Overview

Multi-label learning: a review of the state of the art and ongoing research

Eva Gibaja[1] and Sebastián Ventura[1,2]*

Multi-label learning is quite a recent supervised learning paradigm. Owing to its capabilities to improve performance in problems where a pattern may have more than one associated class, it has attracted the attention of researchers, producing an increasing number of publications. This study presents an up-to-date overview about multi-label learning with the aim of sorting and describing the main approaches developed till now. The formal definition of the paradigm, the analysis of its impact on the literature, its main applications, works developed, pitfalls and guidelines, and ongoing research are presented. © 2014 John Wiley & Sons, Ltd.

How to cite this article:
WIREs Data Mining Knowl Discov 2014, 4:411–444. doi: 10.1002/widm.1139

INTRODUCTION

Multi-label learning (MLL) is a supervised learning paradigm that has attracted a great deal of attention in recent years due to its capabilities of improving performance in many current applications such as the classification of multimedia, the prediction of gene and protein functions, the direct marketing, or the social network mining. All of these applications have in common that not one, but multiple outputs are required. Therefore, the *only-one-label-per-pattern* restriction of classical supervised learning (also known as single-label learning) is not satisfied. As well as dealing with multiple outputs, MLL has to deal with trending challenges such as relationships between labels, the computational costs of generating the models, presence of imbalanced labels, or high dimensionality of data. The key challenge has been recently identified as dealing with the high dimensionality of the output space, especially in domains with a large number of labels.[1] This challenge involves exploring label correlations efficiently. Besides, specialized workshops,[2–4] special issues,[5] repositories of benchmark data sets, and software[6–9] have contributed to the progress in this field. All of these factors have become MLL into a relevant supervised learning paradigm with an increasing number of papers published on it per year.

First tutorials about MLL were published in its earlier stages.[10,11] They introduced the setting of MLL and compiled the contributions and methods developed up to 2007. Next, in Ref 12, a review that has become a reference for the MLL community was published. It included the main proposals developed up to 2008 and also the description of the main evaluation metrics. It is also worth citing two other recent papers. First, in Ref 13 an experimental comparison of 12 well-known MLL methods was carried out using 16 evaluation measures on 11 benchmark data sets. Its aim was to provide a better understanding of the performance of these methods. Second, Zhang and Zhou[1] have recently published a paper, whose main aim is to describe the setting, evaluation metrics, and eight representative MLL algorithms in an elaborated and formal way. Due to the high number of proposals developed the latest years (around 700 new works only from 2009 to 2012), the aim of this study is filling this gap with an update for the topic.

The rest of this study is organized as follows: first, the formal definition of MLL and the main fields of application are described (*Multi-Label Learning* section) followed by the analysis of its impact on the literature (*MLL in the Literature* section). Next the state of art (*MLL Methods* section) and ongoing

*Correspondence to: sventura@uco.es

[1]Department of Computer Sciences and Numerical Analysis, University of Córdoba, Córdoba, Spain

[2]King Abdulaziz University, Jeddah, Saudi Arabia

Conflict of interest: The authors have declared no conflicts of interest for this article.

research (*Ongoing Research* section) are analyzed. *Pitfalls and Guidelines* section summarizes a series of pitfalls and guidelines mainly focused on the selection of a proper multi-label learner. This study finishes with the set of more relevant conclusions.

MULTI-LABEL LEARNING

This section presents the main fields of application of MLL, a formal definition of the paradigm, a summary of evaluation metrics and the description of other learning settings that may share some features with MLL.

Key Applications of MLL

- *Text categorization* consists of assigning a set of predefined categories to documents. As a document can belong simultaneously to more than one category it can be tackled with MLL. It has been applied to many kinds of documents such as legal texts,[14,15] web documents,[16,17] news,[18] research papers,[19] narrative clinical text,[20] patents,[21] or aeronautics reports.[22] Other related applications are document indexing,[23] tag suggestion,[24,25] e-mail filtering,[26] medical coding,[27] query categorization,[28] or the classification of news sentences into multiple emotion categories.[29,30]
- *Multimedia*. MLL techniques have been applied to many types of resources such as images, videos, and sound. Examples of applications are: automatic image annotation,[31,32] face verification,[33] video annotation,[34] object recognition,[35] detection of emotions into music,[36,37] music metadata extraction,[38] and speech emotion classification.[39]
- *Biology*. It is worth noting gene function prediction,[40–46] and protein function prediction,[47–50] applications. Other recent applications are the prediction of proteins' 3D structures[51] and, finally, the problem of protein subcellular multi-location[52] (proteins may simultaneously exist at, or move between, two or more different subcellular locations).
- *Chemical data analysis*. MLL has also been applied to predict adverse drug reactions,[53] to identify the drugs that have two or more different biological actions (drug discovery)[54] and to detect contaminants in machine lubricants by using spectral images (vision-based metal spectral analysis).[55]
- *Social network mining* has become a new area of interest. Collective behavior learning consists of inferring behavior or preferences of individuals.[56,57] Social networking advertising[58] or the automatic annotation of the nodes of a partially labeled multi-relational graph[59] are other fields of application.
- *E-Learning*. MLL has been also applied to classify learning styles based on learners' profiles[60] and to tag learning objects.[61]
- *Other applications*. Other fields of application worth citing are direct marketing, where potential buyers of certain products are identified,[62] and medical diagnosis[63] (many symptoms may be associated with more than one syndrome). Finally, in Ref 64, MLL was applied to classify dermoscopy images of skin lesions which could contain several pattern lesions.

Formal Definition

According to Refs 18 and 65, given $\mathcal{X} = X_1 \times \ldots \times X_d$ a d-dimensional input space of numerical or categorical features, and an output space of q labels, $\mathcal{Y} = \{\lambda_1, \lambda_2, \ldots, \lambda_q\}$, a multi-label pattern can be defined as a pair $(\mathbf{x}, Y)$, where $\mathbf{x} = (x_1, \ldots, x_d) \in \mathcal{X}$ and $Y \subseteq \mathcal{Y}$ is called *label set*. 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. Three different tasks can be included in MLL[12]:

- *Label Ranking* (LR) consists of producing a function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ that induces an ordering of all the possible labels which express the relevance of labels to a given instance $\mathbf{x}$. Thus, label λ_1 is considered to be ranked higher than λ_2 if $f(\mathbf{x}, \lambda_1) > f(\mathbf{x}, \lambda_2)$. For each instance, $\mathbf{x} \in \mathcal{X}$, a rank function, $\tau_{\mathbf{x}} : \mathcal{Y} \to \{1, 2, \ldots, q\}$, can be defined using the output real value of the classifier f, such that 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 value, the better the position in the ranking is.
- *Multi-Label Classification* (MLC) consists of defining a function $h_{\mathrm{MLC}} : \mathcal{X} \to 2^{\mathcal{Y}}$ that returns the set of relevant labels. So for each $\mathbf{x} \in \mathcal{X}$, we have a bipartition $\left(Y, \overline{Y}\right)$ of the label set $\mathcal{Y}$, where $Y = h_{\mathrm{MLC}}(x)$ is the set of relevant labels and $\overline{Y}$ is the set of irrelevant ones. Here, $\overline{Y} = \mathcal{Y} \backslash Y$ denotes the set theoretic complement of Y in $\mathcal{Y}$. Multi-class (MCC) and binary classification (BC) can be seen as a particular case of MLC where $h_{\mathrm{MCC}} : \mathcal{X} \to \mathcal{Y}$ and

$h_{BC} : \mathcal{X} \rightarrow \{0, 1\}$. A multi-label classifier can be derived from a ranking model by using a threshold function. Strategies for thresholding can be found in Refs 28, 44, and 66–69.

- *Multi-Label Ranking* (MLR) is a generalization of MLC and LR consisting of producing, 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, then, in a consistent ranking, labels in Y will have higher rank than labels in $\overline{Y}$.

According to Ref 1, MLL can be considered a particular setting into a wider framework, called *multi-target learning*, where a pattern is associated with multiple outputs. Depending on the kind of outputs different instantiations are considered, calling the settings: (1) MLL when output variables are binary, (2) *multi-dimensional learning*[70] if output variables are multi-class, or (3) *multi-output regression*[71] for numerical outputs. Besides, combination of different types of output variables can be considered.

Metrics to Evaluate the Models

Tsoumakas et al.[12] distinguish two kinds of metrics to evaluate MLL methods: *label-based* metrics and *example-based* metrics. The idea of the label-based approach is computing 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*) and then obtaining an average value. Given B any binary evaluation measure, two different averaging approaches can be used: the *macro approach* computes one metric for each label and then the values are averaged over all the categories, while the *micro approach* considers predictions of all instances together (aggregating the *tp*, *tn*, *fp*, and *fn* values of all classes) and then calculates the measure across all labels.

Example-based metrics are calculated for each test example and then averaged across the test set. They are categorized commonly into two groups: metrics to evaluate rankings and metrics to evaluate bipartitions. Table 1 summarizes the main metrics described in literature according to the categorization described. Besides, metrics to evaluate confidence scores and hierarchies of labels have been included.

Let $T = \{(\mathbf{x}_i, Y_i) | 1 \leq i \leq t\}$ be a multi-label test set with t instances. Given an instance, $\mathbf{x}$, let Y and Z be the set of true and predicted labels, and let $\mathbf{w} = \left(w_{\lambda_1}, w_{\lambda_2}, \ldots, w_{\lambda_q}\right)$ be a vector with normalized output confidence scores in $[0, 1]$. For any predicate, π, $[\![\pi]\!]$ returns 1 if the predicate is true and 0 otherwise. Let τ^* be the true ranking, Δ stands for the symmetric difference of two sets, arg function returns a label, and given a hierarchy of labels, $anc(i)$ returns the set of ancestors of a node i.

Other Related Learning Settings

This section discusses other learning settings, which may share some features with MLL and are worth to be briefly discussed.

- *Multi-task learning or learning parallel tasks* (MTL)[80] tries learning in parallel several tasks that share a common representation. Thus, what is learned for one task can help the others to be learnt better. According to Zhang and Zhou,[1] there are three main differences between multi-label and multi-task learning. First, in MLL, all the examples have the same feature space while in multi-task learning it can be the same or different. The purpose is also different, thus in MLL one task is learned (i.e., predicting the label set associated with an object) while in multi-task learning, multiple tasks are learned simultaneously. Finally, the label space in MLL is large while in multi-task learning is not reasonable to consider a large number of tasks.
- *Multiple-labels learning or partial labeling.*[81] In this kind of problem, each pattern has multiple candidate labels, and only one of them is the correct one. Real problems such as disagreement between assessors can be viewed from this point of view.
- *Multi-instance learning* (MIL)[82] consists of learning a concept where training labels are associated with sets (*bags*) of patterns (*instances*) rather than with individual patterns. A bag is positive if, at least, one of if patterns is positive and negative otherwise. Numerous real-world tasks (e.g., drug activity prediction or web index page recommendation) can be naturally represented as multiple instance problems. *Multi-Instance MLL* section describes the multi-instance multi-label (MIML) setting, in which a bag may have associated not one, but a set of labels.
- Reverse *MLL* (RMLL)[83] consists of carrying out a reverse prediction, i.e., to predict sets of relevant instances given a set of labels.
- *Preferential text classification*[84] is a problem where primary (central topics of the document) and secondary categories associated with a document can be distinguished. Misclassifications related to primary categories should be penalized

TABLE 1 | Taxonomy of Metrics to Evaluate MLL Algorithms

Label-based

12	Macro approach	$B_{macro} = \frac{1}{q}\sum_{i=1}^{q} B\left(tp_i, fp_i, tn_i, fn_i\right)$, e.g., $recall_{\text{macro}} = \frac{1}{q}\sum_{i=1}^{q}\frac{tp_i}{tp_i+fn_i}$
12	Micro approach	$B_{micro} = B\left(\sum_{i=1}^{q} tp_i, \sum_{i=1}^{q} fp_i, \sum_{i=1}^{q} tn_i, \sum_{i=1}^{q} fn_i\right)$, e.g., $recall_{\text{micro}} = \frac{\sum_{i=1}^{q} tp_i}{\sum_{i=1}^{q} tp_i + \sum_{i=1}^{q} fn_i}$

Example-based

Metrics to evaluate bipartitions

72	↑ Subset accuracy	$= \frac{1}{t}\sum_{i=1}^{t} [\![Z_i = Y_i]\!]$
73	↓ Subset 0/1 loss	$= \frac{1}{t}\sum_{i=1}^{t} [\![Z_i \neq Y_i]\!]$
74	↓ Hamming loss	$= \frac{1}{t}\sum_{i=1}^{t}\frac{1}{q}\left\|Z_i \Delta Y_i\right\|$
75	↑ Recall	$= \frac{1}{t}\sum_{i=1}^{t}\frac{\left\|Z_i \cap Y_i\right\|}{\left\|Y_i\right\|}$
75	↑ Precision	$= \frac{1}{t}\sum_{i=1}^{t}\frac{\left\|Z_i \cap Y_i\right\|}{\left\|Z_i\right\|}$
75	↑ Accuracy	$= \frac{1}{t}\sum_{i=1}^{t}\frac{\left\|Z_i \cap Y_i\right\|}{\left\|Z_i \cup Y_i\right\|}$
75	↑ F1-Score	$= \frac{1}{t}\sum_{i=1}^{t}\frac{2\left\|Z_i \cap Y_i\right\|}{\left\|Z_i\right\| + \left\|Y_i\right\|}$

Metrics to evaluate rankings

18	↓ One-error	$= \frac{1}{t}\sum_{i=1}^{t} [\![\arg\min_{\lambda \in \mathcal{Y}} \tau_i(\lambda) \notin Y_i]\!]$
18	↓ Coverage	$= \frac{1}{t}\sum_{i=1}^{t} \max_{\lambda \in Y_i} \tau_i(\lambda) - 1$
74	↓ Ranking loss	$= \frac{1}{t}\sum_{i=1}^{t}\frac{1}{\left\|Y_i\right\|\left\|\overline{Y_i}\right\|}\left\|E\right\|$ where $E = \left\{\left(\lambda, \lambda'\right) \middle\| \tau_i(\lambda) > \tau_i\left(\lambda'\right), \left(\lambda, \lambda'\right) \in Y_i \times \overline{Y_i}\right\}$
76,15	↓ IsError	$= \frac{1}{t}\sum_{i=1}^{t} [\![\sum_{\lambda \in \mathcal{Y}} \left\|\tau_i^*(\lambda) - \tau_i(\lambda)\right\| \neq 0]\!]$
18	↑ Average precision	$= \frac{1}{t}\sum_{i=1}^{t}\frac{1}{\left\|Y_i\right\|}\sum_{\lambda \in Y_i}\frac{\left\|\left\{\lambda' \in Y_i \middle\| \tau_i(\lambda') \leq \tau_i(\lambda)\right\}\right\|}{\tau_i(\lambda)}$
15	↓ Margin loss	$= \frac{1}{t}\sum_{i=1}^{t}\max\left(0, \max\left\{\tau(\lambda) \middle\| \lambda \in Y_i\right\} - \min\left\{\tau\left(\lambda'\right) \middle\| \lambda' \notin Y_i\right\}\right)$
77	↓ Ranking error	$= \frac{1}{t}\sum_{i=1}^{t}\sum_{\lambda \in \mathcal{Y}}\left\|\tau_i^*(\lambda) - \tau_i(\lambda)\right\|^2$

Metrics to evaluate confidence scores

78	↓ Log loss	$= \frac{1}{tq}\sum_{i=1}^{t}\sum_{\lambda \in \mathcal{Y}}\min\left(-LogLoss\left(\lambda, \mathbf{w_i}\right), \ln(t)\right)$ where $LogLoss(\lambda, \mathbf{w}) = \ln\left(w_\lambda\right)$ *if* $\lambda \in Y \ln\left(1 - w_\lambda\right)$ *if* $\lambda \in \overline{Y}$

Metrics for HMC

79	↓ 0/1 loss	$= \frac{1}{t}\sum_{i=1}^{t} [\![Z_i \neq Y_i]\!]$
79	↓ Symmetric diff.	$= \frac{1}{t}\sum_{i=1}^{t}\left\|Z_i \Delta Y_i\right\|$
79	↓ Hierarchical loss	$= \frac{1}{t}\sum_{i=1}^{t}\sum_{j=1}^{q}\left\|\left\{Z_{ij} \neq Y_{ij} \wedge Z_{ik} = Y_{ik}, k \in anc(j)\right\}\right\|$

↑, means the metric has to be maximized; ↓, means the metric has to be minimized.

more severely than those related to secondary ones.

- *Weak-label problem or learning with incomplete class assignment*.[85,86] In this kind of problem, only a partial labeling associated with each training example is provided. In other words, if a label has been assigned to an instance, it is a proper label of the instance but, if a label has not been assigned it cannot be concluded that it is not a proper label.
- *Multi-valued multi-label* (M^2).[87] In this kind of problem, samples are not only associated with a set of labels, they may also have some attributes of the pattern presenting several values.
- *Graded MLC* (GMLC).[43] The aim is to obtain not only a set of labels, but also a membership value for each label in the sense of fuzzy set theory.
- *Multi-view learning*.[88] One object has different representations in the form of several disjoint subsets of features (each subset is a view), each of which is sufficient for learning the target concept. Thus, several classifiers are trained on subsets of features or views. This setting has been combined with active learning in order to reduce the annotation effort in multi-label tasks (see *Semi-Supervised and Active Learning* section).

MLL IN THE LITERATURE

In this section, the visibility of research in MLL is going to be studied by analyzing the number of publications and citations at the two scientific citation databases more broadly used: Thompson ISI and Elsevier Scopus. Table 2 shows the queries made and the overall results. In the case of ISI, the search has been carried out by title and topic, while in Scopus the search has been carried out by title, abstract, and keywords. The search was restricted to articles and conference papers.

The *h-index* reflects both the number of publications and the number of citations per publication. The h-index for the 1116 papers considered in Scopus is 39, which means that 39 of these papers have been cited at least 39 times. In ISI, the h-index of the 638 retrieved references is 28. Two graphics summarizing the total of articles and citations both in ISI and Scopus databases are shown in Figures 1 and 2. It is observed that MLL is a quite recent topic and also the exponential increase of papers and citations in the latest years. Finally, the 10 most cited papers in ISI and Scopus databases with their respective number of citations are listed in Tables 3 and 4.

