Plastino, and Alex Alves Freitas. 2013. A genetic algorithm for optimizing the label ordering in multi-label classifier chains. In *Proceedings of the IEEE 25th International Conference on Tools with Artificial Intelligence (ICTAI'13)*. 469–476.

Teresa Gonçalves and Paulo Quaresma. 2004. Using IR techniques to improve automated text classification, In *Proceedings of the 9th International Conference on Applications of Natural Language to Information Systems*. 374–379.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: An update. *SIGKDD Explorations* 11(1) (2009).

Jianjun He, Hong Gu, and Zhelong Wang. 2012. Multi-instance multi-label learning based on Gaussian process with application to visual mobile robot navigation. *Information Sciences* 190 (2012), 162–177.

Daniel Hsu, Sham Kakade, John Langford, and Tong Zhang. 2009. Multi-label prediction via compressed sensing. In *Advances in Neural Information Processing Systems 22 (NIPS)*, Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta (Eds.). 772–780.

Xiaolei Huang, Wei Wang, Zhiyun Xue, Sameer Antani, L. Rodney Long, and Jose Jeronimo. 2008. Tissue classification using cluster features for lesion detection in digital cervigrams. In *Proceedings SPIE Medical Imaging*.

Eyke Hüllermeier, Johannes Fürnkranz, Weiwei Cheng, and Klaus Brinker. 2008. Label ranking by learning pairwise preferences. *Artificial Intelligence* 172 (2008), 1897–1916.

Marios Ioannou, George Sakkas, Grigorios Tsoumakas, and Ioannis P. Vlahavas. 2010. Obtaining bipartitions from score vectors for multi-label classification. In *Proceedings of the 22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI'10)*. 409–416.

Shuiwang Ji, Lei Tang, Shipeng Yu, and Jieping Ye. 2010. A shared-subspace learning framework for multi-label classification. *ACM Transactions on Knowledge Discovery from Data* 4, 2 (2010), 1–29.

Aiwen Jiang, Chunheng Wang, and Yuanping Zhu. 2008. Calibrated rank-SVM for multi-label image categorization. In *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN'08)*. 1450–1455.

Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas. 2008. Multilabel text classification for automated tag suggestion. In *Proceedings of the ECML/PKDD 2008 Discovery Challenge*. 1–9.

Kentaro Kawai and Yoshimasa Takahashi. 2009. Identification of the dual action antihypertensive drugs using TFS-based support vector machines. *Chem-Bio Informatics Journal* 4 (2009), 44–51.

Dragi Kocev. 2012. *Ensembles for Predicting Structured Outputs*. PhD Dissertation. Jǒzef Stefan International Postgraduate School.

Dragi Kocev, Celine Vens, Jan Struyf, and Sašo Džeroski. 2007. Ensembles of multi-objective decision trees. In *Proceedings of the 18th European Conference on Machine Learning (ECML'07)*. Springer-Verlag, Berlin, 624–631.

Artus Krohn-Grimberghe, Lucas Drumond, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2012. Multi-relational matrix factorization using Bayesian personalized ranking for social network data. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining (WSDM'12)*. ACM, New York, NY, 173–182.

Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. 2009. Attribute and simile classifiers for face verification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'09)*.

FRI UL Bioinformatics Laboratory. 2013. Orange Multitarget Add-on for Orange Data Mining Software Package. Retrieved from http://pypi.python.org/pypi/Orange-Multitarget.

J. Lafferty, A. McCallum, and F. C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labelling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML'01)*. 282–289.

Ken Lang. 2008. The 20 Newsgroup Dataset. Retrieved from http://people.csail.mit.edu/jrennie/20Newsgroups/.

Pedro Larrañaga. 2010. Multi-label Classification. Retrieved from http://www.dynamopro.org/IMG/pdf/tamida2010-larranaga.pdf.

Boris Lauser and Andreas Hotho. 2003. Automatic multi-label subject indexing in a multilingual environment. In *Proceedings of the 7th European Conference, ECDL (Lecture Notes in Computer Science)*, Vol. 2769. 140–151.

LAWS. 2012. *Proceedings of the 1st International Workshop on Learning with Weak Supervision (LAWS'12)*. Retrieved from http://cse.seu.edu.cn/conf/LAWS12/files/LAWS'12.pdf.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2005. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research* 5 (2005), 361–397.

