Mining Multi-label Data

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas

1 Introduction

A large body of research in supervised learning deals with the analysis of *single-label* data, where training examples are associated with a single label λ from a set of disjoint labels L. However, training examples in several application domains are often associated with a *set* of labels $Y \subseteq L$. Such data are called *multi-label*.

Textual data, such as documents and web pages, are frequently annotated with more than a single label. For example, a news article concerning the reactions of the Christian church to the release of the "Da Vinci Code" film can be labeled as both *religion* and *movies*. The categorization of textual data is perhaps the dominant multi-label application.

Recently, the issue of learning from multi-label data has attracted significant attention from a lot of researchers, motivated from an increasing number of new applications, such as semantic annotation of images [1, 2, 3] and video [4, 5], functional genomics [6, 7, 8, 9, 10], music categorization into emotions [11, 12, 13, 14] and directed marketing [15]. Table 1 presents a variety of applications that are discussed in the literature.

This chapter reviews past and recent work on the rapidly evolving research area of multi-label data mining. Section 2 defines the two major tasks in learning from multi-label data and presents a significant number of learning methods. Section 3 discusses dimensionality reduction methods for multi-label data. Sections 4 and 5 discuss two important research challenges, which, if successfully met, can significantly expand the real-world applications of multi-label learning methods: a) exploiting label structure and b) scaling up to domains with large number of labels. Section 6 introduces benchmark multi-label datasets and their statistics, while Section 7 presents the most frequently used evaluation measures for multi-label learn-

Grigorios Tsoumakas · Ioannis Katakis · Ioannis Vlahavas
Dept. of Informatics, Aristotle University of Thessaloniki, 54124 Greece
e-mail: greg,katak,vlahavas@csd.auth.gr

Data type	Application	Resource	Labels Description (Examples)	References
text	categorization	news article	Reuters topics (agriculture, fishing)	[16]
		web page	Yahoo! directory (health, science)	[17]
		patent	WIPO (paper-making, fibreboard)	[18, 19]
		email	R&D activities (delegation)	[20]
		legal document	Eurovoc (software, copyright)	[21]
		research article	Heart conditions (myocarditis)	[22]
		research article	ACM classification (algorithms)	[23]
		bookmark	Bibsonomy tags (sports, science)	[24]
		reference	Bibsonomy tags (ai, kdd)	[24]
		adjectives	semantics (object-related)	[25]
image	semantic annotation	pictures	concepts (trees, sunset)	[1, 2, 3]
video	semantic annotation	news clip	concepts (crowd, desert)	[4]
audio	noise detection	sound clip	type (speech, noise)	[26]
	emotion detection	music clip	emotions (relaxing-calm)	[11, 14]
structured	functional genomics	gene	functions (energy, metabolism)	[7, 6, 8]
	proteomics	protein	enzyme classes (ligases)	[19]
	directed marketing	person	product categories	[15]

Table 1: Applications of multi-label Learning

ing. We conclude this chapter by discussing related tasks to multi-label learning in Section 8.

2 Learning

There exist two major tasks in supervised learning from multi-label data: *multi-label classification* (MLC) and *label ranking* (LR). MLC is concerned with learning a model that outputs a bipartition of the set of labels into relevant and irrelevant with respect to a query instance. LR on the other hand is concerned with learning a model that outputs an ordering of the class labels according to their relevance to a query instance. Note that LR models can also be learned from training data containing single labels, total rankings of labels, as well as pairwise preferences over the set of labels.

Both MLC and LR are important in mining multi-label data. In a news filtering application for example, the user must be presented with interesting articles only, but it is also important to see the most interesting ones in the top of the list. Ideally, we would like to develop methods that are able to mine both an ordering and a bipartition of the set of labels from multi-label data. Such a task has been recently called *multi-label ranking* (MLR) [27] and poses a very interesting and useful generalization of MLC and LR.

In the following subsections we present MLC, LR and MLR methods grouped into the two categories proposed in [28]: i) *problem transformation*, and ii) *algorithm adaptation*. The first group of methods are algorithm independent. They trans-

form the learning task into one or more single-label classification tasks, for which a large bibliography of learning algorithms exists. The second group of methods extend specific learning algorithms in order to handle multi-label data directly.