MLL METHODS

This section is going to follow the taxonomy of MLL methods presented in Ref 12 that distinguishes between: *problem transformation methods* and *algorithm adaptation methods*. The former are algorithm independent and transform the multi-label problem into one or more single-label ones in which is then applied a single-label classification algorithm, whilst the latter extend a single-label algorithm in order to directly deal with multi-label data. A summary of the main problem transformation methods is found in Table 5. Table 6 summarizes the main algorithm adaptation methods developed according to the adapted paradigm.

Problem Transformation Methods

Ranking Via Single-Label Learning

This approach consists of transforming the instances in a multi-label data set in order to obtain a single-label one. Then a single-label classifier, which is able to produce a score for each label (e.g., probability), will allow to obtain a ranking.[12] Three strategies can be followed: ignoring all multi-label instances (*ignore* method), transforming every multi-label instance into several ones, one per label (*copy* and *copy-weight* methods) or selecting one of the labels of the multiple-labeled patterns (*select-max*, *select-min*, or *select-random* methods). Despite being simple, these methods produce problems of information loss in terms of labels or label relationships and so they may not be very useful.

Binary Methods

The *Binary Relevance* method (BR)[12] follows the *one-versus-all* (OVA) philosophy and builds one binary data set for each label. Patterns predicting the label are considered positive patterns and the rest are considered to be negative. Once an unknown pattern is presented to the model, the output will be the set of positive classes predicted. The main problem of BR is the assumption of label independence that ignores the relationships between labels[164] and may lead to failure to predicting label combinations or rankings of labels.[100] Nevertheless, it is computationally simple, scales linearly with the number of labels and can be parallelized.[97]

Some approaches have been developed in order to overcome the label independence assumption of BR while maintaining a reasonable complexity. They are described below.

The *Classifier Chains* (CC) model[97] generates q binary classifiers, but they are linked in such a way that each classifier incorporates the labels predicted by the

TABLE 2 | Queries Made to ISI and Scopus

Database	ISI	Scopus
Query	TOPIC (('multi-label' OR 'multilabel') AND ('classification' OR 'learning')) OR TITLE: (('multi-label' OR 'multilabel') AND ('classification' OR 'learning')) Refined by: DOCUMENT TYPES: (ARTICLE OR MEETING) Timespan: all years	TITLE-ABS-KEY((((multi-label) OR (multilabel)) AND ((classification) OR (learning)))) AND (LIMIT-TO(DOCTYPE, 'cp') OR LIMIT-TO(DOCTYPE, 'ar'))
Date	July 6, 2014	July 2, 2014
No. documents	638	1116
h-index	28	39

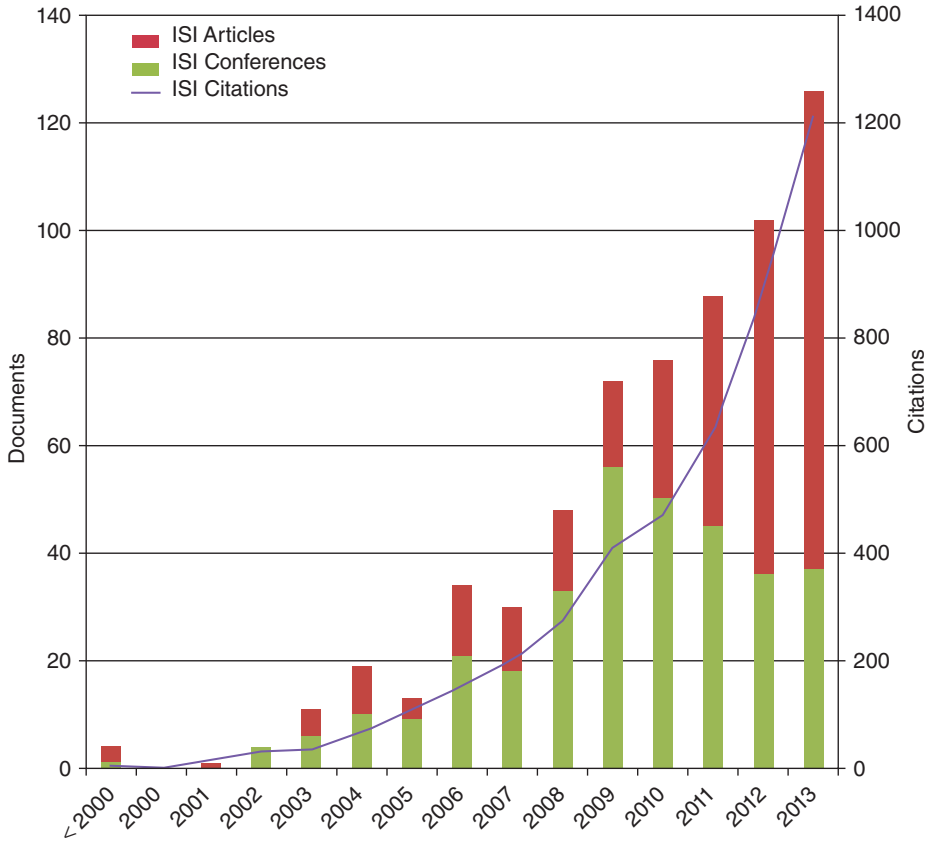

FIGURE 1 | Multi-label Learning in ISI.

previous classifiers in the chain as additional features. Therefore, label correlations are considered in a random manner while the complexity is linear with the number of labels. CC can be parallelized in the training stage. Experiments proved CC overall improved over BR with a similar complexity. As the order of the chain itself can influence the performance, an *Ensemble of CC* (ECC), which trained a set of CC classifiers with a random chain ordering and a random subset of training patterns sampled with replacement, was also proposed. These authors also proposed an *Ensemble of BR classifiers* (EBR) developed identically to ECC but without chaining. ECC and EBR obtained promising predictive results in a wide range of metrics and data sets and demonstrated to be efficient on large data sets without significant losses in predictive performance. In Ref 98, *Probabilistic CC* (PCC), a Bayes optimal way of forming CC that outperformed CC, was described. Its drawback was the computational complexity at prediction time (it must look at each of 2^q possible combinations) being recommendable only with a small to moderate number of labels ($q < 15$).

A different approach is 2BR, also called *Meta-BR* (MBR) in Ref 97, which basically consists

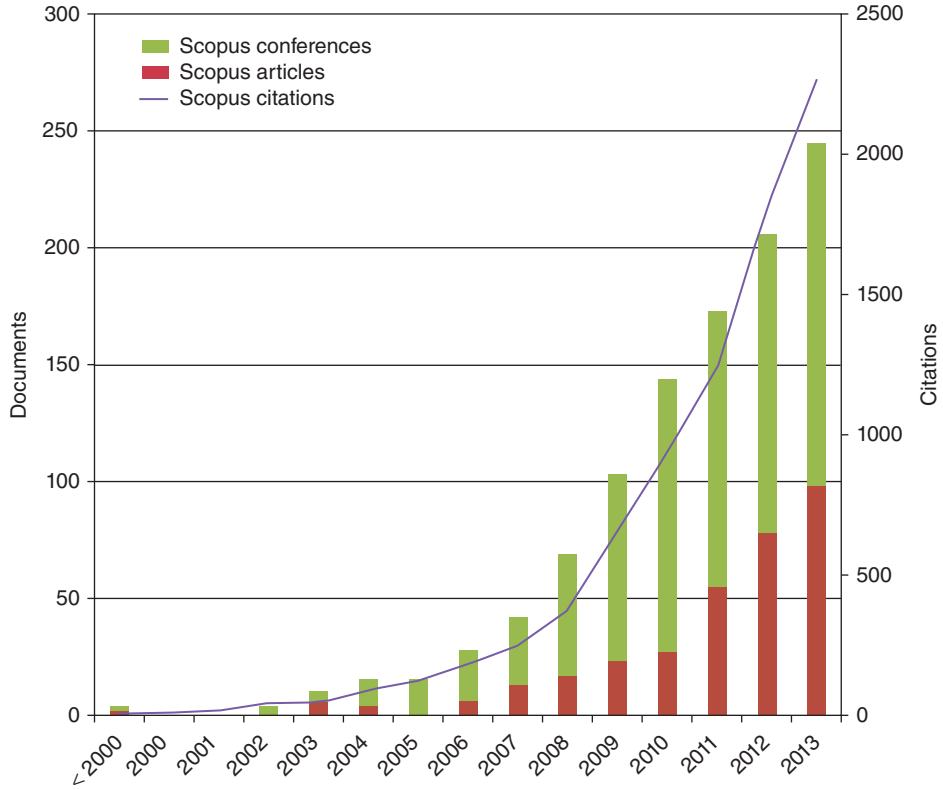

FIGURE 2 | Multi-label Learning in Scopus.

of applying BR twice.[38,99,100] It follows the philosophy of *Stacking*[165] and maintains the linear time complexity with respect to the number of labels in the data set. During the first step (the base-level), a BR classifier is learnt and the second BR step (the meta-level) implements a meta-learning stage. To do this, the input feature set of the meta-level is augmented with q extra features consisting of the predictions of each binary classifier in the base-level. After that, q new binary classifiers are trained with this extended data set considering the desired outputs as targets. For classification, outputs of the first level of classifiers are used to extend the examples and these new examples are classified by the second level. In Ref 100, only the predictions of the base-level models regarding correlated labels (according to the ϕ coefficient) were considered when constructing the meta-level training examples for a certain label. Decision trees and support vector machines (SVM) were used in both base and meta levels and linear regression at meta-learning level. This approach was able to improve the efficiency substantially, without significant loss in predictive performance.

Finally, *BRplus* (BR+)[101] also incremented the feature space of the BR classifiers with labels, but in this case with $q-1$ features corresponding to the other labels in the data set. During the classification stage, features of unlabeled examples were augmented by using a BR predictor trained with the original training data. The reported experiments showed it effectively improved BR and maintained results comparable to CC.

Pairwise Methods

The *Ranking by Pairwise Comparison* (RPC) approach[102] follows a *one-versus-one* (OVO) philosophy and transforms a data set with q labels into $q(q-1)/2$ binary data sets, one per each pair of labels. Each data set uses the examples belonging to one of the two classes as positive or negative examples respectively (patterns belonging to both labels are not considered). Then, a binary classifier is built for each data set. Given an unknown pattern, the prediction is obtained by invoking all models and obtaining a rank from the counting votes for each label. It is also worth mentioning *Calibrated* LR (CLR),[103,104] which adds to the RPC transformation q data sets, λ_i vs λ_0, which are identical to a BR transformation. The key idea of this approach is the virtual label, λ_0, that acts as a split point separating the relevant for the irrelevant labels obtaining a consistent ranking and bipartition. Empirical results in the area of text

TABLE 3 | ISI Top Ten Most Cited Papers

No.	Cit.	Title	Authors	Publication	Year
1	911	Improved boosting algorithms using confidence-rated predictions[74]	Shapire and Singer	Machine Learning Theory	1999
2	378	RCV1: A new benchmark collection for text categorization research[89]	Lewis et al.	Journal of Machine Learning Research	2004
3	259	Learning multi-label scene classification[90]	Boutell et al.	Pattern Recognition	2004
4	228	ML-KNN: A lazy learning approach to multi-label learning[91]	Zhang and Zhou	Pattern Recognition	2007
5	151	iLoc-Euk: A multi-label classifier for predicting the subcellular localization of singleplex and multiplex eukaryotic proteins[52]	Chou et al.	PLoS ONE	2011
6	143	Multi-label neural networks with applications to functional genomics and text categorization[46]	Zhang and Zhou	IEEE Transactions on Knowledge and Data Engineering	2006
7	120	Optimization method based extreme learning machine for classification[92]	Huang et al.	Neurocomputing	2010
8	97	A kernel method for multi-labeled classification[44]	Elisseeff and Weston	NIPS	2001
9	82	iLoc-Virus: a multi-label learning classifier for identifying the subcellular localization of virus proteins with both single and multiple sites[93]	Xiao et al.	Journal of Theoretical Biology	2011
10	81	Decision trees for hierarchical multi-label classification[94]	Vens et al.	Machine Learning	2008

categorization, image classification, and gene analysis showed it outperformed BR, MMP, and MLPP (these last two methods will be described in *Neural Networks* section). Authors also found that CLR is typically a bit more conservative in predicting the number of relevant labels. This translates into lower recall, but precision and F1-score values tend to be high. The main drawbacks of pairwise methods described are the space complexity and the need to query all the generated (q^2) binary models at runtime, which may become impractical for large number of labels. *QWeighted CLR* (QCLR)[105] combined CLR with the *QWeighted*[14] voting schema. Multi-class QWeighted efficiently computes the class with the highest accumulated voting mass (i.e., the top-ranked class) without evaluating all pairwise classifiers. In the multi-label case, the process is iteratively applied until the virtual label is returned, which means that all remaining labels are irrelevant. It speeded up the voting process reducing the evaluations needed from $q(q-1)/2$ to $q\log(q)$ in practice, which is near the q evaluations processed by BR. The remaining bottleneck is the need to store a quadratic number of base classifiers. Finally, *Dual Layer Voting Method* (DLVM)[106] is another voting strategy consisting of a two-stage architecture. In the first layer, BR models output a probability about the relevance of every label. Then, if this probability is above a certain threshold, the pairwise models of the second layer are consulted. DLVM outperformed the CLR's majority voting in terms of speed whilst maintaining the prediction performance.

Label Combination Methods

The *Label Powerset* (LP) approach[12,90] builds a single-label data set where each possible combination of labels is considered as a class itself. Then a multi-class algorithm is applied. Given a new instance, LP outputs a class, which actually is a label set in the original data set. Despite being able to model label correlations, the drawback is its complexity, in the worst case, exponential with the number of labels. *Pruned Problem Transformation* (PPT) or *Pruned Sets* (PS)[107] tries to reduce this complexity focusing on the most important combinations of labels by pruning examples with less frequent label sets; to compensate for such information loss it reintroduces the pruned examples along with subsets of their label sets. After that, LP is applied. Like LP, PS is not able to output label sets that are not in the training set. A

TABLE 4 | Scopus Top Ten Most Cited Papers

No.	Cit.	Title	Authors	Publication	Year
1	1360	Improved boosting algorithms using confidence-rated predictions[74]	Shapire and Singer	Machine Learning	1999
2	415	Learning multi-label scene classification[90]	Boutell et al.	Pattern Recognition	2004
3	381	Multi-label classification: An overview[11]	Tsoumakas and Katakis	International Journal of Data Warehousing and Mining	2007
4	359	ML-KNN: A lazy learning approach to multi-label learning[91]	Zhang and Zhou	Pattern Recognition	2007
5	193	Multi-label neural networks with applications to functional genomics and text categorization[46]	Zhang and Zhou	IEEE Transactions on Knowledge and Data Engineering	2006
6	155	iLoc-Euk: A multi-label classifier for predicting the subcellular localization of singleplex and multiplex eukaryotic proteins[52]	Chou et al.	PLoS ONE	2011
7	149	Improved boosting algorithms using confidence-rated predictions[95]	Shapire and Singer	ACM Conference on Computational Learning Theory	1998
8	142	Optimization method based extreme learning machine for classification[92]	Huang et al.	Neurocomputing	2010
9	128	Semantic annotation and retrieval of music and sound effects[96]	Turnbull et al.	IEEE Trans. Audio, Speech and Language Processing	2008
10	127	A kernel method for multi-labeled classification[44]	Elisseeff and Weston	NIPS	2001

method to tackle this issue is to combine the results of several classifiers in an ensemble. For each classifier in the ensemble a subset of the training set (i.e., 63%) is sampled without replacement and a PS classifier is trained. During the prediction stage, outputs are combined by a voting scheme and a threshold separates relevant and irrelevant labels. EPS proved to be competitive with LP and RA*k*EL in terms of efficiency and predictive accuracy.

RAndom k-labEL sets (RA*k*EL)[109] is based on random projections of the label space and builds an ensemble of LP classifiers, each one trained with a random subset of k labels. Thus, it is able to deal with label correlations whilst avoiding the computational complexity of LP. During classification, the output of the classifiers is averaged per label and thresholding is used to assign the label set. Disjoint and overlapping subsets of labels were studied and experiments showed that both improved over LP, even in large data sets. RA*k*EL with overlapping label sets and C4.5 as base classifier improved RA*k*EL with disjoint subsets. In order to minimize the noise or errors in collaborative filtering, cost-sensitive approaches of RA*k*EL and stacking have been applied to audio tagging by considering the tag count as a misclassification cost.[110] In general, the cost-sensitive version outperformed the cost-insensitive counterpart.

LPBR[108] is a hybrid method which carried out combinations of LP and BR rounds. After a first round with BR, the most dependent labels were clustered into a new label. LP was applied within the groups of dependent labels (a group with a limited number of labels) while BR was applied to the independent groups of labels. The process was repeated until the accuracy did not improve. The approach where the dependence between labels is computed by using the χ^2 score was called *ChiDep*. In order to improve the performance, *ChiDep Ensemble* (CDE) was also proposed. A high number of random label-set partitions were generated (e.g., 10,000), and for each partition a score was computed based on sum of the χ^2 of the pairs of labels in the partition. The top high-scored partitions were selected as members of the ensemble. It obtained competitive prediction results. Train time of ChiDep and CDE was relatively long and approximate to that of RA*k*EL and 2BR. Test time of ChiDep was comparable to BR while test time of CDE was longer.