Vivian F. López, Fernando de la Prieta, Mitsunori Ogihara, and Ding Ding Wong. 2012. A model for multilabel classification and ranking of learning objects. *Expert Systems with Applications* 39, 10 (2012), 8878–8884.

Eneldo Loza and Johannes Fürnkranz. 2007. An evaluation of efficient multilabel classification algorithms for large-scale problems in the legal domain. In *Proceedings of the LWA 2007: Lernen - Wissen - Adaption*, Alexander Hinneburg (Ed.). 126–132.

Eneldo Loza and Johannes Fürnkranz. 2008. Efficient pairwise multilabel classification for large-scale problems in the legal domain. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD'08)*. Springer-Verlag, 50–65.

Eneldo Loza and Johannes Fürnkranz. 2010. Efficient multilabel classification algorithms for large-scale problems in the legal domain. In *Semantic Processing of Legal Texts*. Lecture Notes in Computer Science, Vol. 6036. 192–215.

Eneldo Loza and Johannes Fürnkranz. 2013. The EUR-Lex Dataset. Retrieved from http://www.ke.tu-darmstadt.de/resources/eurlex.

Eneldo Loza, Sang-Hyeun Park, and Johannes Fürnkranz. 2009. Efficient voting prediction for pairwise multilabel classification. In *Proceedings of the 17th European Symposium on Artificial Neural Networks (ESANN'09)*. 117–122.

Aiyesha Ma, Ishwar Sethi, and Nilesh Patel. 2009. Multimedia content tagging using multilabel decision tree. In *Proceedings of the 2009 11th IEEE International Symposium on Multimedia (ISM'09)*. 606–611.

Gjorgji Madjarov, Dejan Gjorgjevikj, and Sašo Džeroski. 2011. Dual layer voting method for efficient multilabel classification. In *Proceedings of the 5th Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA) (Lecture Notes in Computer Science)*, Vol. 6669. 232–239.

Gjorgji Madjarov, Dragi Kocev, Dejan Gjorgjevikj, and Sašo Džeroski. 2012. An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition* 45, 9 (2012), 3084–3104.

M. A. Mammadov, A. M. Rubinov, and J. Yearwood. 2007. The study of drug-reaction relationships using global optimization techniques. *Optimization Methods Software* 22 (Feb. 2007), 99–126.

Andrew Kachites McCallum. 1999. Multi-label text classification with a mixture model trained by EM. In *Proceedings of the AAAI 99 Workshop on Text Learning*.

ML2. 2012. *Machine Learning*. Special Issue on Learning from Multi-Label Data.

MLD. 2009. *Proceedings of the 1st International Workshop on Learning from Multi-Label Data (MLD'09)*. Retrieved from http://lpis.csd.auth.gr/workshops/mld09/mld09.pdf.

MLD. 2010. *Proceedings of the 2nd International Workshop on Learning from Multi-Label Data (MLD'10)*. Retrieved from http://cse.seu.edu.cn/conf/MLD10/files/MLD'10.pdf.

MLKD. 2012. MLKD - Multilabel Library. Retrieved from http://www.citeulike.org/group/7105/tag/multilabel.

Arturo Montejo-Ráez and Luis Ureña López. 2006. Selection strategies for multi-label text categorization. In *Advances in Natural Language Processing*. Lecture Notes in Computer Science, Vol. 4139. 585–592.

Pio Nardiello, Fabrizio Sebastiani, and Alessandro Sperduti. 2003. Discretizing continuous attributes in adaboost for text categorization. In *Advances in Information Retrieval*. Lecture Notes in Computer Science, Vol. 2633. 320–334.

Gulisong Nasierding and Abbas Z. Kouzani. 2010. Empirical study of multi-label classification methods for image annotation and retrieval. In *Proceedings of the International Conference on Digital Image Computing: Techniques and Applications (DICTA'10)*. 617–622.

Cao D. Nguyen, Tran A. Dung, and Tru H. Cao. 2005. Text classification for DAG-structured categories. In *Advances in Knowledge Discovery and Data Mining*. Lecture Notes in Computer Science, Vol. 3518. 1–18.

NIPS. 2013. Extreme Classification: Multi-Class & Multi-Label Learning with Millions of Categories. Retrieved from http://nips.cc/Conferences/2013/Program/event.php?ID=3707.

N. Oza, J. P. Castle, and J. Stutz. 2009. Classification of aeronautics system health and safety documents. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 39, 6 (2009), 670–680.

François Pachet and Pierre Roy. 2009. Improving multilabel analysis of music titles: A large-scale validation of the correction approach. *IEEE Transactions on Audio, Speech, and Language Processing* 17, 2 (2009), 335–343.

Sang-Hyeun Park and Johannes Fürnkranz. 2008. *Multi-Label Classification with Label Constraints*. Technical Report. Knowledge Engineering Group, TU Darmstadt.

R. S. Parpinelli, H. S. Lopes, and A. A Freitas. 2002. Data mining with an ant colony optimization algorithm. *IEEE Transactions on Evolutionary Computation* 6, 4 (2002), 321–332.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

J. P. Pestian, C. Brew, P. M. Matykiewicz, D. J. Hovermale, N. Johnson, K. B. Cohen, and W. Duch. 2007. A shared task involving multi-label classification of clinical free text. In *Proceedings of ACL BioNLP*. 97–104.

S. Peters, L. Denoyer, and P. Gallinari. 2010. Iterative annotation of multi-relational social networks. In *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM'10)*. 96–103.

Guo J. Qi, Xian S. Hua, Yong Rui, Jinhui Tang, Tao Mei, and Hong J. Zhang. 2007. Correlative multi-label video annotation. In *Proceedings of the 15th International Conference on Multimedia (MULTIMEDIA'07)*. ACM, New York, NY, 17–26.

Guo-Jun Qi, Xian-Sheng Hua, Yong Rui, Jinhui Tang, and Hong-Jiang Zhang. 2009. Two-dimensional multilabel active learning with an efficient online adaptation model for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31, 10 (2009), 1880–1897.

J. Ross Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Rafal Rak, Lukasz Kurgan, and Marek Reformat. 2008. A tree-projection-based algorithm for multi-label recurrent-item associative-classification rule generation. *Data & Knowledge Engineering* 64, 1 (2008), 171–197.

Jesse Read. 2008. A pruned problem transformation method for multi-label classification. In *Proceedings of the NZ Computer Science Research Student Conference*.

Jesse Read. 2010. *Scalable Multi-label Classification*. PhD Dissertation. University of Waikato.

Jesse Read. 2011. Advances in Multi-label Classification. Retrieved from http://users.ics.aalto.fi/jesse/talks/Charla-Malaga.pdf.

Jesse Read. 2012. MEKA: A Multi-label Extension to WEKA. Retrieved from http://meka.sourceforge.net/.

Jesse Read, Albert Bifet, Geoffrey Holmes, and Bernhard Pfahringer. 2010. *Efficient Multi-label Classification for Evolving Data Streams*. Technical Report. University of Waikato, Department of Computer Science.

Jesse Read, Albert Bifet, Geoff Holmes, and Bernhard Pfahringer. 2012. Scalable and efficient multi-label classification for evolving data streams. *Machine Learning* 88, 1 (2012), 243–272.

Jesse Read, Bernhard Pfahringer, and Geoff Holmes. 2008. Multi-label classification using ensembles of pruned sets. In *Proceedings of the 2008 8th IEEE International Conference on Data Mining (ICDM'08)*. IEEE Computer Society, Washington, DC, 995–1000.

Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. 2011. Classifier chains for multi-label classification. *Machine Learning* 85, 3 (2011), 1–27.

Lior Rokach, Alon Schclar, and Ehud Itach. 2014. Ensemble methods for multi-label classification. *Expert Systems Applications* 41, 16 (Nov. 2014), 7507–7523.

Timothy Rubin, America Chambers, Padhraic Smyth, and Mark Steyvers. 2012. Statistical topic models for multi-label document classification. *Machine Learning* 88, 1 (2012), 157–208.

Robert E. Schapire and Yoram Singer. 1999. Improved boosting algorithms using confidence-rated predictions. *Machine Learning* 37(3) (1999), 297–336.

Robert E. Schapire and Yoram Singer. 2000. BoosTexter: A boosting-based system for text categorization. *Machine Learning* 39, 2/3 (2000), 135–168.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM Computing Survey* 34, 1 (March 2002), 1–47.