Finally, in Refs 166 and 167, an improvement of RA*k*EL, where the subsets were not selected randomly but in advance was described. The subset selection

TABLE 5 | Taxonomy of the MLL Transformation Methods

Ranking via single-label learning	
Ignore	12
Select (*min*, *max*, *random*)	
Copy, copy-weight	
Binary	
BR	12
CC	97
PCC	98
2BR (MBR)	38, 99 and 100
BR+	101
Pairwise	
RPC	102
CLR	103 and 104
QCLR	105
DLVM	106
Label combination	
LP	90 and 12
PS (PPT)	107
LPBR(ChiDep)	108
Ensembles of MLL methods	
ECC	97
EBR	97
EPS	107
RA*k*EL	109
RA*k*EL-CSML	110
CDE	108
RAKEL++	111
RF-PCT	112
RFML-C4.5	13
TREMLC	113
DST-fusion	114
Other transformations	
InsDif	91
OBO	115

problem was formulated as a *Set Covering Problem* (SCP) (i.e., cover the set of labels by a set of label subsets of a given size k) and an approximation greedy algorithm was used to derive the subsets. The work has been recently generalized[111] by proposing a general framework that allows the application of a wide variety of optimization criteria (i.e., balanced representation of each label, coverage of inter-label correlations, or both). Authors have also proposed in this study RAKEL++, an improved version of RA*k*EL in which, instead of voting, the confidence values are

TABLE 6 | Taxonomy of the MLL Adaptation Methods

Decision trees	
ML-C4.5[116]	M2[117]
PCT[118]	IS-MLT[36]
Support vector machines	
SVM-HF[75]	BandSVM, ConfMat[75]
ML-PC[119]	SVM-ML[120]
PSVM[121]	SSVM[121]
OVO3C-SVM[122]	OVODL-SVM[123]
Rank-SVM[44]	Calibrated-RankSVM[124]
Instance-based	
ML-kNN[65]	IBLR[125]
BR*k*NN[126]	LP*k*NN[126]
DML-kNN[127]	KNNMLC[32]
kNN-MLR*[128]	FkNN[30]
FV-kNN[129]	EML-kNN[130]
FSKNN[131]	Mr.kNN[132]
Neural networks	
MMP[76]	BP-MLL[46]
DMLPP[14]	CMLPP[105]
QCMLPP[105]	ML-RBF[133]
PNN[134]	PNN-centroid[135]
ML-FAM, ML-ARAM[136]	MLPP[105]
Generative and probabilistic models	
Multi-label Mixture Model*[137]	PMM1[17]
PMM2[17]	EPMM[138]
CoLModel[139]	MAdGen[140]
Flat-LDA[16]	Prior-LDA[16]
Dependency-LDA[16]	CML, CMLF[72]
Associative classification	
MMAC[141]	RMR[142]
BF-TP, DF-TP[143]	CLAC[144]
Bio-inspired approaches	
MuLAM[49]	G3P-ML[145]
GC[146]	GEP-MLC[147]
ML-2OKM[148]	Moml[149]
EnML[150]	GACC[151]
Ensembles	
AdaBoost.MH[18,74]	AdaBoost.MR[18,74]
AdaBoost.MH*kr*[152]	AdaBoost.MH with discr.[153]
AdaBoost.MH*kr* with discr.[153]	ADTboost.MH[154]
AdaBoost.SZ[155]	AdaBoost.SP[155]
MLBoost[156]	MP-Boost[157]
MSSBoost[158]	ML-RDT[159]
CCA-OC[160]	ML-BCHRF, ML-CRF[161]
MCSP-ECOC[162]	FDT[163]

Names with a final * have been given by the authors of the present paper.

taken into account by thresholding the average of the probabilities provided by the base-classifier for each label. Besides, it uses built-in cross validation for deriving the threshold value. The proposed strategies perform in a efficient and stable manner overall obtaining better results than RA*k*EL.

Other Transformations

Other transformations that do not fit into the previously defined categories have been described. An example is *INStance DIFferentiation* (InsDif)[91] whose aim is exploiting the relationship between the input ambiguity and output ambiguity by supposing that an object belongs to several labels due to the diverse information embodied in the object (in the input space). For each label, a prototype vector was calculated by averaging all instances belonging to such label. Then each example was transformed into a bag of q instances each one being equal to the difference between the original example and one of the prototype vectors. A two-level classification strategy learnt from the transformed data set. Finally, in Ref 115, a *one-by-one* (OBO) transformation generated a new data set for each label where instances were only the positive ones and each subproblem was solved by means of a *one-class* classifier (i.e., Support Vector Data Description[168]). To compensate missing correlations between labels, linear ridge regression integrated predictions of all subclassifiers.

Ensembles of MLL Methods

Ensemble methods whose base classifiers are multi-label learners are considered by Madjarov et al.[13] a special group of methods because they are developed on top of problem transformation and algorithm adaptation approaches. The RA*k*EL, EPS, ECC, EBR, PCC, and CDE approaches previously described are examples of ensembles of MLL methods. Binary methods are occasionally referred to as ensemble methods because they use multiple binary models. As none of these models is multi-label capable, the term ensemble is preferable in the sense of an ensemble of MLL methods.[97]

Other than the above-cited methods, *Random Forest of Predictive Clustering Trees* (RF-PCT)[112] is an ensemble that uses PCT (described in *Decision Trees* section) as base classifier. The diversity among the base classifiers is obtained by using bagging and selecting, at each node, the best feature from a random subset of the input attributes. The outputs are combined using a voting scheme. Finally, *Random forest of ML-C4.5* (RFML-C4.5)[13] follows the same philosophy but uses ML-C4.5 trees (described in *Decision Trees* section) as base classifiers. In Ref 13, an intensive experimental evaluation involving a wide variety of algorithms, metrics, and statistical tests was carried out in which RF-PCT obtained very competitive prediction results. The main findings of this experimental evaluation are analyzed in *Pitfalls and Guidelines* section.

Triple-Random Ensemble MLC (TREMLC)[113] integrated the ideas of random subspace method, bagging, and RA*k*EL by randomly selecting feature subsets, instance subsets, and label subsets respectively to build an ensemble of LP multi-label classifiers. Its performance was competitive but with a high computational cost maybe due to this triple randomization. In Ref 114, the feature space was divided into a number of subsets and a baseline induction algorithm (i.e., AdaBoost.MH described in *Ensembles* section) was run over each one. Confidence outputs of AdaBoost.MH were combined by an approach based on the Dempster-Shafer theory[169] outperforming traditional voting schemes.

Algorithm Adaptation Methods

Decision Trees

Clare and King[116] proposed ML-C4.5, an adaptation of the C4.5 algorithm to the MLL setting. It allowed multiple labels in the leaves and modified the definition of entropy in order to consider not only membership, but also nonmembership of each class. This method has become a reference and has been mainly used as base classifier in ensembles of MLL methods (e.g., RFML-C4.5 described in *Ensembles of MLL Methods* section).

Predictie Clustering Tree (PCT)[118] is framework for prediction that can be instantiated to a particular task by defining a distance metric and a prototype. Thus, PCTs have been used for predicting tuples of variables, time series, and even classes organized into a hierarchy (this issue is described in *Hierarchical MLC* section). Particularly, in MLL, each label is a component of the target tuple. A PCT is top–down generated. At each node, data are partitioned into clusters in such a way that the intra-cluster variation is minimized. The result of the induction process is a decision tree in which each leaf contains the prototype of the instances belonging to that leaf. PCTs have yield very good classification results combined with random forest (RF-PCTs are described in *Ensembles of MLL Methods* section).

The two described approaches are the most widely used, besides other tree-based algorithms have been proposed. First, in Ref 117, the fact that the greedy search of predictors tends to select those with many splits was analyzed and the *M2 tree* was

introduced. It was a two-stage method that separated the splitting-variable selection (using the statistic test of Nettleton and Banerjee[170]) and the splitting-point selection (that generates binary partitions of data) steps. M2 reduced the bias in predictor selection, but accuracy values were lower when the target vector did not have any significant correlation structure. Finally, it is worth citing *Iterative Split Multi-Label decision Tree* (IS-MLT)[36] that was built following a top–down strategy. At each node, the set of labels was split into two groups by clustering. These two sets become the target objective over which to find the best split with a SVM. The leaf node frequencies were considered scores and labels were assigned by threshold.

Support Vector Machines

Many approaches have used single-label SVMs with OVA approach.[90,171–173] In Ref 75, 2BR (described in *Binary Methods* section) was called *SVMs with heterogeneous feature kernels* (SVM-HF) as it considered SVM in both base and meta-learning levels. Besides, two mechanisms for improving the margin quality of SVMs in an OVA setting were presented. The first one, *band-removal method* (BandSVM), operated at the instance level. Once an OVA SVM had been learnt, it removed similar negative examples that were within a threshold distance (band) from the learned decision hyperplane. Then, the model was re-trained. *Confusion-matrix-based pruning method* (ConfMat) worked at the label level. A confusion matrix, *M*, over the original learning problem was obtained with any fast, moderately accurate classifier. Each M_{ij} represented the percentage of class *i* misclassified as class *j*. Then all training examples of confusing classes were removed and an OVA SVM was learnt. ConfMat was faster to train than BandSVM and required only one OVA SVM step. Experiments on text classification showed that SVM-HF and BandSVM were comparable in their results, being better than ConfMat and OVA SVM.

Other authors have developed OVO decomposition with strategies to treat the special case occurring when samples have double labels at the same time. With this aim, *Multi-Label Paired Comparisons* (ML-PC)[119] separated each pair of overlapping classes by using two probabilistic binary classifiers (e.g., SVM). The individual probabilities of the binary classifiers were combined with the extended Bradley-Terry model with ties.[174] It yielded competitive predictive results but the method to combine predictions was computationally expensive, so it may not be proper for large data sets. Later, in Ref 121, two algorithms known as *Parallel SVMs* (PSVMs) and *Sequential SVMs* (SSVMs) were devised. Double-labeled instances were considered as a new independent class, and two parallel hyperplanes were used to separate the three possible classes. The drawback of this strategy was the number of binary classifiers needed for data sets with large number of labels. In Ref 122, the *triple-class SVM* (OVO3C-SVM), that was able to deal with positive, negative, and mixed classes simultaneously, was proposed. It performed well and the algorithmic complexity was less than PSVMs. Finally, the so-called *Double Label SVM* (OVODL-SVM)[123] considered double-labeled instances to be located at marginal region between positive and negative instances by building a double-label SVM in which a bias term was needed. The predictive performance was comparable to OVO3C-SVM but faster at train and test procedures.

The algorithm adaptation approach has also been used. Thus, Elisseeff and Weston[44] proposed *Rank-SVM*, a ranking method that has become a reference in MLL. It defines a set of *q* linear classifiers that are optimized to minimize a measure, which evaluates the average fraction of label pairs that are reversely ordered for the instance (i.e., the empirical ranking loss defined in Table 1). In addition, it handles nonlinear cases with kernel trick. The optimization is carried out under quadratic programming framework in its dual form. This quadratic programming problem has high computational complexity due to a huge number of variables to be solved. In order to obtain a bipartition from the ranking generated, a threshold selection stage, named *set size predictor*, is carried out. In this stage, a proper threshold is determined on the basis of the obtained rank using linear least squares. This stage has no interactive effects with the rank stage, so the threshold selected may be not the optimal. Due to this reason, *Calibrated-RankSVM*[124] added a virtual label to determine the relevant labels whose optimal coefficients were determined embedded during the ranking learning process. Nevertheless training procedure is still time consuming. Finally, *SVM-ML*[120] was proposed to overcome these drawbacks. It consisted of adding a zero label to detect relevant labels and simplified the original form of Rank-SVM. This led to a novel quadratic programming problem in which each class had an independent equality constraint. It reduced the computational cost in comparison to Rank-SVM and obtained competitive performance.

Instance-Based Algorithms

Multi-label K-nearest neighbor (ML-kNN)[65] was one of the first lazy MML proposals. After determining the *k* nearest neighbors, a membership counting vector with the number of neighbors belonging to each possible class was computed. Based on this statistical

information, gained from the label sets of the neighbors, the set of labels for the unseen instance was identified by using the *maximum a posteriori* (MAP) principle. The experiments showed that it performed well on several real-world data sets. Nevertheless, it is often criticized because it does not take into account label correlations. Therefore, *dependent multi-label kNN* (DML-kNN)[127] was proposed as a version of ML-kNN where dependencies between classes were considered by defining a *global* MAP rule. Unlike ML-kNN, that for each label to be predicted considered only the number of neighbors containing such label, DML-kNN took into account the numbers of all labels in the neighborhood. The experiments conducted proved the effectiveness of the method as compared to ML-kNN.

Instance-Based Learning by Logistic Regression (IBLR)[125] combined instance-based learning (IBL) and logistic regression. It took label correlations into account by using the labels of neighbor examples as extra attributes in a logistic regression scheme. The experimental results proved its performance on several real-world problems. Together with ML-kNN, it can be considered the state-of-art in instance-based MLL.

BR*k*NN[126] is also worth citing. It was equivalent to using BR with kNN as the base classifier, but much faster because, instead of computing q times the k nearest neighbors, it only searched once. Two extensions, dubbed BR*k*NN-a and BR*k*NN-b, to tackle the case where BR outputs an empty set for any test instance, were proposed. Defining the confidence of a label as the percentage of the k nearest neighbors that included it, the first method output the label with the highest confidence, and the second one calculated the average size of the label sets of the k nearest neighbors, s, and then output the $[s]$ (nearest integer of s) labels with the highest confidence. The experiments carried out supported that both extensions were beneficial. Besides, authors proposed LP*k*NN, which consisted of an LP transformation with kNN as the base classifier and also yielded good performance results. In Ref 128, MLR was described as a special case of rank aggregation (with ties) supplemented with an additional virtual label within a case-based framework.

In the field of image annotation, kNN multi-label classification (KNNMLC)[32] was developed with the aim of reducing the bias between semantic similarity of concepts and visual similarity of images annotated with these concepts. It combined a weighted version of kNN and multiple SVM classifiers, used for jointly finding optimal margins in both spaces.

Approximate reasoning techniques have also been incorporated in many proposals. For instance, FkNN[30] adapted the well-known *Fuzzy kNN algorithm.*[175] First, the k nearest neighbors were found and then, a membership degree for each label was computed based on the memberships of the neighbors to each class and on the distances between the test instance and the neighbors. It output a membership degree for each label and the bipartition was obtained applying an α-cut. Other example was *Mr.kNN,*[132] a two-step approach. First, a modified fuzzy c-means clustering algorithm, that treated each label as a cluster, assigned each training example a soft relevance value for each label that indicates the strength of an instance related to a label. Then, a kNN algorithm implemented a voting factor based on the soft relevance of each neighbor and on the distances between a test instance and its neighbors. Reported experiments showed it outperformed ML-kNN, nevertheless its computational complexity was higher.

Evidential Multi-Label kNN (EML-kNN)[130] used an evidence-theoretic rule that extended the Dempster–Shafer framework to the MLL setting with only a moderate increase in complexity as compared to the classical case. *Fuzzy Veristic kNN* (FV-kNN)[129] used a fuzzy kNN rule for MLL based on the theory of *Veristic Variables.*[176] It was able to generate fuzzy label sets for instances that have been originally labeled by crisp ones and obtained competitive results. Finally, FSKNN[131] was based on a *fuzzy similarity measure* (FSM) and kNN. The main aim was reducing the computational power required for finding the neighbors in kNN-like algorithms. First, a FSM grouped the training patterns into clusters. Given a test instance, only those clusters whose similarities to the test instance exceeded a predefined threshold were used to calculate the nearest neighbors. An unseen document was labeled based on its nearest neighbors using the MAP estimate. Experiments in text categorization showed the reduction of the computational costs while the performance was maintained.

Neural Networks

Crammer and Singer[76] proposed *Multi-Label Multi-Class Perceptron* (MMP), a family of online algorithms for topic-ranking on text documents. One perceptron was used for each label but, unlike BR, the performance of the whole ensemble was considered to update each perceptron. They showed that MMP outperformed BR on text classification tasks.

The pairwise approach is often regarded as superior to BR because the former profits from simpler decision boundaries in the subproblems. Thus, while in MMP one perceptron was trained for each class, in Ref 105, *multi-label pairwise perceptron* (MLPP)

was described as the instantiation of the RPC transformation with perceptrons as base classifiers. It was less efficient than MMP, but it resulted in a gain of accuracy and was able to tackle large text corpora (i.e., RCV1) in a pairwise approach. In order to alleviate the complexity of the RPC approach, quadratic with the number of labels, *dual MLPP algorithm* (DMLPP)[14] formulated the perceptrons in dual form. Thus, the prediction time depended linearly on the number of labels. Authors recommend this approach when the number of classes is high. Nevertheless, it is still less efficient than MMP and, as it keeps the whole training set in memory, it has problems to handle training sets with many instances. On the contrary, MLPP is advisable if the number of classes is low and the number of examples high. Instead the RPC approach, CMLPP[105] used a CLR decomposition strategy, and QCMLPP[105] used iteratively the QWeighted voting schema (described in *Pairwise Methods* section) until the calibrated label was found. Reported experiments showed that QCMLPP achieved a good trade-off between predictive performance and time complexity.

The algorithm adaptation approach has also been used. Zhang and Zhou[46] developed *Backpropagation for MLL* (BP-MLL), an adaptation of the traditional multilayer feed-forward neural network. The key idea is the definition of an error function, closely related to the ranking loss (defined in Table 1). The error function is minimized with gradient descent combined with the error backpropagation. The net has one input unit per input feature, one output unit per label, and the hidden layer is fully connected with weights to the input and output layers. Its computational complexity in the training phase is high, but the time cost of predictions is quite trivial.

Multi-Label Radial Basis Function (ML-RBF)[133] was inspired in the well-known RBF method. The first layer was obtained through k-means clustering on instances of each possible class, the centre of each cluster being the prototype vector of a basis function. Weights of the second layer were learnt by minimizing a sum-of-squares function. Experimental results showed it outperformed methods as Rank-SVM and BP-MLL in a wide range of metrics and data sets. However, while the prediction time was similar to BP-MLL, training time was much less in ML-RBF.

It is worth citing works adapting other models of artificial neural networks. *Probabilistic Neural Network* (PNN)[177] has been adapted to the MLL setting in the field of text categorization.[134] Later, *PNN-centroid*[135] used a technique of centroids with good performance results that was faster and needed less memory than PNN. Finally, ML-FAM and ML-ARAM[136] were two extensions of the neuro-fuzzy models based on *Adaptive Resonance Theory* (ART) that outperformed the single-label proposal.

Generative and Probabilistic Models

Generative models have been developed, mainly related to text categorization, under the assumption that a document is generated by a mixture of single-label document models (one per category). In Ref 137, McCallum presented a probabilistic generative model that was based on naive Bayes with *expectation maximization* (EM)[178] to learn the mixture weights and the word distributions in each mixture component. The proposed model tried to capture the relationship between the classes and word occurrences, but it did not consider the correlation within the classes.

Parametric Mixture Models PMM1 and PMM2[17] were two probabilistic generative approaches in which documents were modeled by a single multinomial distribution for each class. Experiments showed that PMM obtained better classification performance than the binary decomposition in several text categorization problems. The PMM was a very efficient model but tended to underfit, which led to the development of the *Extended PMM* (EPMM).[138] It incorporated *latent categories* into PMM, so that the model complexity could be adaptively controlled according to the given data. The experiments conducted showed classification performance higher than PMM. *Correlated Labeling Model* (CoLModel)[139] was proposed to formulate the correlation between different classes. It captured the underlying structures via the latent random variables in a supervised manner. In *Additive-Generative Multi-Label Model* (MAdGen),[140] a deconvolution approach estimated the individual contribution of each label to a given data item and, in Ref 16, a set of three models based on the *Latent Dirichlet Allocation* (LDA)[179] framework was presented.

Finally, *Conditional Random Fields* (CRFs)[180] have been used in *collective multi-label* (CML) and *collective multi-label with features* (CMLF),[72] two multi-label graphical models for text classification that parametrized label co-occurrences. In the field of image categorization, CRFs have been used to capture associations between labels in Refs 35 and 181.

Associative Classification

Multi-class multi-label associative classification (MMAC)[141] was one of the first algorithms for MLL based on associative classification. The antecedent of a rule was a set of attribute-value pairs in conjunctive form (i.e., an item), and the consequent was a list of ranked class labels. MMAC first applied association

rule mining over the training data to discover and generate an initial set of classification rules in which each rule was associated with the most obvious class label. After that, the process was repeated and new rules sets were generated from the remaining unclassified instances, until no more frequent items could be discovered. The rules sets derived at each iteration were merged to form a multi-label classifier. As in different iterations the same antecedent, but with different class labels, might have been discovered, the algorithm merged such antecedents and ranked the labels according to their frequency in the training patterns satisfying the antecedent. The same authors proposed *Ranked Multi-Label Rule* (RMR),[142] similar to MMAC but including a pruning step that removed training objects shared by rules.

Although MMAC tackled with multiple labels, it generated rules by iteratively repeating the adopted method for generating single-label rules. In Ref 143, an algorithm that generated multi-label rules in a single run was proposed. It was based on the *tree-projection-based frequent pattern mining* algorithm[182] and had two possible versions based on *breadth-first* (BF-TP) and *depth-first* (DF-TP) search algorithms.

Finally, *Correlated Lazy Associative Classification* (CLAC)[144] was a lazy approach, which delayed the inductive process until a test instance that was used as a filter to remove irrelevant examples from the training data, was given for classification. After that, a greedy heuristic built a specific model for the test instance. This model was composed of class association rules that satisfied thresholds of support and confidence and included labels into the antecedent. By including labels in the antecedent, interactions among labels were explored.

Bio-Inspired Approaches

Several bio-inspired approaches have been described that built a MLL classifier. One example is *Multi-Label Ant-Miner* (MuLAM)[49] that was based on the *Ant Colony Optimization* (ACO) Ant-Miner algorithm.[183] Unlike the original Ant-Miner, a pheromone matrix was created for each class, each ant discovered a set of rules (at most one rule for each label) and more than one class in the rule consequent was allowed. Experimental results did not showed significant differences with the single-label proposal. A subsequent version for hierarchical MLL is described in *Hierarchical MLC* section. GEP-MLC[147] was another proposal based on *Gene Expression Programming* (GEP).[184] Each individual codified a discriminant function that was applied to the input features of the pattern to produce a numerical value in such a way that a threshold determined membership to a class. A population of discriminant functions evolved and a niching algorithm was used to guarantee diversity in the solutions and to determine the functions that finally composed the classifier. Later authors proposed GC,[146] also based in GEP, but in this case each individual codified a rule, a model more interpretable than a discriminant function. The final classifier consisted of a set of rules and several labels were allowed as consequent. Reported experiments in both works showed competitive results in data sets on different domains. Finally, it is worth citing G3P-ML,[145] a Grammar-Guided Genetic Programming algorithm in which a population of classification rules evolved following a classical generational and elitist evolutionary algorithm.

Other approaches have been focused on the optimization of a population of MML algorithms over several objectives simultaneously. Thus, ML-2OKM[148] was a multi-label kernel algorithm, derived from Rank-SVM (described in *Support Vector Machine* section). Two objectives, the model regularization term and the ranking loss, were optimized by using NSGA-II,[185] an elitist multi-objective genetic algorithm (MOGA) that provided a Pareto optimal set of solutions to implement ML-2OKM. Unlike ML-2OKM, which optimized two particular objectives, *Multi-Objective Multi-Label* (MOML)[149] allowed the optimization of any evaluation metric. Its base model was ML-RBF (described in *Neural Networks* section) with an additional regularization term added to reduce overfitting risks. The best models were selected and the final prediction was obtained with a majority vote. The reported experiments showed an improvement in performance accuracy, not only limited to the optimization objectives. *EnML*[150] was another multi-objective evolutionary approach whose aim was to obtain an optimal ensemble consisting of a group of accurate and diverse multi-label base learners (i.e., ML-RBF). Finally, in Ref 151, a genetic algorithm, dubbed GACC, has been proposed to optimize the chain ordering in CC classifiers (described in *Binary Methods* section). Experiments on diverse benchmark data sets indicated the improvement of the output classifiers. Bio-inspired approaches have been also developed in the context of hierarchical MLL. A description of these proposals can be found in *Hierarchical MLC* section.

Ensembles

Schapire and Singer proposed *AdaBoost.MH* and *AdaBoost.MR*,[18,74] two boosting algorithms for text categorization inspired in the popular AdaBoost.[186] The purpose was to find a highly accurate classifier (*final hypothesis*) by combining many classifiers

(*weak hypotheses*), each of which might be only moderately accurate. *AdaBoost.MH* tried to minimize the number of misclassified labels (i.e., the Hamming loss described in Table 1), for which it maintained a set of weights not only over the training examples (as AdaBoost does), but also over the labels. Thus, each round, training examples and their corresponding labels that were harder to predict got incrementally higher weights while examples and labels that were easy to predict got lower weights. In practice, it mapped the original MLL problem into a binary learning problem, which was solved by the traditional AdaBoost algorithm with one-level decision trees as base learners. The purpose of *AdaBoost.MR* was minimizing the number of labels misorderings (i.e., the ranking loss described in Table 1), so relevant labels would be ranked above the irrelevant ones, for which a weight distribution was maintained over examples and pairs of labels.

Many variants of AdaBoost.MH have been subsequently proposed. For instance, *AdaBoost.MH with K-fold real-valued predictions* (AdaBoost.MH^{KR}).[152] Its main idea was producing, at each iteration, not a single weak hypothesis, but a complex weak hypothesis consisting of a subcommittee of simple weak hypotheses (committees of decision stumps), which, at that iteration, looked the most promising. AdaBoost.MH and AdaBoost.MH^{KR} required documents to be represented by binary vectors corresponding to the presence or absence of the terms in the document. In order to overcome this drawback, in Ref 153, the potential of weighted (*tf* * *idf*) representations was studied with AdaBoost-like algorithms by means of two different entropy-based discretization methods. Experiments showed that binary representations obtained after discretization outperformed the original binary representation. Other example is ADTboost.MH[154] that can be viewed as an extension of AdaBoost.MH, that allows a better readability of the classification rules, as well as an extension of ADTBoost,[187] that extends the formalism of ADTrees to deal with multi-label problems. Finally, other boosting-type algorithms have been proposed in Refs 155–157.

Multi-Label RDT (ML-RDT)[159] was based on *Random Decision Tree* (RDT)[188] and built multiple decision trees by selecting, at each step, a random feature until the number of examples of a node was under a threshold or the depth exceeded a limit. Apart from the label probability distribution counting on each leaf node it did not use any information of labels being independent from the number of labels. This fact reduced the computation time and made it effectively handle large number of labels. *Fast Decision Tree induction* (FDT)[163] used feature pre-selection and data partitioning to induce a set of C4.5 decision trees from different subsets of the training data. These subsets were nonoverlapping, equally sized, and maintained the same ratio between positive and negative classes. Multi-label data were handled by inducing a binary classifier for each class separately. The final output was computed by a strategy called *one-vote* that biased the output toward the minority class by considering the label as positive if at least one of the trees said so. *Model-shared subspace boosting* (MSSBoost)[158] was a boosting-type algorithm that worked at feature and data level. Thus, each model was learnt from random feature subspace and bootstrap data samples. It exploited the label space redundancy by sharing base models across multiple labels.

Error Correcting Output Codes (ECOCS or ECC[a]) have been used in multi-class learning due to its capabilities to reduce a multi-class problem to a set of binary-classification problems and also to the improvement in accuracy that their error-correcting properties may provide.[189] The immediate translation to MLL is to consider each combination of labels in the data set as a new one and to apply the ECOC framework in order to build a set of binary-classification problems (e.g., with OVO or OVA coding schemes). This approach was developed in *Multiple Classifier Method for Structured Output Prediction based on Error Correcting Output Codes* (MCSP-ECOC).[162] Other coding schemes have been studied in the field of MLL, e.g., BCH,[161] convolution code,[161] canonical correlation analysis,[160] repetition code,[190] Hamming code,[190] and low-density parity-check code.[190] The framework for MLL ECOCs was formalized in Refs 190 and 191. The main idea was to transform the original multi-label problem into another (larger) MLL task. This was carried out by an ECOC encoder that expanded the original label sets to codewords with redundant information in such a way that provided correcting capabilities. Then a multi-label classifier was learnt considering this augmented label space. During prediction, an ECOC decoder transformed the outputs of the classifier to the original set of labels taking advantage of its error-correcting properties. Experimental results showed the validity of this framework when coupled with classic ECOC coding schemes. Experiments carried out in Ref 191 showed that LP and RA*k*EL did not benefit from ECOC while BR did. The redundancy is just the main drawback of this approach. The coding of original label information always provides a higher dimensionality and this implies a computational cost. Finally, in Ref 192, multi-class ECOC

was interpreted as a way to map a conventional multi-class problem into a multi-label one. This obvious correspondence, and the solution of multi-class problems via MLL, had not yet been noted in the literature.

ONGOING RESEARCH

Label Dependence

Exploring correlations between labels may reduce complexity when data have a moderate or high number of labels. From a probabilistic point of view, two types of dependence in multi-label data have been identified.[98,193] *Conditional-label dependence* captures the dependences between labels given a specific instance, reflecting how likely or unlikely labels are to occur together given the attribute values of a specific instance. This kind of dependence has been explored in Refs 98, 12, and 109. *Unconditional (marginal) label dependence* is independent of a certain instance and refers to the idea that certain labels are likely or unlikely to occur together. This kind of dependence has been explored in Refs 125, 194, and 195.

In Ref 196 a categorization of multi-label techniques based on the order of correlations was considered. Thus, *first-order* approaches ignore label dependences decomposing the problem into a set of independent binary problems (e.g., Refs 90, 154, 197, and 198), *second-order* approaches consider the pairwise relations between labels such as their ranking constraint (e.g., Refs 18, 46, and 104) or their co-occurrence patterns (e.g., Ref 72) and finally, *high-order* approaches consider high-order relations between labels (e.g., Refs 97, 109, 125, and 199).

It is also worth citing that recent studies have pointed to the presence of *asymmetric*[200] and *local*[201] correlations, more consistent with realistic situations. Asymmetric correlations are present when the influence of one label to the other is not necessary the same in the inverse direction (e.g., the label *bear* implies *mammal* but the inverse may not be true). Local correlations are shared by subsets of instances rather than all the instances, and exploiting such correlations globally could affect the performance by predicting some irrelevant labels.

Many approaches have already been described in this paper that tackle with label dependence. Nevertheless it is still worth noting other ones. For instance, in the field of automatic video annotation, *Correlative Multi-Label* (CML)[202] simultaneously classified concepts and modeled correlations between them in a single step. Instances were transformed into high-dimensional vectors by encoding correlation information between inputs and outputs, and a maximum-margin type algorithm learned from these transformed vectors. In Ref 203, a method for modeling the correlations between categories based in the principle of *maximum entropy model* (MEM) was presented. According to Ref 204, it could be effective on a set of samples that vary linearly, but it might fail to capture the structure of the feature space if the variations among the samples were nonlinear (e.g., image classification). Therefore, it was extended to a non-linear case by incorporating a kernel function into the model.[204] Later, in Ref 205, a general framework to encode class dependences was described. A subspace was assumed to be shared among multiple labels that were computed by solving a generalized eigenvalue problem. *MLL by Exploiting lAbel Dependency* (LEAD)[196] was an hybrid generative-discriminative approach, that first built a Bayesian network to encode the conditional dependencies. Then it decomposed the problem into a set of single-label ones considering as additional features its parental labels in the Bayesian network. Its complexity was linear with the number of labels. The reported experiments over a wide range of data sets showed it was comparable to the state-of-the-art especially on large-scale problems with large number of labels and examples.

Other authors have opted for explicitly represent the dependence structure between the classes via *multi-dimensional Bayesian network classifiers* (MBC). An MBC is a Bayesian network of restricted topology consisting of three subgraphs: one for the class variables, other for the feature variables, and a bridge subgraph that interconnects the class and feature subgraphs allowing only arcs from classes to features. The parameter set defines the conditional probability distribution of each variable given its parents. Several approaches have been recently proposed to learn MBCs.[70,206,207] The main drawback of this approach is the high computational cost of determining the optimal network structure and computing the most probable explanation for any instance with unknown values for the classes. Due to this reason in Ref 208, *Bayesian Chain Classifiers* (BCC), combined a first stage, in which Bayesian network with a simple tree-based structure captured the dependency relationships between labels, with a second stage in which a CC (described in *Binary Methods* section) was built based on the dependence structure. The network constrained the possible chaining orders and reduced the number of classes. This approach yielded competitive results both in complexity and predictive performance.

Dimensionality Reduction

Reduction of the Input Space (Features)

This section tackles *feature selection* and *feature extraction* methods. The former obtains a new feature set which is a subset of the original one and the latter obtains new features by combinations and transformations of the original ones.

Feature selection

The *wrapper* approach, in which the subset of features to be used is determined by the learning method, can be directly applicable to MLL by searching for a subset of features optimizing a multi-label loss function on an evaluation data set.[12] Another strategy is transforming the data set into one or more single-label data sets and using any existing *filter* method, in which features are scored depending on a measure and the best ones are selected.[209] Thus, in Ref 210, the χ^2 method was separately applied to each label in order to obtain a ranking of all features for each label. Then the top 500 features based on the maximum rank over all labels were selected. In Ref 37, an LP transformation was applied and then a common attribute evaluation statistic (i.e., χ^2) was used, while in Ref 211, a PS transformation with a greedy feature selection algorithm, based on a nearest neighbor estimator of multidimensional mutual information, was applied. Once the features had been ranked, the original multi-label problem was considered again with all the samples. A two-stage filter-wrapper feature selection strategy was described in Ref 212, It first removed irrelevant and redundant features by using *Principal Component Analysis* (PCA)[213] and then selected the more appropriate subset of features by means of a genetic algorithm whose fitness function addressed the label correlations. Then, MLL was solved using OVA decomposition and binary naive Bayes classifiers with Gaussian density estimation. Finally, *Hybrid Optimization based Multi-Label feature selection* (HOML)[63] combined simulated annealing, hill climbing and a genetic algorithm. It reduced the data dimension, improved the classification performance and yielded competitive results compared to benchmark feature selection/feature reduction techniques.

Feature extraction

PCA and *Linear Discriminant Analysis* (LDA)[214] are two common single-label unsupervised techniques for data classification and dimensionality reduction. In Ref 215, a novel method, dubbed *Multi-label LDA* (MLDA), that generalized the single-label framework and surpassed label correlations, was proposed. Another generalized LDA for multi-label problems was studied in Ref 216 concluding the effectiveness of dimension reduction for high-dimensional data. In Ref 217, *Self-Organizing Feature Map* (SOM) and *Latent Semantic Indexing* (LSI) were studied as unsupervised representations of the input space for text classification. The experiments carried out showed that SOM approach obtained better results with multi-label data while LSI performed better with single-label data.

A criticism of unsupervised methods is that they ignore information in labels, thus *Multi-label informed Latent Semantic Indexing* (MLSI)[218] was presented as a multi-label extension of LSI that carried out a mapping from the input features into a new feature space by taking into account not only the information of the inputs, but also capturing the correlations in the label space. Then a set of q linear SVMs was trained on this projected space. *Multi-label Dimensionality reduction via Dependence Maximization* (MDDM)[219] was another supervised method. Based on the Hilbert-Schmidt independence criterion (HSIC),[220] it performed dimensionality reduction by maximizing the dependence between the feature description and the associated class labels. The reported experiments showed it was slightly superior to PCA and significantly superior to MLSI. Finally, *multi-label learning with Label specIfic FeaTures* (LIFT),[221] considered a different feature set for each label. In a first stage, cluster analysis was conducted for each label considering its positive and negative instances in the training data and new specific features for each label were constructed based on the clustering results. After that, q classifiers were trained taking into account only these specific features. Experiments validated its effectiveness as compared to other well established MLL algorithms.

Reduction of the Output Space (Labels)

Hierarchy Of Multi-label classifiERs (HOMER)[210] is an algorithm whose complexity depends on q. The main idea is the transformation of a MLL problem into a tree hierarchy of simpler MLL tasks, each one dealing with a small number of classes. At each node, labels are split with a balanced clustering algorithm that groups similar labels into a *meta-label*. A multi-label classifier is then built in order to predict one or more of these meta-labels. Each child node filters the data of his parent, keeping only the examples that are annotated with at least one of his own labels. Leaves represent one label of the data set. Experimental results showed that HOMER with BR as classifier at the nodes outperformed BR in time and accuracy. Further experimental research has identified HOMER as a computationally efficient MLL method specifically designed for large multi-label data sets.[13] Later,

in Ref 222, the use of HOMER with a QCLR classifier (described in *Pairwise Methods* section) at each node improved the predictive performance results of HOMER with BR at a small expense in training time and classification time. It also improved simple BR and QCLR in prediction performance, train and classification time, and QCLR also in terms of memory usage.