Fabrizio Sebastiani, Alessandro Sperduti, and Nicola Valdambrini. 2000. An improved boosting algorithm and its application to text categorization. In *Proceedings of the 9th International Conference on Information and Knowledge Management (CIKM'00)*. ACM, New York, NY, 78–85.

Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis P. Vlahavas. 2011. On the stratification of multi-label data. In *Proceedings of the Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD (part III) (Lecture Notes in Computer Science)*, Vol. 6913. 145–158.

Huan Shao, GuoZheng Li, GuoPing Liu, and YiQin Wang. 2010. Symptom selection for multi-label data of inquiry diagnosis in traditional Chinese medicine. *Science China Information Sciences* 1 (2010), 1–13.

Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. 2009. TextonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision* 81 (2009), 2–23.

Andrew Skabar, Dennis Wollersheim, and Tim Whitfort. 2006. Multi-label classification of gene function using MLPs. In *Proceedings of the International Joint Conference on Neural Networks*. 2234–2240.

Cees G. M. Snoek, Marcel Worring, Jan C. van Gemert, Jan-Mark Geusebroek, and Arnold W. M. Smeulders. 2006. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proceedings of ACM Multimedia*. 421–430.

Tal Sobol-Shikler and Peter Robinson. 2010. Classification of complex information: Inference of co-occurring affective states from their expressions in speech. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 7 (2010), 1284–1297.

Yang Song, Lu Zhang, and C. Lee Giles. 2011. Automatic tag recommendation algorithms for social recommender systems. *ACM Transactions on the Web* 5, 1 (2011), 4:1–4:31.

Stephan Spat, Bruno Cadonna, Ivo Rakovac, Christian Gütl, Hubert Leitner, Günther Stark, and Peter Beck. 2008. Enhanced information retrieval from narrative German-language clinical text documents using automated document classification. In *Proceedings of MIE 2008 the 21st International Congress of the European Federation for Medical Informatics*. 473–478.

Eleftherios Spyromitros, Grigorios Tsoumakas, and Ioannis Vlahavas. 2008. An empirical study of lazy multilabel classification algorithms. In *Proceedings of the 5th Hellenic Conference on Artificial Intelligence (SETN'08)*. 401–406.

Ashok Srivastava and Brett Zane-Ulman. 2005. Discovering recurring anomalies in text reports regarding complex space systems. In *Proceedings of the 2005 IEEE Aerospace Conference*. 3853–3862.

Liang Sun, Shuiwang Ji, and Jieping Ye. 2013. *Multi-Label Dimensionality Reduction*. Chapman & Hall/CRC Machine Learning & Pattern Recognition.

Muhammad A. Tahir, Josef Kittler, Fei Yan, and Krystian Mikolajczyk. 2009. Kernel discriminant analysis using triangular kernel for semantic scene classification. In *Proceedings of the 7th International Workshop on Content-Based Multimedia Indexing (CBMI'09)*. IEEE, Los Alamitos, CA, 1–6.

Farbound Tai and Hsuan-Tien Lin. 2010. Multi-label classification with principal label space transformation. In *Proceedings of the 2nd International Workshop on Learning from Multi-Label Data (MLD'10)*. 45–52.

Lei Tang and Huan Liu. 2009. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09)*. ACM, New York, NY, 817–826.

Lei Tang, Suju Rajan, and Vijay K. Narayanan. 2009. Large scale multi-label classification via metalabeler. In *Proceedings of the 18th International Conference on World Wide Web*. New York, NY, 211–220.

Lena Tenenboim, Lior Rokach, and Bracha Shapira. 2010. Identification of label dependencies for multi-label classification. In *Proceedings of the 2nd International Workshop on Learning from Multi-Label Data (MLD'10)*. 53–60.

Fadi A. Thabtah, Peter Cowling, and Yonghong Peng. 2004. MMAC: A new multi-class, multi-label associative classification approach. In *Proceedings of the 4th IEEE International Conference on Data Mining (ICDM'04)*. 217–224.

Fadi A. Thabtah and Peter L. Cowling. 2007. A greedy classification algorithm based on association rule. *Applied Soft Computing* 7, 3 (2007), 1102–1111.

K. Trohidis, Grigorios Tsoumakas, G. Kalliris, and Ioannis Vlahavas. 2008. Multi-label classification of music into emotions. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR'08)*.