Label Reduction with Association Rules (LRwAR)[223] applied a different approach. It first run an association rule algorithm over the label space (i.e., FP-Growth[224]) in order to obtain a set of association rules representing the presence of certain labels when others were present. Then, the data set was preprocessed by hiding the inferred labels. After the classification stage, these rules were used to retrieve the relevant labels in a postprocessing phase. Results maintained or even improved the evaluation measures of BR, LP, BP-MLL, IBLR, and ML-kNN and were obtained in a shorter run time.

Other methods map labels into a reduced label space to reduce computational and space complexities. Examples are *Canonical Correlation Analysis* (CCA),[225] *Compressed Sensing* (ML-CS),[194] *Principal-Label Space Transform* (PLST),[226] or *Compressed Labeling* (CL).[164]

Finally, it is worth highlighting that recent research has shifted its focus to problems on large-scale problems where the number of labels is assumed to be extremely large.[227,228] The key challenge being the design of scalable algorithms that offer real-time predictions, have a small memory footprint and even are able to accommodate missing labels (human annotators tag only with categories they know about).

Multi-Instance Multi-Label Learning

In MIML, a pattern (*bag*) is described by multiple instances and each pattern is associated with a set of labels. For example, an image (bag) can contain multiple regions (instances) and the image can belong to several classes simultaneously. With a MIML representation, the relation between the input patterns and the semantic meanings may become more easily discoverable. For example, it can be discovered that one object (bag) has a certain label because it contains a certain instance, and even that the occurrence of several instances triggers other labels.[229] Besides, in many MLL problems, different labels are often tied to the different parts of the object, so, developing classifiers based on the whole object would incur too much noise and harm the performance.[230]

In Refs 231 and 229, two approaches based on a simple degeneration strategy for MIML problems were proposed. The first one used MIL as the bridge and obtained a MIL task by applying a category-wise decomposition. After that, any MIL algorithm could be used or the problem could be transformed again to obtain a traditional supervised learning task. Particularly, Zhou and Zhang proposed using MiBoosting[232] calling this approach MimlBoost. Some authors have noticed that this approach is time-consuming.[233] The second approach used MLL as the bridge. First, the problem was transformed into a MLL task and then any MLL technique could be used. In this case, constructive clustering combined all instances in a bag into a single instance. Then a BR approach was used with SVM as base classifier. This approach was called MimlSvm. Besides authors proved that MIML setting is even useful in tasks where data are represented as single-instance multi-label or as multi-instance single-label examples by using two algorithms, InsDif (described in section Other Transformations) and *SUB-COncept Discovery* (SubCod), which transformed, respectively, examples in single-instance multi-label or multi-instance single-label format to MIML.

Although these reduction processes are feasible, the performance of the resultant algorithms may suffer from the information loss incurred during the reduction process.[234] Thus, several methods have been proposed to learn from MIML examples directly in order to avoid this loss of information and to model the connections between instances and labels. Zhou et al. proposed D-MimlSvm,[229] a regularization method. It achieved better performance than degeneration methods, but it could only deal with moderate size of training set due to the associated demanding optimization problem.[235] MIML-*k*NN[234] not only considered the neighbors of an example, but also those training examples (bags) that counted the example as a neighbor (i.e., citers). In this way, the correlations between instances and labels of an example were exploited. After that, a label counting vector was constructed from its neighbors and citers, and then fed to *q* trained linear classifiers for prediction. Experimental results showed that MIML-*k*NN outperformed MimlBoost and MimlSvm in predictive performance. Its training and testing efficiency were slightly worse than MimlSvm and far superior than MimlBoost.

Maximum Margin Method for Multi-Instance Multi-Label (M^3MIML)[236] was based on a maximum margin method which directly exploited connections between instances and labels. The task was formulated as a quadratic programming problem and implemented in its dual form. Experimental results showed that this algorithm achieved superior performance than MimlSvm and MimlBoost,

but the cost of this learning algorithm was quite high.[233] *Multi-instance MLRBF* (MIMLRBF)[235] used RBF neural networks to learn from MIML patterns and to exploit connections between instances and labels. The input of the first layer was a bag of *d*-dimensional instances. The first layer consisted of medoids (i.e., bags of instances) formed by performing k-medoids clustering on MIML examples for each possible class. Second layer weights of MIMLRBF network were optimized by minimizing a sum-of-squares error function. Each output unit was related to a possible class label. The reported experiments on two real-world data sets showed it was competitive with MimlBoost and MimlSvm. Finally, it is worth citing other approaches for MIML such as the probabilistic generative model called *Dirichlet-Bernoulli alignment* (DBA)[237] or the *Multi-Instance Multi-Label Gaussian Process* (MIMLGP).[230]

Semi-Supervised and Active Learning

In many applications, data are unlabeled or labeling is expensive or impractical. This fact is even more challenging in MLL. Thus, efforts have been also focused in *semi-supervised* (using large amounts of unlabeled data to augment limited labeled data) and *Active Learning* (AL) (the algorithm iteratively asks for labeling examples carefully chosen with the goal of minimizing the labeling effort).

Semi-Supervised Learning

Semi-supervised methods get benefit of the information provided by unlabeled instances outperforming supervised learning when the number of training data is relatively small and the number of classes is large. In the context of MLL, it is worth citing CNMF,[238] in which the key assumption is that two examples tend to have large overlap in their assigned class memberships (class-based similarity) if they share high similarity in their input patterns (input-based similarity). Based in this assumption, CNMF computed two similarity matrices for input patterns and labels. By minimizing the difference of these two matrices, CNMF determined the labels of unlabeled data. The optimization problem was formulated as a *Constrained Non-Negative Matrix Factorization*. Experiments on text categorization showed that CNMF had more stable performance than the competing MLSI approach (cited in *Dimensionality Reduction* section).

While traditional graph-based semi-supervised methods only construct a graph on instance level (in which each node represents one instance and each edge the similarity between corresponding pairwise instances), in SMSE[239] two graphs, on instance (based on both labeled and unlabeled instances) and category level respectively, were constructed. By combining the regularization terms for the two graphs, a regularization framework for MLL was suggested and the labels of unlabeled instances were obtained by solving a *Sylvester Equation*.[240] Experiments on text categorization showed competitive results against binary SVMs, CNMF, and MLSI (cited in *Dimensionality Reduction* section). Zha et al.[241] proposed other graph-based framework for video annotation that used one loss function and two types of regularizer. One was used to tackle the label consistency on the graph and the other was adopted to tackle the correlations of multiple labels. Based on the proposed framework, two novel graph-based algorithms were developed and experiments showed this framework outperformed key existing graph-based methods and semi-supervised MLL approaches. It is worth citing a different approach, based on *Semi-supervised Impurity-based Subspace Clustering* and called SISC-ML.[242] During subspace clustering, labeled and unlabeled examples were used and prototypes maintained the summary about the percentage of each label within each cluster. To obtain predictions, a kNN approach was used considering *k* nearest neighbor clusters. It was applied to text categorization and performed well even when a very limited amount of labeled training data was available.

Most of the approaches on semi-supervised MLL work under the transductive setting. Nonetheless, *inductive MLC with Unlabeled data* (iMLCU)[243] is one recent work which tackles semi-supervised MLL under the inductive setting. The inductive semi-supervised MLL is formulated as an optimization problem of learning *q* linear models, which fits labeled data by exploiting pairwise label correlations and uses unlabeled data via appropriate regularizations. After that the resulting optimization, which is nonconvex, is solved via the *ConCave Convex Procedure* (CCCP).[244]

Active-Learning

The key in active learning is the sample selection strategy whose aim is choosing the most informative instance to obtain the best classification performance. BinMin[245] selected unlabeled examples with respect to the most uncertain label and OVA was used for MLL with SVM as the base classifier. This method did not take advantages of the multi-label information. In the field of image classification, it was proposed the *Mean Max Loss* (MML) or 1DAL strategy,[173] that selected the unlabeled instance that had the maximum mean loss value over the predicted classes. One SVM was trained for each label and a threshold cutting method decided the relevant labels. The overall loss

value was averaged over the labels. As this strategy selected only along the sample dimension, it did not take advantages of the label correlations to reduce human labeling cost. Besides, when one sample was selected, all its labels had to be labeled. 1DAL was improved in Ref 246, where *Two-Dimensional Active Learning* (2DAL) was described. It considered both relationships between samples and between labels. Sample-label pairs were selected in order to minimize the multi-label Bayesian error bound. This allowed the annotation of a subset of labels and the inference of the rest of labels was performed from labels correlations with EM. It improved 1DAL and random selection in image classification tasks and was later extended in Ref 247 to the multi-view learning framework (described in *Other Related Learning Settings* section). By taking advantage of both active learning and multi-view learning, the annotation effort was reduced in comparison with random selection, 1DAL and 2DAL. As multi-view learning and active learning can be effectively integrated, this may be a line to be explored in the MLL setting.

An approach for text classification, so-called MMC, was proposed in Ref 248. One SVM was trained for each label and the overall loss of the classifier was measured by gathering the loss of all binary classifiers. Instead of estimating the labels for each instance, the number of labels was estimated by applying logistic regression. The training features of the logistic regression model were the probabilities obtained by the binary classifiers and the number of labels was the categorical target to be predicted. It outperformed random selection, 1DAL, and Bin-Min in the domain of text classification and reduced significantly the labeling cost. Finally, in Ref 249, several strategies to carry out a global labeling in text classification, in which a unique ranking of unlabeled patterns combined the outputs of q individual binary classifiers, have been proposed.

On-Line MLL and Data Streams

Many challenging real-world problems involve multi-label data streams and learning in such scenarios has special requirements: (1) one example is processed at a time and only once, (2) there are limitations of memory and time, (3) data distribution evolves producing concept drift, and also (4) the model must be ready to predict at any time. Very few authors have explored this task in a MLL setting. In Ref 250, the incoming data were partitioned into chunks and, to take advance of label correlations, a SVM-HF classifier (described in *Binary Methods* section) was built for each chunk. Concept drift was managed by keeping only the latest trained classifiers and discarding the oldest ones. The empirical results showed that it outperformed partitioning data into chunks with BR as classifier, that does not consider label correlations, as classifier.

Later, Read et al.[195] discussed the use of BR, LP, CC, and PS transformation methods in evolving scenarios by instantiating incremental base classifiers. Besides, an on-line *Multi Label Hoeffding Tree* that used the Hoeffding bound[251] to determine the number of instances needed to split a node was described. The tree used the multi-label definition of entropy proposed by Clare and King[116] and applied a PS classifier at the leaves. It achieved faster and more accurate performance than the cited transformation methods. Besides, authors used Adwin *Bagging*[252] to deal with concept drift. In Ref 253, a framework to generate synthetic multi-label data streams can be found. Its aim is facilitating the study and evaluation of MLL algorithms in a data stream scenario.

Finally, it is worth citing *Multiple Windows* (MW),[254] which used a double-window mechanism for each label (one for positive examples and one for negative examples) to deal with the fact that labels do not drift simultaneously and with the same rate (multiple concept drift). Besides, this mechanism was able to deal with the imbalance problem by oversampling the positive examples and undersampling the negative ones according to a user defined parameter. It outperformed SVM-HF.

Hierarchical MLC

In *hierarchical MLC*, in contrast to *flat classification*, examples can be associated with multiple labels and labels are organized in a hierarchical structure such as a tree or a *directed acyclic graph* (DAG), which allows a child category to have more than one parent category. This entails several challenges. First, classes in the bottom of the hierarchy tend to be more difficult to identify because the number of samples is usually less than in upper classes. Secondly, the closer a category is to the root, the more a wrong decision affects lower levels. And finally, predictions must respect the class hierarchy. Thus, according to the *true path rule* (TPR), borrowed from the Gene Ontology and FunCat taxonomies,[255] an example that belongs to some class automatically belongs to all its ancestors, and negative predictions for a given node are propagated to the descendants to preserve the consistency of the hierarchy. Typical examples of hierarchical domains are protein function prediction and text categorization.

In Refs 256 and 257, two approaches for hierarchical MLC are described: the *local* one (top–down)

consists of training a hierarchy of classifiers (e.g., SVM or decision trees), which are used in a top–down fashion for the classification of new examples; and the *global* one (one-shot, big-bang), that induces a unique classifier using all classes of the hierarchy at once and is able to predict just in one step. Three baseline approaches were identified in Ref 94, the two first corresponded to local methods and the last one corresponded to global methods: *Single-label Classification approach* (SC), *Hierarchical Single-label Classification* (HSC), and *Hierarchical* MLC (HMC).

SC trains a binary classifier for each label in the hierarchy considering as positive examples those labeled with such class and the rest are considered to be negative. This approach has several drawbacks. First, it needs to train one classifier per class (which can be hundreds or thousands). Besides, as the hierarchy is not taken into account, it is also possible having inconsistencies. Finally, it is very likely having a problem of imbalanced data at lower levels of the hierarchy with only a few positive patterns and too many negative ones.

HSC, also called *hierarchical* BR (HBR),[12] consists of adapting SC by considering the hierarchy during the prediction in such a way that a classifier only predicts positive if the classifier for the parent class also makes a positive prediction. The hierarchy can also be considered during the training step by restricting the training set of a classifier to those instances belonging to the parent class. HSC has been followed in Refs 40 and 79.

Finally, HMC consists of generating a global model which is able to predict the class associated with a pattern at any level of the hierarchy. This approach has been followed in Refs 258, 259 and 257.

In the group of local methods, it can be cited *Bayes-optimal classifier* (HBayes).[260] It followed the HBR approach but the output for a given pattern was computed by a bottom–up process. Later, HBayes-cs[42] tackled the problem of sparsity of annotations by a cost sensitive parameter to control the trade-off between precision and recall. According to the authors, HBayes-cs resulted more suitable for dealing with skewed data sets. Esuli et al. presented *TreeBoost.MH,*[261] a recursive algorithm that generated an AdaBoost.MH classifier for each non-root category. As multi-label (instead of binary) classifiers were built, it is considered a generalization of HBR.[12] In Ref 19, three local approaches for text classification with DAG categories were proposed: *flat*, *tree-based*, and *DAG-based* that, respectively, transformed the DAG into a flat structure (in which categories were treated in isolation), an equivalent tree or generated one classifier per parent when the node had several parents. SVM were used as base classifiers. The results showed that the flat approach had a comparable performance to the hierarchical approaches when the number of categories involved was small and tree-based and DAG-based approaches had nearly the same classification accuracy, but the former tended to produce larger trees. To finish with local methods, in Ref 41, the convenience of combining techniques of hierarchical classification (to take into account relationships between classes), data fusion techniques (to integrate multiple sources of data), and cost-sensitive methods (to address the imbalance between positive and negative examples) has been studied. It was found that the combined action of some of these techniques achieved better performance than the average of the performances of the strategies used separately.

Regarding to global methods, Clare[197] proposed one based on the ML-C4.5 described in *Decision Trees* section. It modified the definition of entropy to take into account both the multi-label aspect and the hierarchical relationship between labels. In Ref 257, a global approach applied to a DAG hierarchy was presented. The main idea was transforming an initial (possibly single-label) task into a multi-label task by expanding the label set of each training example with the corresponding ancestor labels. After that AdaBoost.MH algorithm was applied and, finally, inconsistently classified instances were re-labeled. Experiments carried out in annotation of gene functions demonstrated that the approach improved the flat AdaBoost.MH (i.e., AdaBoost without considering any hierarchical information) and was comparable to local AdaBoost.MH. Clus-HMC,[258] based on the PCT algorithm described in *Decision Trees* section, is other global approach, which trains only one decision-tree to cope with the entire classification problem. Experiments concluded that it was fast, identified features relevant for all the labels together and performed similar to Clus-SC (that learnt a separate PCT for each class). Clus-SC, Clus-HSC and Clus-HMC were later compared in Ref 94 finding that Clus-HMC performed better for tree and DAG class hierarchies.

HMC has also been tackled with bio-inspired approaches. Thus, *Hierarchical Classification Ant-Miner* (*hm*Ant-Miner)[50] was based on ACO and discovered a global classification model in the form of an ordered list of IF-THEN rules. The construction of a rule was divided into two ant colonies which worked in a cooperative manner, one for the rule antecedents and the other for the rule consequents. At each iteration, a rule was built by the pairing of an antecedent ant with a consequent ant. It operated both with tree or DAG hierarchies and the results obtained were competitive in terms of

accuracy and complexity of the model. *Artificial immune systems* (AIS) were used in Refs 47 and 48 defining a local and a global version of an algorithm called *Multi-label Hierarchical classification with AIS* (HMC-AIS) to discover classification rules for protein function prediction. A sequential covering procedure iteratively called a rule evolution procedure, that evolved classification rules (antibodies) and added the best one to the set of discovered rules, until all (or almost all) training examples (antigens) were covered by the discovered rules. *Hierarchical* MLC *with Genetic Algorithm* (HMC-GA)[256] was another bio-inspired approach. It evolved the antecedents of decision rules with a sequential covering strategy, removing from the training set examples already covered by the generated rules. The fitness function was biased towards rules with high example coverage. Experiments showed that *hm*Ant-Miner, Clus-HMC and HMC-GA produced considerably less rules than the local methods (i.e., Clus-SC and Clus-HSC), resulting in much simpler and interpretable final models.

Finally an innovative approach which was able to obtain not only the classifier, but also the hierarchy from the multi-label prediction was described in. It is more complex than standard HMC where the class hierarchy is known a priori.

Dealing with Class Imbalance

It is commonly accepted that multi-label data may suffer from imbalance. Thus, *label skew*[78] is defined as a relatively high number of examples associated with the most common label sets, while a relatively high number of examples are associated with infrequent label sets. When each label is considered separately, label skew becomes *class imbalance*. In this case, not only may some labels be more frequent than others (inter class), but a strong imbalance between positive and negative examples for each label (inner or intra class) may also occur.[254] Measures regarding imbalance in MLL are found in Refs 263 and 264. The proposals developed till now are sparse and have been focused on adaptations of MLL algorithms, the use of ensembles of classifiers or preprocessing methods.[263–265]

PITFALLS AND GUIDELINES

Given the high number of available algorithms, selecting the most suitable set of learners to be applied to a given data set is an important issue. This section intends to give some guidelines in the light of the results reported in literature.