Grigorios Tsoumakas, A. Dimou, E. Spyromitros, V. Mezaris, I. Kompatsiaris, and Ioannis Vlahavas. 2009. Correlation-based pruning of stacked binary relevance models for multi- label learning. In *Proceedings of the 1st International Workshop on Learning from Multi-Label Data (MLD'09)*. 101–116.

Grigorios Tsoumakas and Ioannis Katakis. 2007. Multi label classification: An overview. *International Journal of Data Warehousing and Mining* 3, 3 (2007), 1–13.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. 2008. Effective and efficient multilabel classification in domains with large number of labels. In *Proceedings of the ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. 2010a. *Data Mining and Knowledge Discovery Handbook, Part 6*. Springer, Chapter Mining Multi-label Data, 667–685.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. 2010b. Random k-labelsets for multi-label classification. *IEEE Transactions on Knowledge and Data Engineering* 23, 7 (2010), 1079–1089.

Grigorios Tsoumakas, Eleftherios Spyromitros-Xioufis, Jozef Vilcek, and Ioannis Vlahavas. 2011. Mulan: A java library for multi-label learning. *Journal of Machine Learning Research* 12 (2011), 2411–2414.

Grigorios Tsoumakas and Ioannis Vlahavas. 2007. Random k-labelsets: An ensemble method for multilabel classification. In *Proceedings of the 18th European Conference on Machine Learning (ECML'07)*, Vol. 4701. 406–417.

Grigorios Tsoumakas, Min Ling Zhang, and Zhi-Hua Zhou. 2009. Learning from multi-label data. *ECML/PKDD'09*. (September 2009).

D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. 2008. Semantic annotation and retrieval of music and sound effects. *IEEE Transactions on Audio, Speech, and Language Processing* 16, 2 (2008), 467–476.

Naonori Ueda and Kazumi Saito. 2002a. Parametric mixture models for multi-labeled text. In *Neural Information Processing Systems 15 (NIPS)*. 721–728.

Naonori Ueda and Kazumi Saito. 2002b. Yahoo Dataset. Retrieved from http://www.kecl.ntt.co.jp/as/members/ueda/yahoo.tar.gz.

Eranga Ukwatta and Jagath Samarabandu. 2009. Vision based metal spectral analysis using multi-label classification. In *Proceedings of the Canadian Conference on Computer and Robot Vision (CRV'09)*. 132–139.

Celine Vens, Jan Struyf, Leander Schietgat, Sašo Džeroski, and Hendrik Blockeel. 2008. Decision trees for hierarchical multi-label classification. *Machine Learning* 73, 2 (2008), 185–214.

Sergeja Vogrincic and Zoran Bosnic. 2011. Ontology-based multi-label classification of economic articles. *Computer Science and Information Systems* 8, 1 (2011), 101–119.

Jingdong Wang, Yinghai Zhao, Xiuqing Wu, and Xian-Sheng Hua. 2010. A transductive multi-label learning approach for video concept detection. *Pattern Recognition* 44 (2010), 2274–2286.

Mei Wang, Xiangdong Zhou, and Tat S. Chua. 2008. Automatic image annotation via local multi-label classification. In *Proceedings of the 2008 International Conference on Content-based Image and Video Retrieval (CIVR'08)*. ACM, New York, NY, 17–26.

Wei Wang and Zhi-Hua Zhou. 2008. On multi-view active learning and the combination with semi-supervised learning. In *Proceedings of the 25th International Conference on Machine Learning (ICML'08)*. USA, 1152–1159.

David H. Wolpert. 1992. Stacked generalization. *Neural Networks* 5 (1992), 241–259.

Jianhua Xu. 2012. An efficient multi-label support vector machine with a zero label. *Expert Systems with Applications* 39, 5 (2012), 4796–4804.

Jianhua Xu. 2013a. Fast multi-label core vector machine. *Pattern Recognition* 46, 3 (2013), 885–898.

Jianhua Xu. 2013b. Laboratory of Intelligent Computation. Retrieved from http://computer.njnu.edu.cn/Lab/LABIC/LABIC_Software.html.

Rong Yan, Jelena Tesic, and John R. Smith. 2007. Model-shared subspace boosting for multi-label classification. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 834–843.

Yan Yan, Glenn Fung, Jennifer G. Dy, and Romer Rosales. 2010. Medical coding classification by leveraging inter-code relationships. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'10)*. ACM, New York, NY, 193–202.