Despite that some empirical evaluations have been carried out in concrete domains as image annotation,[266] video annotation,[267] or within a concrete family of algorithms,[126] some authors recently suggested that the development of empirical comparisons needs to be more explored.[1] In our opinion, these comparisons should take into account a wide range of data sets, algorithms, and performance metrics, conduct statistical tests to determine significant differences between proposals and consider execution time in training and test stages.

In Ref 13, an extensive experimental comparison with statistical proofs was conducted on 12 state-of-the-art MLL algorithms, 16 evaluation metrics, and 11 benchmark data sets. The overall conclusion was that the best performing methods were RF-PCT, HOMER, BR, and CC and thus, they were recommended as good methods for MLL and as benchmarks for testing new algorithms. Particularly, for each measure, the best algorithm was RF-PCT followed by HOMER. This last method was specially recommended for large data sets.[13,210] ML-kNN performed poor across all evaluation measures, but in spite of its limitations, it resulted more efficient than ECC for large training sets. More specifically, in example-based and label-based measures RF-PCT was best according to precision (the prediction was more exact) and HOMER was better in recall and poor in precision (the prediction was more complete). With ranking-based measures RF-PCT was the best followed by CC and BR while HOMER performance was poor. Regarding base classifiers, SVMs and trees were used. Experiments lead to think that SVMs work better in domains with large number of features as text classification, typically $d > 500$, and small number of patterns, while trees perform better in domains with large number of examples. This may be due to the fact that SMVs are able to exploit the information of all features while trees only exploit a subset.

In Ref 97, the experiments carried out yielded interesting conclusions about complexity and limits of the algorithms that can help researchers to choose the proper method. First, CC and BR time complexity was similar (up to $q > 128$) and overall, were considered the best candidates for very large problems. RA*k*EL run out of memory when $q > 256$. CLR scaled well with respect to the number of patterns but not with respect to the number of labels (it became intractable for $q > 64$) and vice versa for IBLR. CC, BR, and CLR were able to complete the task for 819,200 patterns.

Chekina et al.[268] postulated that additional information (besides the target evaluation measure)

was needed to select the appropriate MLL method and contemplated a meta-learning approach based on 49 descriptive characteristics of the data sets (e.g., number of instances, number of labels, number of distinct labels in the data set, etc.). A set of 10 algorithms, 12 data sets, and 18 evaluation measures were used. Almost in two-thirds of the cases the meta-learner was able to predict which MLL method outperformed other methods. Besides, in the light of experiments, authors suggested a set of meta-features of data sets that may determine the performance of the MLL methods (e.g., number of labels, average examples per class, etc.).

In Refs 193 and 269, conditional and unconditional dependences were studied (see definitions in *Label Dependence* section) obtaining interesting findings that are listed below:

- Most MLL algorithms learn by explicitly or implicitly optimize a specific metric (the main metrics are summarized in Table 1). The same MLL method rarely will be optimal for different types of losses. Depending on the metric being minimized conditional dependence treatment can improve or not the performance.
- Minimizing the subset 0/1 loss requires modeling dependences between labels. A classifier that minimizes the subset 0/1 loss may cause low values for the Hamming loss and vice versa.
- The conditional dependence is more related to nondecomposable losses (e.g., subset 0/1 loss) than to decomposable ones (e.g., Hamming loss).

A general conclusion is that algorithms perform different according to the data sets and evaluation metrics, and the algorithm to use will depend on the needs of the problem. A guideline could be decision trees for efficiency, ensembles for predictive performance and transformation methods for the flexibility of using any single-label classier.[270] Besides, other features as scalability (in patterns, features, or labels) and interpretability of the model could be taken into account.

CONCLUDING REMARKS

This study has carried out a review of the state-of-the-art in MLL and ongoing research. The descriptions of the multi-label framework and the main areas of application have provided us with the background needed to understand the works reviewed. The review shows that MLL has been successfully applied to fields such as text, image and video annotation, detection of emotions in music, medical diagnosis, gene and protein function prediction, and even new areas of application are arising (e.g., speech emotion recognition, or social network mining). A bibliometric study has revealed the increasing interest and number of papers that have been published in this field. The more accepted taxonomy of MLL methods has been used to categorize, sort, and describe the relevant literature showing that many of the main traditional single-label classification models have been adapted to the MLL framework. Guidelines and pitfalls to choose the proper method have also been provided. The key challenge and open issue has been recently identified as dealing with the high dimensionality of the output space, especially in domains with a large number of labels. This challenge involves exploring label correlations efficiently. The study of recent work also reveals that there exist many challenges that could merit further attention in the future, such as dimensionality reduction, MIML learning, semi-supervised and active learning, structured prediction or managing imbalanced data.

NOTE

[a] We will use ECOC instead ECC in order to avoid confusion with Ensemble of Classifier Chains.

ACKNOWLEDGMENT

This work was supported by the Ministry of Science and Technology project TIN-2011-22408.

REFERENCES

1. Zhang ML, Zhou ZH. A Review On Multi-Label Learning Algorithms. *IEEE Trans Knowl Data Eng* 2014, 26:1819–1837.
2. First International Workshop on Learning from Multi-Label Data (MLD'09). Available at: http://lpis.csd.auth.gr/workshops/mld09/mld09.pdf. (2009).

3. Second International Workshop on Learning from Multi-Label Data (MLD'10). http://cse.seu.edu.cn/conf/MLD10/files/MLD'10.pdf (2010).
4. Extreme Classification: Multi-Class & Multi-Label Learning with Millions of Categories. Available at: http://nips.cc/Conferences/2013/Program/event.php?ID=3707 (2013).
5. Special issue on learning from multi-label data. *Mach Learn* 2012, 88.
6. LAMDA: Learning and Mining from Data. Data & Code. Available at: http://lamda.nju.edu.cn/Data.ashx.
7. Chang CC, Lin CJ. LIBSVM: a library for support vector machines. *ACM Trans Intell Syst Technol* 2, 27:1–27:27 (2011). Available at: http://www.csie.ntu.edu.tw/cjlin/libsvm.
8. Read J. MEKA: a multi-label extension to WEKA. Available at: http://meka.sourceforge.net/. (2012).
9. Tsoumakas G, Spyromitros-Xioufis E, Vilcek J, Vlahavas I. Mulan: a java library for multi-label learning. *J Mach Learn Res* 2011, 12:2411–2414.
10. de Carvalho A, Freitas A. A tutorial on multi-label classification techniques. In: *Foundations of Computational Intelligence*, vol. 5, Berlin/Heidelberg: Springer; 2009, 177–195.
11. Tsoumakas G, Katakis I. Multi label classification: an overview. *Int J Data Warehousing Min* 2007, 3:1–13.
12. Tsoumakas G, Katakis I, Vlahavas I. Mining multi-label data. In: *Data Mining and Knowledge Discovery Handbook, Part 6*. Springer; 2010, 667–685.
13. Madjarov G, Kocev D, Gjorgjevikj D, Džeroski S. An extensive experimental comparison of methods for multi-label learning. *Pattern Recogn* 2012, 45:3084–3104.
14. Loza E, Fürnkranz J. 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-2008)*, Springer-Verlag; 2008, 50–65.
15. Loza E, Fürnkranz J. 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, Berlin/Heidelberg: Springer; 2010, 192–215.
16. Rubin T, Chambers A, Smyth P, Steyvers M. Statistical topic models for multi-label document classification. *Mach Learn* 2012, 88:157–208.
17. Ueda N, Saito K. Parametric mixture models for multi-labeled text. In: *Proceedings on Neural Information Processing Systems (NIPS)*; 2002, 721–728.
18. Schapire RE, Singer Y. BoosTexter: a boosting-based system for text categorization. *Mach Learn* 2000, 39:135–168.
19. Nguyen CD, Dung TA, Cao TH. Text classification for DAG-structured categories. In: *Advances in Knowledge Discovery and Data Mining*, Lecture Notes in Computer Science, vol. 3518, chap. 36, Berlin/Heidelberg: Springer; 2005; 1–18.
20. Spat S, Cadonna B, Rakovac I, Gütl C, Leitner H, Stark G, Beck P. Enhanced information retrieval from narrative german-language clinical text documents using automated document classification. In: *eHealth Beyond the Horizon—Get IT There, Proceedings of MIE2008, The XXIst International Congress of the European Federation for Medical Informatics*, Göteborg, Sweden; 2008, 473–478.
21. Cong H, Tong LH. Grouping of TRIZ inventive principles to facilitate automatic patent classification. *Expert Systems with Applications* 2008, 34:788–795.
22. Oza N, Castle JP, Stutz J. Classification of aeronautics system health and safety documents. *IEEE Trans Syst Man Cybern C Appl Rev* 2009, 39:670–680.
23. Lauser B, Hotho A. Automatic multi-label subject indexing in a multilingual environment. In: *European Conference on Digital Libraries (ECDL)*, Lecture Notes in Computer Science, vol. 2769; 2003, 140–151.
24. Katakis I, Tsoumakas G, Vlahavas I. Multilabel text classification for automated tag suggestion. In: *Proceedings of the ECML/PKDD 2008 Discovery Challenge*; 2008.
25. Song Y, Zhang L, Giles CL. Automatic tag recommendation algorithms for social recommender systems. *ACM Trans Web* 2011, 5:1–31.
26. Yearwood J, Mammadov M, Banerjee A. Profiling phishing emails based on hyperlink information. In: *International Conference on Advances in Social Networks Analysis and Mining*; 2010, 120–127.
27. Yan Y, Fung G, Dy JG, Rosales R. Medical coding classification by leveraging inter-code relationships. In: *Proceedings of the 16th International Conference on Knowledge Discovery and Data Mining (KDD '10)*, New York, NY, USA; 2010, 193–202.
28. Tang L, Rajan S, Narayanan VK. Large scale multi-label classification via metalabeler. In: *Proceedings of the 18th International Conference on World Wide Web (WWW '09)*, New York, NY, USA; 2009, 211–220.
29. Bhowmick PK, Basu A, Mitra P. Reader perspective emotion analysis in text through ensemble based multi-label classification framework. *Comput Inf Sci* 2009, 2:64–74.
30. Bhowmick PK, Basu A, Mitra P, Prasad A. Sentence level news emotion analysis in fuzzy multi-label classification framework (special issue on natural language processing and its applications). *Res Comput Sci* 2010, 46:143–154.
31. Nasierding G, Kouzani A. Image to text translation by multi-label classification. In: *Advanced Intelligent Computing Theories and Applications with Aspects*

of Artificial Intelligence, Lecture Notes in Computer Science, vol. 6216, Berlin/Heidelberg: Springer; 2010, 247–254.

32. Wang M, Zhou X, Chua TS. Automatic image annotation via local multi-label classification. In: *Proceedings of the 2008 International Conference on Content-Based Image and Video Retrieval (CIVR '08)*, New York, NY, USA; 2008, 17–26.
33. Kumar N, Berg AC, Belhumeur PN, Nayar SK. Attribute and simile classifiers for face verification. In: *IEEE International Conference on Computer Vision (ICCV)*; 2009.
34. Wang J, Zhao Y, Wu X, Hua XS. A transductive multi-label learning approach for video concept detection. *Pattern Recogn* 2010, 44:2274–2286.
35. Shotton J, Winn J, Rother C, Criminisi A. TextonBoost for image understanding: multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *Int J Comput Vision* 2009, 81:2–23.
36. Ma A, Sethi I, Patel N. Multimedia content tagging using multilabel decision tree. In: *11th IEEE International Symposium on Multimedia (ISM '09)*; 2009, 606–611.
37. Trohidis K, Tsoumakas G, Kalliris G, Vlahavas I. Multi-label classification of music into emotions. In: *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR 2008)*; 2008, 325–330.
38. Pachet F, Roy P. Improving multilabel analysis of music titles: a large-scale validation of the correction approach. *IEEE Trans Audio Speech Lang Proc* 2009, 17:335–343.
39. Sobol-Shikler T, Robinson P. Classification of complex information: inference of co-occurring affective states from their expressions in speech. *IEEE Trans Pattern Anal Mach Intell* 2010, 32:1284–1297.
40. Barutcuoglu Z, Schapire RE, Troyanskaya OG. Hierarchical multi-label prediction of gene function. *Bioinformatics* 2006, 22:830–836.
41. Cesa-Bianchi N, Re M, Valentini G. Synergy of multi-label hierarchical ensembles, data fusion, and cost-sensitive methods for gene functional inference. *Mach Learn* 2012, 88:209–241.
42. Cesa-Bianchi N, Valentini G. Hierarchical cost-sensitive algorithms for genome-wide gene function prediction. *J Mach Learn Res* 2010, 8:14–29.
43. Cheng W, Dembczyński K, Hüllermeier E. Graded multilabel classification: the ordinal case. In: *Proceedings of the 27 th International Conference on Machine Learning (ICML)*; 2010, 223–230.
44. Elisseeff A, Weston J. A kernel method for multi-labelled classification. In: *Advances in Neural Information Processing Systems (NIPS)*, vol. 14; 2001, 681–687.
45. Valentini G, Re M. Weighted true path rule: a multilabel hierarchical algorithm for gene function prediction. In: *Proceedings of the 1st International Workshop on Learning from Multi-Label Data (MLD-ECML 2009)*, Bled, Slovenia; 2009, 132–145.
46. Zhang ML, Zhou ZH. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Trans Knowl Data Eng* 2006, 18:1338–1351.
47. Alves RT, Delgado MR, Freitas AA. Multi-label hierarchical classification of protein functions with artificial immune systems. In: *Proceedings of the Brazilian Symposium in Bioinformatics (BSB-2008)*, Lecture Notes in Bioinformatics, vol. 5167; 2008, 1–12.
48. Alves RT, Delgado MR, Freitas AA. Knowledge discovery with artificial immune systems for hierarchical multi-label classification of protein functions. In: *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Barcelona, Spain; 2010, 1–8.
49. Chan A, Freitas AA. A new ant colony algorithm for multi-label classification with applications in bioinfomatics. In: *GECCO '06: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, New York, USA; 2006, 27–34.
50. Otero F, Freitas A, Johnson C. A hierarchical multi-label classification ant colony algorithm for protein function prediction. *Memetic Comput* 2010, 2:165–181.
51. Duwairi R, Kassawneh A. A framework for predicting proteins 3D structures. In: *IEEE/ACS International Conference on Computer Systems and Applications (AICCSA '08)*, Washington, DC, USA; 2008, 37–44.
52. Chou KC, Wu ZC, Xiao X. iLoc-Euk: a multi-label classifier for predicting the subcellular localization of singleplex and multiplex eukaryotic proteins. *PLoS ONE* 2011, 6.
53. Mammadov MA, Rubinov AM, Yearwood J. The study of drug-reaction relationships using global optimization techniques. *Optim Method Softw* 2007, 22:99–126.
54. Kawai K, Takahashi Y. Identification of the dual action antihypertensive drugs using TFS-based support vector machines. *Chem-Bio Inf J* 2009, 4:44–51.
55. Ukwatta E, Samarabandu J. Vision based metal spectral analysis using multi-label classification. In: *Canadian Conference on Computer and Robot Vision (CRV '09)*; 2009, 132–139.
56. Tang L, Liu H. Relational learning via latent social dimensions. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*, New York, NY, USA; 2009, 817–826.
57. Tang L, Liu H. Scalable learning of collective behavior based on sparse social dimensions. In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM '09)*, New York, NY, USA; 2009, 1107–1116.

58. Krohn-Grimberghe A, Drumond L, Freudenthaler C, Schmidt-Thieme L. Multi-relational matrix factorization using Bayesian personalized ranking for social network data. In: *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining (WSDM '12)*, New York, NY, USA; 2012, 173–182.
59. Peters S, Denoyer L, Gallinari P. Iterative annotation of multi-relational social networks. In: *International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*; 2010, 96–103.
60. Özpolat E, Akar GB. Automatic detection of learning styles for an e-learning system. *Comput Educ* 2009, 53:355–367.
61. López VF, de la Prieta F, Ogihara M, Wong DD. A model for multi-label classification and ranking of learning objects. *Expert Syst Appl* 2012, 39: 8878–8884.
62. Zhang Y, Burer S, Street WN, Bennett K, Parrado-hern E. Ensemble pruning via semi-definite programming. *J Mach Learn Res* 2006, 7:1315–1338.
63. Shao H, Li G, Liu G, Wang Y. Symptom selection for multi-label data of inquiry diagnosis in traditional Chinese medicine. *Sci China Ser F-Info Sci* 2010, 1:1–13.
64. Abbas Q, Celebi M, Serrano C, García IF, Ma G. Pattern classification of dermoscopy images: a perceptually uniform model. *Pattern Recogn* 2013, 46:86–97.
65. Zhang ML, Zhou ZH. A k-nearest neighbor based algorithm for multi-label classification. In: *Proceedings of the IEEE International Conference on Granular Computing (GrC)*, Beijing, China; 2005, 718–721.
66. Fan RE, Lin CJ. A study on threshold selection for multi-label classification. Technical Report, National Taiwan University; 2007.
67. Ioannou M, Sakkas G, Tsoumakas G, Vlahavas IP. Obtaining bipartitions from score vectors for multi-label classification. In: *22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*; 2010, 409–416.
68. Montejo-Ráez A, Ureña López L. Selection strategies for multi-label text categorization.In: *Advances in Natural Language Processing*, Lecture Notes in Computer Science, vol. 4139; 2006, 585–592.
69. Yang Y. 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)*, New York, NY, USA; 2001, 137–145.
70. Bielza C, Li G, Larrañaga P. Multi-dimensional classification with Bayesian networks. *Int J Approx Reasoning* 2011, 52:705–727.
71. Liu G, Lin Z, Yu Y. Multi-output regression on the output manifold. *Pattern Recogn* 2009, 42: 2737–2743.
72. Ghamrawi N, McCallum A. Collective multi-label classification. In: *CIKM '05: Proceedings of the 14th ACM International Conference on Information and Knowledge Management*; 2005, 195–200.
73. Dembczyński K, Waegeman W, Cheng W, Hüllermeier E. Regret analysis for performance metrics in multi-label classification: the case of hamming and subset zero-one loss. In: *Machine Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science, vol. 6321. Berlin/Heidelberg: Springer; 2010, 280–295.
74. Schapire RE, Singer Y. Improved boosting algorithms using confidence-rated predictions. *Mach Learn* 1999, 37:297–336.
75. Godbole S, Sarawagi S. Discriminative methods for multi-labeled classification. In: *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*; 2004, 22–30.
76. Crammer K, Singer Y. A family of additive online algorithms for category ranking. *J Mach Learn Res* 2003, 3:1025–1058.
77. Park SH, Fürnkranz J. Multi-label classification with label constraints. Technical Report, TUD-KE-2008-04, Knowledge Engineering Group, TU Darmstadt; 2008. Available at: http://www.ke.tu-darmstadt.de/publications/reports/tud-ke-2008-04.pdf.
78. Read J. Scalable multi-label classification. PhD Thesis, University of Waikato, 2010.
79. Bianchi NC, Gentile C, Zaniboni L. Incremental algorithms for hierarchical classification. *J Mach Learn Res* 2006, 7:31–54.
80. Menca EL. Multi-label classification in parallel tasks. In: *2nd International Workshop on Learning from Multi-Label Data (MLD'10)*; 2010, 29–36.
81. Nguyen N, Caruana R. Classification with partial labels. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*, New York, NY, USA; 2008, 551–559.
82. Zafra A, Gibaja E, Ventura S. Multiple instance learning with multiple objective genetic programming for web mining. *Appl Soft Comput* 2011, 11:93–102.
83. Petterson J, Caetano TS. Reverse multi-label learning. In: *Advances in Neural Information Processing Systems (NIPS)*; 2010, 1912–1920.
84. Aiolli F, Cardin R, Sebastiani F, Sperduti A. Preferential text classification: learning algorithms and evaluation measures. *Inf Retr* 2009, 12:559–580.
85. Bucak S, Jin R, Jain A. Multi-label learning with incomplete class assignments. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; 2011, 2801 –2808.
86. Sun YY, Zhang Y, Zhou ZH. Multi-label learning with weak label. In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence; 2010, 593–598.

87. Li H, Guo YJ, Wu M, Li P, Xiang Y. Combine multi-valued attribute decomposition with multi-label learning. *Expert Syst Appl* 2010, 37:8721–8728.
88. Blum A, Mitchell T. Combining labeled and unlabeled data with co-training. In: *11th Annual Conference on Computational Learning Theory*, Madison, WI; 1998, 92–100.
89. Lewis DD, Yang Y, Rose TG, Li F. RCV1: a new benchmark collection for text categorization research. *J Mach Learn Res* 2005, 5:361–397.
90. Boutell M, Luo J, Shen X, Brown C. Learning multi-label scene classification. *Pattern Recogn* 2004, 37:1757–1771.
91. Zhang ML, Zhou ZH. Multi-label learning by instance differentiation. In: *AAAI Conference on Artificial Intelligence*, 2007, 669–674.
92. Huang GB, Ding X, Zhou H. Optimization method based extreme learning machine for classification. *Neurocomputing* 2010, 74:155–163.
93. Xiao X, Wu ZC, Chou KC. iLoc-Virus: a multi-label learning classifier for identifying the subcellular localization of virus proteins with both single and multiple sites. *J Theor Biol* 2011, 284:42–51.
94. Vens C, Struyf J, Schietgat L, Džeroski S, Blockeel H. Decision trees for hierarchical multi-label classification. *Mach Learn* 2008, 73:185–214.
95. Schapire RE, Singer Y. Improved boosting algorithms using confidence-rated predictions. In: *Proceedings of the Eleventh Annual Conference on Computational Learning Theory (COLT' 98)*, New York, NY, USA; 1998, 80–91.
96. Turnbull D, Barrington L, Torres D, Lanckriet G. Semantic annotation and retrieval of music and sound effects. *IEEE Trans Audio Speech Lang Proc* 2008, 16:467–476.
97. Read J, Pfahringer B, Holmes G, Frank E. Classifier chains for multi-label classification. *Mach Learn* 2011, 85:1–27.
98. Dembczyński K, Cheng W, Hüllermeier E. Bayes optimal multilabel classification via probabilistic classifier chains. In: *Proceedings of the 27th International Conference on Machine Learning (ICML)*; 2010, 279–286.
99. Antenreiter M, Ortner R, Auer P. 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*, Bled, Slovenia; 2009, 16–27.
100. Tsoumakas G, Dimou A, Spyromitros E, Mezaris V, Kompatsiaris I, Vlahavas I. 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)*, Bled, Slovenia; 2009, 101–116.
101. Cherman EA, Metz J, Monard MC. Incorporating label dependency into the binary relevance framework for multi-label classification. *Expert Syst Appl* 2012, 39:1647–1655.
102. Hüllermeier E, Fürnkranz J, Cheng W, Brinker K. Label ranking by learning pairwise preferences. *Artif Intell* 2008, 172:1897–1916.
103. Brinker K, Fürnkranz J, Hüllermeier E. A unified model for multilabel classification and ranking. In: *Proceeding of the ECAI 2006: 17th European Conference on Artificial Intelligence*; 2006, 489–493.
104. Fürnkranz J, Hüllermeier E, Loza menca E, Brinke K. Multilabel classification via calibrated label ranking. *Mach Learn* 2008, 73:133–153.
105. Loza E, Park SH, Fürnkranz J. Efficient voting prediction for pairwise multilabel classification. *Neurocomputing* 2010, 73:1164–1176.
106. Madjarov G, Gjorgjevikj D, Džeroski S. Dual layer voting method for efficient multi-label classification. In: *Proceedings of the 5th Iberian Conference on Pattern Recognition and Image Analysis*, Lecture Notes in Computer Science, vol. 6669; 2011, 232–239.
107. Read J. A pruned problem transformation method for multi-label classification. In: *Proceedings of the NZ Computer Science Research Student Conference*; 2008, 143–150.
108. Tenenboim L, Rokach L, Shapira B. Identification of label dependencies for multi-label classification. In: *2nd International Workshop on Learning from Multi-Label Data (MLD'10)*; 2010, 53–60.
109. Tsoumakas G, Katakis I, Vlahavas I. Random k-Labelsets for Multi-Label Classification. *IEEE Trans Knowl Data Eng* 2010, 23:1079–1089.
110. Lo H, Wang J, Wang H, Lin S. Cost-sensitive multi-label learning for audio tag annotation and retrieval. *IEEE Trans Multimedia* 2011, 13:518–529.
111. Rokach L, Schclar A, Itach E. Ensemble methods for multi-label classification. *Expert Syst Appl* 2014, 41:7507–7523.
112. Kocev D, Vens C, Struyf J, Džeroski S. Ensembles of multi-objective decision trees. In: *Proceedings of the 18th European Conference on Machine Learning (ECML '07)*, Berlin/Heidelberg: Springer; 2007, 624–631.
113. Nasierding G, Kouzani AZ, Tsoumakas G. A triple-random ensemble classification method for mining multi-label data. In: *Proceedings of the 2010 IEEE International Conference on Data Mining Workshops (ICDMW '10)*, Washington, DC, USA; 2010, 49–56.
114. Sarinnapakorn K, Kubat M. Induction from multi-label examples in information retrieval systems: a case study. *Appl Artif Intell* 2008, 22:407–432.
115. Xu J. Constructing a fast algorithm for multi-label classification with support vector data description. In: *Proceedings of the IEEE International Conference on Granular Computing (GrC)*; 2010, 817–821.

116. Clare A, King RD. Knowledge discovery in multi-label phenotype data. In: *PKDD '01 Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery*, Lecture Notes in Computer Science, vol. 2168; 2001; 42–53.
117. Noh HG, Song MS, Park SH. An unbiased method for constructing multilabel classification trees. *Comput Stat Data Anal* 2004, 47:149–164.
118. Blockeel H, Raedt LD, Ramon J. Top-down induction of clustering trees. In: *Proceedings of the Fifteenth International Conference on Machine Learning (ICML '98)*, San Francisco, CA, USA; 1998, 55–63.
119. Petrovskiy M. Paired comparisons method for solving multi-label learning problem. In: *Sixth International Conference on Hybrid Intelligent Systems (HIS '06)*; 2006, 42.
120. Xu J. An efficient multi-label support vector machine with a zero label. *Expert Syst Appl* 2012, 39:4796–4804.
121. Wang L, Chang M, Feng J. Parallel and sequential support vectormachines for multi-label classification. *Int J Inf Technol* 2005, 11:11–18.
122. Wan SP, Xu JH. A multi-label classification algorithm based on triple class support vector machine. In: *International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR '07)*, vol. 4; 2007, 1447–1452.
123. Li J, Xu J. A fast multi-label classification algorithm based on double label support vector machine. In: *Proceedings of the 2009 International Conference on Computational Intelligence and Security (CIS '09)*, Washington, DC, USA; 2009, 30–35.
124. Jiang A, Wang C, Zhu Y. Calibrated Rank-SVM for multi-label image categorization. In: *IEEE World Congress on Computational Intelligence (IJCNN)*; 2008, 1450–1455.
125. Cheng W, Hüllermeier E. Combining instance-based learning and logistic regression for multilabel classification. *Mach Learn* 2009, 76:211–225.
126. Spyromitros E, Tsoumakas G, Vlahavas I. An empirical study of lazy multilabel classification algorithms. In: *SETN '08: Proceedings of the 5th Hellenic Conference on Artificial Intelligence*, Berlin, Heidelberg; 2008, 401–406.
127. Younes Z, Aballah F, Denoeux T. Multi-label classification algorithm derived from k-nearest neighbor rule with label dependencies. In: *Proceedings of the 16th European Signal Processing Conference*; 2008.
128. Brinker K, Hüllermeier E. Case-based multilabel ranking. In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, San Francisco, CA, USA; 2007, 702–707.
129. Younes Z, Abdallah F, Denoux T. Fuzzy multi-label learning under veristic variables. In: *IEEE International Conference on Fuzzy Systems, FUZZ-IEEE* 2010, 1–8.
130. Younes Z, Abdallah F, Denœux T. Evidential multi-label classification approach to learning from data with imprecise labels. In: *Computational Intelligence for Knowledge-Based Systems Design*, Lecture Notes in Computer Science, vol. 6178, Berlin/Heidelberg: Springer; 2010, 119–128.
131. Jiang JY, Tsai SC, Lee SJ. FSKNN: multi-label text categorization based on fuzzy similarity and k nearest neighbors. *Expert Syst Appl* 2012, 39:2813–2821.
132. Lin X, Chen XW. Mr.KNN: soft relevance for multi-label classification. In: *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM '10)*, New York, NY, USA; 2010, 349–358.
133. Zhang ML. Ml-rbf: RBF neural networks for multi-label learning. *Neural Proc Lett* 2009, 29:61–74.
134. Ciarelli PM, Oliveira E, Badue C, Souza AF. Multi-label text categorization using a probabilistic neural network. *Int J Comput Inf Syst Ind Manage Appl* 2009, 1:133–144.
135. Ciarelli P, Oliveira E. An enhanced probabilistic neural network approach applied to text classification. In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, Lecture Notes in Computer Science, vol. 5856, chap. 78. Berlin/Heidelberg: Springer; 2009, 661–668.
136. Sapozhnikova E. ART-based neural networks for multi-label classification. In: *Advances in Intelligent Data Analysis VIII*, Lecture Notes in Computer Science, vol. 5772, Berlin/Heidelberg: Springer; 2009, 167–177.
137. McCallum AK. Multi-label text classification with a mixture model trained by EM. In: *AAAI 99 Workshop on Text Learning*; 1999.
138. Kaneda Y, Ueda N, Saito K. Extended parametric mixture model for robust multi-labeled text categorization. In: *Knowledge-Based Intelligent Information and Engineering Systems*, Lecture Notes in Computer Science, vol. 3214; 2004, 616–623.
139. Wang H, Huang M, Zhu X. A generative probabilistic model for multi-label classification. In: *ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, Washington, DC, USA; 2008, 628–637.
140. Streich A, Buhmann J. Classification of multi-labeled data: a generative approach. In: *Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD '08)*, Springer-Verlag; 2008, 390–405.
141. Thabtah FA, Cowling P, Peng Y, Rastogi R, Morik K, Bramer M, Wu X. MMAC: a new multi-class, multi-label associative classification approach. In: *Proceedings of the Fourth IEEE International Conference on Data Mining, ICDM 2004*; 2004, 217–224.

142. Thabtah FA, Cowling PI. A greedy classification algorithm based on association rule. *Appl Soft Comput* 2007, 7:1102–1111.

143. Rak R, Kurgan L, Reformat M. A tree-projection-based algorithm for multi-label recurrent-item associative-classification rule generation. *Data Knowl Eng* 2008, 64:171–197.

144. Veloso A, Meira W Jr, Gonçalves MA, Zaki MJ. Multi-label lazy associative classification. In: *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, Warsaw, Poland; 2007, 605–612.

145. Cano A, Zafra A, Galindo ELG, Ventura S. A grammar-guided genetic programming algorithm for multi-label classification. In: *16th European Conference, EuroGP*, Lecture Notes in Computer Science, vol. 7831; 2013, 217–228.

146. Ávila J, Gibaja E, Ventura S. Evolving multi-label classification rules with gene expression programming: a preliminary study. In: *Hybrid Artificial Intelligence Systems (HAIS)*, Lecture Notes in Computer Science, vol. 6077; 2010, 9–16.

147. Ávila JL, Gibaja EL, Zafra A, Ventura S. A gene expression programming algorithm for multi-label classification. *J Mult-Valued Log S* 2011, 17:183–206.

148. Xu H, Xu J. Designing a multi-label kernel machine with two-objective optimization. In: *Proceedings of the 2010 International Conference on Artificial Intelligence and Computational Intelligence (AICI'10): Part I*, Berlin, Heidelberg; 2010, 282–291.

149. Shi C, Kong X, Yu P, Wang B. Multi-objective multi-label classification. In: *Proceedings of the SIAM International Conference on Data Mining*, Anaheim, CA, USA; 2012, 355–366.

150. Shi C, Kong X, Yu PS, Wang B. Multi-label ensemble learning. In: *European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*; 2011, 223–239.

151. Gonçalves EC, Plastino A, Freitas AA. A genetic algorithm for optimizing the label ordering in multi-label classifier chains. In: *IEEE 25th International Conference on Tools with Artificial Intelligence (ICTAI)*; 2013, 469–476.

152. Sebastiani F, Sperduti A, Valdambrini N. An improved boosting algorithm and its application to text categorization. In: *Proceedings of the Ninth International Conference on Information and Knowledge Management (CIKM '00)*, New York, NY, USA; 2000, 78–85.

153. Nardiello P, Sebastiani F, Sperduti A. Discretizing continuous attributes in adaboost for text categorization. In: *Advances in Information Retrieval*, Lecture Notes in Computer Science, vol. 2633, Berlin/Heidelberg: Springer; 2003, 320–334.

154. De Comité F, Gilleron R, Tommasi M. 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)*, Berlin/Heidelberg: Springer; 2003, 35–49.

155. Diao L, Hu K, Lu Y, Shi C. Boosting simple decision trees with Bayesian learning for text categorization. In: *Proceedings of the 4th World Congress on Intelligent and Automation*, Shanghai, China; 2002, 321–325.

156. Johnson M, Cipolla R. Improved image annotation and labelling through multi-label boosting. In: *British Machine Vision Association (BMVC)*; 2005.

157. Esuli A, Fagni T, Sebastiani F. MP-Boost: a multiple-pivot boosting algorithm and its application to text categorization. In: *String Processing and Information Retrieval (SPIRE)*, Lecture Notes in Computer Science, vol. 4209. Berlin/Heidelberg: Springer; 2006, 1–12.

158. Yan R, Tesic J, Smith JR. Model-shared subspace boosting for multi-label classification. In: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '07)*, New York, NY, USA; 2007, 834–843.

159. Zhang X, Yuan Q, Zhao S, Fan W, Zheng W, Wang Z. Multi-label classification without the multi-label cost. In: *Proceedings of the 10th SIAM International Conference on Data Mining*; 2010.

160. Zhang Y, Schneider J. Multi-label output codes using canonical correlation analysis. In: *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*; 2011, 873–882.

161. Kouzani A. Multilabel classification using error correction codes. In: *Advances in Computation and Intelligence*, Lecture Notes in Computer Science, vol. 6382; 2010, 444–454.

162. Kajdanowicz T, Wozniak M, Kazienko P. Multiple classifier method for structured output prediction based on error correcting output codes. In: *Intelligent Information and Database Systems*, Lecture Notes in Computer Science, vol. 6592; 2011, 333–342.

163. Vateekul P, Kubat M. Fast induction of multiple decision trees in text categorization from large scale, imbalanced, and multi-label data. In: *Proccedings of the 2009 IEEE International Conference on Data Mining Workshops (ICDMW)*; 2009, 320–325.

164. Zhou T, Tao D, Wu X. Compressed labeling on distilled labelsets for multi-label learning. *Mach Learn* 2012, 88:69–126.

165. Wolpert DH. Stacked generalization. *Neural Networks* 1992, 5:241–259.

166. Rokach L, Itach E. An ensemble method for multi-label classification using a transportation model. In: *Proceedings of the 1st Workshop on Learning from Multilabel Data (MLD) Held in Conjunction with ECML/PKDD*, Bled, Slovenia; 2009, 49–60.

167. Rokach L, Itach E. An ensemble method for multi-label classification using an approximation algorithm for

the set covering problem. In: *Proceedings of the 2nd International Workshop on Learning from Multilabel Data (MLD)*, Haifa, Israel; 2010, 37–44.