Yiming Yang. 1999. An evaluation of statistical approaches to text categorization. *Information Retrieval* 1, 1–2 (1999), 69–90.

Yiming Yang. 2001. A study of thresholding strategies for text categorization. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01)*. ACM, New York, NY, 137–145.

Yiming Yang and Siddharth Gopal. 2012. Multilabel classification with meta-level features in a learning-to-rank framework. *Machine Learning* 88, 1 (2012), 47–68.

Yiming Yang and Xin Liu. 1999. A re-examination of text categorization methods. In *Proceedings of the 22nd Annual International SIGIR*. 42–49.

Yang Yang and Bao-Liang Lu. 2006. Prediction of protein subcellular multi-locations with a min-max modular support vector machine. In *Advances in Neural Networks (ISNN'06)*. Lecture Notes in Computer Science, Vol. 3973. Springer, Berlin, 667–673.

Yiming Yang and Jan O. Pedersen. 1997. A comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning (ICML'97)*. Morgan Kaufmann, San Francisco, CA, 412–420.

John Yearwood, Musa Mammadov, and Arunava Banerjee. 2010. Profiling phishing emails based on hyperlink information. In *Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining*. IEEE, 120–127.

Kai Yu, Shipeng Yu, and Volker Tresp. 2005. Multi-label informed latent semantic indexing. In *SIGIR'05: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'05)*. ACM, New York, NY, 258–265.

Jintao Zhang and Jun Huan. 2012. Inductive multi-task learning with multiple view data. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 543–551.

Min-Ling Zhang. 2009. Ml-rbf: RBF neural networks for multi-label learning. *Neural Processing Letters* 29, 2 (2009), 61–74.

Min-Ling Zhang, José M. Peña, and Victor Robles. 2009. Feature selection for multi-label naive Bayes classification. *Information Sciences* 179, 19 (2009), 3218–3229.

Min L. Zhang and Kun Zhang. 2010. Multi-label learning by exploiting label dependency. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'10)*. ACM, New York, NY, 999–1008.

Min-Ling Zhang and Zhi-Hua Zhou. 2005. A k-nearest neighbor based algorithm for multi-label classification. In *Proceedings of the IEEE International Conference on Granular Computing (GrC'05)*. 718–721.

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

Min L. Zhang and Zhi H. Zhou. 2007. Multi-label learning by instance differentiation. In *AAAI*. 669–674.

Min-Ling Zhang and Zhi-Hua Zhou. 2014. A review on multi-label learning algorithms. *IEEE Transaction on Knowledge and Data Engineering* 26, 8 (2014), 1819–1837.

Xiatian Zhang, Quan Yuan, Shiwan Zhao, Wei Fan, Wentao Zheng, and Zhong Wang. 2010. Multi-label classification without the multi-label cost. In *Proceedings of the 10th SIAM International Conference on Data Mining*.

Yi Zhang. 2012. *Learning with Limited Supervision by Input and Output Coding*. PhD Dissertation. Carnegie Mellon University.

Yi Zhang, Samuel Burer, and W. Nick Street. 2006. Ensemble pruning via semi-definite programming. *Journal of Machine Learning Research* 7 (2006), 1315–1338.

Yin Zhang and Zhi H. Zhou. 2010. Multilabel dimensionality reduction via dependence maximization. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 4, 3 (2010), 14:1–14:21.

Tianyi Zhou, Dacheng Tao, and Xindong Wu. 2012a. Compressed labeling on distilled labelsets for multi-label learning. *Machine Learning* 88, 1–2 (2012), 69–126.

Zhi H. Zhou and Min L. Zhang. 2006. Multi-instance multi-label learning with application to scene classification. In *Advances in Neural Information Processing Systems 19 (NIPS'06)*, Bernhard Schölkopf, John C. Platt, and Thomas Hoffman (Eds.). 1609–1616.

Zhi-Hua Zhou, Min-Ling Zhang, Sheng-Jun Huang, and Yu-Feng Li. 2012b. Multi-instance multi-label learning. *Artificial Intelligence* 176, 1 (2012), 2291–2320.

Shenghuo Zhu, Xiang Ji, Wei Xu, and Yihong Gong. 2005. Multi-labelled classification using maximum entropy method. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'05)*. ACM, New York, NY, 274–281.

Received October 2013; revised July 2014; accepted January 2015