168. Tax D, Duan RPW. Support vector data description. *Mach Learn* 2004, 54:45–66.
169. Shafer G. *A Mathematical Theory of Evidence*. Princeton University Press; 1976.
170. Nettleton D, Banerjee T. Testing the equality of distributions of random vectors with categorical components. *Comput Stat Data Anal* 2001, 37:195–208.
171. Gonçalves T, Quaresma P. A preliminary approach to the multilabel classification problem of portuguese juridical documents. *Prog Artif Intell, Lect Notes Comput Sci* 2003, 2902:435–444.
172. Gonçalves T, Quaresma P. The impact of NLP techniques in the multilabel text classification problem. In: *Proceedings of Intelligent Information Processing and Web Mining (IIPWM'04), Advances in Soft Computing*; 2004, 424–428.
173. Li X, Wang L, Sung E. Multilabel SVM active learning for image classification. In: *International Conference on Image Processing (ICIP '04)*; 2004, 2207–2210.
174. Rao P, Kupper L. Ties in paired-comparison experiments: a generalization of the bradley-terry model. *Am Stat Assoc* 1967, 62:194–204.
175. Keller JM, Gray MR, Givens JA. Fuzzy K-Nearest neighbor algorithm. *IEEE Trans Syst Man Cybern* 1985, 15:580–585.
176. Yager RR. Veristic variables. *IEEE Trans Syst Man Cybern* 2000, 30:71–84.
177. Specht DF. Probabilistic neural networks. *Neural Netw* 1990, 3:109–118.
178. Dempster AP, Laird NM. Rubin., D.B.: Maximum likelihood from incomplete data via the EM algorithm. *J R Stat Soc B* 1977, 39:1–38.
179. Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation. *J Mach Learn Res* 2003, 3.
180. Lafferty J, McCallum A, Pereira F. Conditional random fields: probabilistic models for segmenting and labelling sequence data. In: *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*; 2001, 282–289.
181. Wang X, Liu X, Shi Z, Shi Z, Sui H. Voting conditional random fields for multi-label image classification. In: *3rd International Congress on Image and Signal Processing (CISP)*; 2010, 1984–1988.
182. Agarwal R, Aggarwal C, Prasad V. A tree projection algorithm for generation of frequent item sets. *J Parallel Distr Com* 2001, 61:350–371.
183. Parpinelli R, Lopes H, Freitas A. Data mining with an ant colony optimization algorithm. *IEEE Trans Evol Comput* 2002, 6:321–332.
184. Ferreira C. Gene expression programming: a new adaptive algorithm for solving problems. *Complex Syst* 2001, 13:87–129.
185. Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 2002, 3:182–197.
186. Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. *J Comput Syst Sci* 1997, 55:119–139.
187. Freund Y, Mason L. The alternating decision tree learning algorithm. In: *Proceedings of the Sixteenth International Conference on Machine Learning (ICML '99)*, San Francisco, CA, USA; 1999, 124–133.
188. Fan W, Wang H, Yu PS, Ma S. Is random model better? On its accuracy and efficiency. In: *Proceedings of the Third IEEE International Conference on Data Mining (ICDM '03)*; 2003.
189. Dietterich TG, Bakiri G. Solving multiclass learning problems via error-correcting output codes. *J Artif Intell Res* 1995, 2:263–286.
190. Ferng CS, Lin HT. Multi-label classification with error-correcting codes. *J Mach Learn Res* 2011, 20:281–295.
191. Kajdanowicz T, Kazienko P. Multi-label classification using error correcting output codes. *Int J Appl Math Comput Sci* 2012, 22:829–840.
192. Fürnkranz J, Park SH. Error-correcting output codes as a transformation from multi-class to multi-label prediction. In: *Discovery Science*, Lecture Notes in Computer Science, vol. 7569 Berlin/Heidelberg: Springer; 2012, 254–267.
193. Dembczyński K, Waegeman W, Cheng W, Hüllermeier E. On label dependence in multi-label classification. In: *Proceedings of the 2nd International Workshop on Learning from Multi-Label Data (MLD'10)*; 2010, 5–12.
194. Hsu D, Kakade S, Langford J, Zhang T. Multi-label prediction via compressed sensing. In: *Advances in Neural Information Processing Systems (NIPS)*; 2009, 772–780.
195. Read J, Bifet A, Holmes G, Pfahringer B. Scalable and efficient multi-label classification for evolving data streams. *Mach Learn* 2012, 88:243–272.
196. Zhang ML, Zhang K. Multi-label learning by exploiting label dependency. In: *Proceedings of the 16th International Conference on Knowledge Discovery and Data Mining (KDD '10)*, New York, NY, USA; 2010, 999–1008.
197. Clare A, King RD. Predicting gene function in Saccharomyces cerevisiae. *Bioinformatics* 2003, 2:42–49.
198. Zhang ML, Zhou ZH. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recogn* 2007, 40:2038–2048.
199. Read J, Pfahringer B, Holmes G. Multi-label classification using ensembles of pruned sets. In: *ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, Washington, DC, USA; 2008, 995–1000.

200. Huang SJ, Yu Y, Zhou ZH. Multi-label hypothesis reuse. In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD12)*, Beijing, China; 2012, 525–533.

201. Huang SJ, Zhou ZH. Multi-label learning by exploiting label correlations locally. In: *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*; 2012.

202. Qi GJ, Hua XS, Rui Y, Tang J, Mei T, Zhang HJ. Correlative multi-label video annotation. In: *Proceedings of the 15th International Conference on Multimedia*, New York, NY, USA; 2007, 17–26.

203. Zhu S, Ji X, Xu W, Gong Y. Multi-labelled classification using maximum entropy method. In: *SIGIR '05: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA; 2005, 274–281.

204. Qi GJ, Hua XS, Rui Y, Tang J, Zhang HJ. Two-dimensional active learning for image classification. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)*; 2008, 1–8.

205. Ji S, Tang L, Yu S, Ye J. A shared-subspace learning framework for multi-label classification. *ACM Trans Knowl Discov Data* 2010, 4:1–29.

206. Borchani H, Bielza C, Toro C, Larrañaga P. Predicting human immunodeficiency virus inhibitors using multi-dimensional bayesian network classifiers. *Artif Intell Med* 2013, 57:219–229.

207. van der Gaag LC, de Waal PR. Multi-dimensional Bayesian network classifiers. In: *Third European Workshop on Probabilistic Graphical Models*; 2006, 107–114.

208. Sucar LE, Bielza C, Morales EF, Hernandez-Leal P, Zaragoza JH, Larrañaga P. Multi-label classification with bayesian network-based chain classifiers. *Pattern Recogn Lett* 2014, 41:14–22.

209. Yang Y, Pedersen JO. A comparative study on feature selection in text categorization. In: *Proceedings of the Fourteenth International Conference on Machine Learning (ICML '97)*, San Francisco, CA, USA; 1997, 412–420.

210. Tsoumakas G, Katakis I, Vlahavas I. 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)*, 2008.

211. Doquire G, Verleysen M. Feature selection for multi-label classification problems. In: *11th International Work-Conference on Artificial Neural Networks (IWANN)*, Lecture Notes in Computer Science, vol. 6691; 2011, 9–16.

212. Zhang ML, Peña JM, Robles V. Feature selection for multi-label naive Bayes classification. *Inform Sci* 2009, 179:3218–3229.

213. Jolliffe I. *Principal Component Analysis*. Springer; 1986.

214. Fisher RA. The use of multiple measurements in taxonomic problems. *Ann Eugen* 1936, 7:179–188.

215. Wang H, Ding C, Huang H. Multi-label linear discriminant analysis. In: *Computer Vision—ECCV 2010*, Lecture Notes in Computer Science, vol. 6316, Berlin/Heidelberg: Springer; 2010, 126–139.

216. Park CH, Lee M. On applying linear discriminant analysis for multi-labeled problems. *Pattern Recogn Lett* 2008, 29:878–887.

217. Luo X, Heywood ZAN. Evaluation of two systems on multi-class multi-label document classification. In: *Proceedings of the 15th International Symposium on Methodologies for Intelligent Systems*; 2005, 161–169.

218. Yu K, Yu S, Tresp V. Multi-label informed latent semantic indexing. In: *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA; 2005, 258–265.

219. Zhang Y, Zhou ZH. Multilabel dimensionality reduction via dependence maximization. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 4, paper 14; 2010.

220. Gretton A, Bousquet O, Smola A, Schölkopf B: Measuring statistical dependence with Hilbert-Schmidt norms. In: *Proceedings of the 16th International Conference on Algorithmic Learning Theory (ALT'05)*; 2005, 63–77.

221. Zhang ML. LIFT: multi-label learning with label-specific features. In: *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*; 2011, 1609–1614.

222. Tsoumakas G, Menca EL, Katakis I, Park S, Fürnkrnaz J. On the combination of two decompositive multi-label classification methods. In: *Workshop on Preference Learning, ECML PKDD 09*; 2009, 114–133.

223. Charte F, Rivera A, del Jesus M, Herrera F. Improving multi-label classifiers via label reduction with association rules. In: *Hybrid Artificial Intelligent Systems*, Lecture Notes in Computer Science, vol. 7209. Berlin/Heidelberg: Springer; 2012, 188–199.

224. Han J, Pei J, Yin Y, Mao R. Mining frequent patterns without candidate generation: a frequent-pattern tree approach. *Data Min Knowl Discov* 2004, 8:53–87.

225. Sun L, Ji S, Ye J. Canonical correlation analysis for multilabel classification: a least-squares formulation, extensions, and analysis. *IEEE Trans Pattern Anal Mach Intell* 2011, 33:194–200.

226. Tai F, Lin HT. Multi-label classification with principal label space transformation. In: *2nd International Workshop on Learning from Multi-Label Data (MLD'10)*; 2010; 45–52.

227. Agrawal R, Gupta A, Prabhu Y, Varma M. Multi-label learning with millions of labels: recommending advertiser bid phrases for web pages. In: *Proceedings of the 22nd International Conference on World Wide Web (WWW13)*; 2013, 13–24.
228. Dekel O, Shamir O. Multiclass-multilabel classification with more labels than examples. In: *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*; 2010, 137–144.
229. Zhou ZH, Zhang ML, Huang SJ, Li YF. Multi-instance multi-label learning. *Artif Intell* 2012, 176:2291–2320.
230. He J, Gu H, Wang Z. Multi-instance multi-label learning based on Gaussian process with application to visual mobile robot navigation. *Inform Sci* 2012, 190:162–177.
231. Zhou ZH, Zhang ML. Multi-instance multi-label learning with application to scene classification. In: *NIPS*; 2006, 1609–1616.
232. Xu X, Frank E. Logistic regression and boosting for labeled bags of instances. In: *Advances in Knowledge Discovery and Data Mining*, Lecture Notes in Computer Science, vol. 3056, Berlin/Heidelberg: Springer; 2004, 272–281.
233. Shen C, Jing L, Ng M. Sparse-MIML: a sparsity-based multi-instance multi-learning algorithm. In: *Energy Minimization Methods in Computer Vision and Pattern Recognition*, Lecture Notes in Computer Science, vol. 8081; 2013, 294–306.
234. Zhang ML. A k-nearest neighbor based multi-instance multi-label learning algorithm. In: *Proceedings of the 22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, vol. 2, Arras, France; 2010, 207–212.
235. Zhang ML, Wang ZJ. MIMLRBF: RBF neural networks for multi-instance multi-label learning. *Neurocomputing* 2009, 72:3951–3956.
236. Zhang ML, Zhou ZH. M3MIML: a maximum margin method for multi-instance multi-label learning. In: *ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, Washington, DC, USA; 2008, 688–697.
237. Yang SH, Zha H, Hu BG. Dirichlet–Bernoulli alignment: a generative model for multi-class multi-label multi-instance corpora. In: *Annual Conference on Neural Information Processing Systems*, Vancouver, British Columbia, Canada; 2009, 2143–2150.
238. Liu Y, Jin R, Yang L. Semi-supervised multi-label learning by constrained non-negative matrix factorization. In: *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI06)*, vol. 1; 2006, 421–426.
239. Chen G, Song Y, Wang F, Zhang C. Semi-supervised multi-label learning by solving a sylvester equation. In: *Proceedings of the SIAM International Conference on Data Mining (SDM)*; 2008, 410–419.
240. Hu DY, Reichel L. Krylov-subspace methods for the Sylvester equation. *Linear Algebra Appl* 1992, 172:283–313.
241. Zha ZJ, Mei T, Wang J, Wang Z, Hua XS. Graph-based semi-supervised learning with multiple labels (special issue on emerging techniques for multimedia content sharing, search and understanding). *Journal of Visual Communication and Image Representation* 2009, 20:97–103.
242. Ahmed MS, Khan L, Oza NC, Rajeswari M. Multi-label ASRS dataset classification using semi supervised subspace clustering. In: *Proceedings of the 2010 Conference on Intelligent Data Understanding (CIDU)*; 2010, 285–299.
243. Wu L, Zhang ML. Multi-label classification with unlabeled data: an inductive approach. In: *Proceedings of the 5th Asian Conference on Machine Learning (ACML'13)*, Canberra, Australia; 2013, 197–212.
244. Chapelle O, Sindhwaniand V, Keerthi SS. Optimization techniques for semisupervised support vector machines. *J Mach Learn Res* 2008, 9:203–233.
245. Brinker K. On active learning in multi-label classification. In: *From Data and Information Analysis to Knowledge Engineering, Studies in Classification, Data Analysis, and Knowledge Organization*. Berlin/Heidelberg: Springer; 2006, 206–213.
246. Qi GJ, Hua XS, Rui Y, Tang J, Zhang HJ. Two-dimensional multilabel active learning with an efficient online adaptation model for image classification. *IEEE Trans Pattern Anal Mach Intell* 2009, 31:1880–1897.
247. Zhang X, Cheng J, Xu C, Lu H, Ma S. Multi-view multi-label active learning for image classification. In: *IEEE International Conference on Multimedia and Expo*, 2009, 258–261.
248. Yang B, Sun JT, Wang T, Chen Z. Effective multi-label active learning for text classification. In: *KDD '09: Proceedings of the 15th International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA; 2009, 917–926.
249. Esuli A, Sebastiani F. Active learning strategies for multi-label text classification. In: *Advances in Information Retrieval*, Lecture Notes in Computer Science, vol. 5478. Berlin/Heidelberg: Springer; 2009, 102–113.
250. Qu W, Zhang Y, Zhu J, Qiu Q. Mining multi-label concept-drifting data streams using dynamic classifier ensemble. In: *Advances in Machine Learning*, Lecture Notes in Computer Science, vol. 5828, Berlin/Heidelberg: Springer; 2009, 308–321.
251. Domingos P, Hulten G. Mining high-speed data streams. In: *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '00)*, New York, NY, USA; 2000, 71–80.
252. Bifet A, Holmes G, Pfahringer B, Kirkby R, Gavaldà R. New ensemble methods for evolving data streams.

In: *15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; 2009, 139–148.

253. Read J, Pfahringer B, Holmes G. Generating synthetic multi-label data streams. In: *ECML/PKKD 2009 Workshop on Learning from Multi-label Data (MLD'09)*; 2009.
254. Xioufis ES, Spiliopoulou M, Tsoumakas G, Vlahavas IP. Dealing with concept drift and class imbalance in multi-label stream classification. In: *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, Barcelona, Spain; 2011, 1583–1588.
255. The Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nat Genet* 2000, 25:25–29.
256. Cerri, R., Barros, R.C., de Carvalho, A.C.P.L.F. A genetic algorithm for Hierarchical Multi-Label Classification. In: *Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC '12)*, New York, NY, USA; 2012, 250–255.
257. Kiritchenko S, Matwin S, Famili AF. Functional annotation of genes using hierarchical text categorization. In: *Proceedings of the BioLINK SIG: Linking Literature, Information and Knowledge for Biology* (held at ISMB-05); 2005.
258. Blockeel H, Schietgat L, Struyf J, Dzrěoski S, Clare A. Decision trees for hierarchical multilabel classification: a case study in functional genomics. In: *10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, Lecture Notes in Computer Science, vol. 4213; 2006, 18–29.
259. Clare A. Machine learning and data mining for yeast functional genomics. PhD Thesis, University of Wales, 2003.
260. Cesa-Bianchi, N., Gentile, C., Zaniboni, L. Hierarchical classification: combining Bayes with SVM. In: *Proceedings of the Twenty-Third International Conference on Machine Learning (ICML)*; 2006, 177–184.
261. Esuli A, Fagni T, Sebastiani F. TreeBoost.MH: a boosting algorithm for multi-label hierarchical text categorization. In: *String Processing and Information Retrieval (SPIRE)*, Lecture Notes in Computer Science, vol. 4209. Berlin/Heidelberg: Springer; 2006, 13–24.
262. Brucker F, Benites F, Sapozhnikova E. Multi-label classification and extracting predicted class hierarchies. *Pattern Recogn* 2010, 44:724–738.
263. Charte F, Rivera A, del Jesus M, Herrera F. A first approach to deal with imbalance in multi-label datasets. In: *HAIS 2013—LNAI 8073*; 2013; 150–160.
264. Dendamrongvit S, Kubat M. Undersampling approach for imbalanced training sets and induction from multi-label text-categorization domains. In: *New Frontiers in Applied Data Mining*, LNCS, vol. 5669. Berlin/Heidelberg: Springer; 2010, 40–52.
265. Tahir MA, Kittler J, Bouridane A. Multilabel classification using heterogeneous ensemble of multi-label classifiers. *Pattern Recogn Lett* 2012, 33:513–523.
266. Nasierding G, Kouzani AZ. Empirical study of multi-label classification methods for image annotation and retrieval. In: *Digital Image Computing: Techniques and Applications*; 2010, 617–622.
267. Dimou A, Tsoumakas G, Mezaris V, Kompatsiaris I, Vlahavas I. An empirical study of multi-label learning methods for video annotation. In: *International Workshop on Content-Based Multimedia Indexing*, IEEE Computer Society, Los Alamitos, CA, USA; 2009, 19–24.
268. Chekina L, Rokach L, Shapira B. Meta-learning for selecting a multi-label classification algorithm. In: *IEEE 11th International Conference on Data Mining Workshops (ICDMW)*; 2011, 220–227.
269. Dembczyński K, Waegeman W, Cheng W, Hüllermeier E. On label dependence and loss minimization in multi-label classification. *Mach Learn* 2012, 88:5–45.
270. Read J. Advances in multi-label classification. Available at: http://users.ics.aalto.fi/jesse/talks/Charla-Malaga.pdf. (2011).