For the formal description of these methods, we will use $L = \{\lambda_j : j = 1 \ldots M\}$ to denote the finite set of labels in a multi-label learning task and $D = \{(x_i, Y_i), i = 1 \ldots N\}$ to denote a set of multi-label training examples, where x_i is the feature vector and $Y_i \subseteq L$ the set of labels of the i-th example.

2.1 Problem Transformation

Problem transformation methods will be exemplified through the multi-label data set of Figure 1. It consists of four examples that are annotated with one or more out of four labels: λ_1, λ_2, λ_3, λ_4.

Fig. 1 Example of a multi-label data set

Example	Label set
1	$\{\lambda_1, \lambda_4\}$
2	$\{\lambda_3, \lambda_4\}$
3	$\{\lambda_1\}$
4	$\{\lambda_2, \lambda_3, \lambda_4\}$

There exist several simple transformations that can be used to convert a multi-label data set to a single-label data set with the same set of labels [1, 29]. A single-label classifier that outputs probability distributions over all classes can then be used to learn a ranking. The *copy* transformation replaces each multi-label example (x_i, Y_i) with $|Y_i|$ examples (x_i, λ_j), for every $\lambda_j \in Y_i$. A variation of this transformation, dubbed *copy-weight*, associates a weight of $\frac{1}{|Y_i|}$ to each of the produced examples. The *select* family of transformations replaces Y_i with one of its members. This label could be the most (*select-max*) or least (*select-min*) frequent among all examples. It could also be randomly selected (*select-random*). Finally, the *ignore* transformation simply discards every multi-label example. Figure 2 shows the transformed data set using these simple transformations.

Label powerset (LP) is a simple but effective problem transformation method that works as follows: It considers each unique set of labels that exists in a multi-label training set as one of the classes of a new single-label classification task. Figure 3 shows the result of transforming the data set of Figure 1 using LP.

Given a new instance, the single-label classifier of LP outputs the most probable class, which is actually a set of labels. If this classifier can output a probability distribution over all classes, then LP can also rank the labels following the approach in [30]. Table 2 shows an example of a probability distribution that could be produced by LP, trained on the data of Figure 3, given a new instance x with unknown label set. To obtain a label ranking we calculate for each label the sum of the probabilities of the classes that contain it. This way LP can solve the complete MLR task.

Ex.	Label
1a	λ_1
1b	λ_4
2a	λ_3
2b	λ_4
3	λ_1
4a	λ_2
4b	λ_3
4c	λ_4

(a)

Ex.	Label	Weight
1a	λ_1	0.50
1b	λ_4	0.50
2a	λ_3	0.50
2b	λ_4	0.50
3	λ_1	1.00
4a	λ_2	0.33
4b	λ_3	0.33
4c	λ_4	0.33

(b)

Ex.	Label
1	λ_4
2	λ_4
3	λ_1
4	λ_4

(c)

Ex.	Label
1	λ_1
2	λ_3
3	λ_1
4	λ_2

(d)

Ex.	Label
1	λ_1
2	λ_4
3	λ_1
4	λ_3

(e)

Ex.	Label
3	λ_1

(f)

Fig. 2: Transformation of the data set in Figure 1 using (a) *copy*, (b) *copy-weight*, (c) *select-max*, (d) *select-min*, (e) *select-random* (one of the possible) and (f) *ignore*

Fig. 3 Transformed data set using the label powerset method

Ex.	Label
1	$\lambda_{1,4}$
2	$\lambda_{3,4}$
3	λ_1
4	$\lambda_{2,3,4}$

Table 2: Example of obtaining a ranking from LP

c	$p(c\|x)$	λ_1	λ_2	λ_3	λ_4
$\lambda_{1,4}$	0.7	1	0	0	1
$\lambda_{3,4}$	0.2	0	0	1	1
λ_1	0.1	1	0	0	0
$\lambda_{2,3,4}$	0.0	0	1	1	1
	$\sum_c p(c\|x)\lambda_j$	0.8	0.0	0.2	0.9

The computational complexity of LP with respect to M depends on the complexity of the base classifier with respect to the number of classes, which is equal to the number of distinct label sets in the training set. This number is upper bounded by $\min(N, 2^M)$ and despite that it typically is much smaller, it still poses an important complexity problem, especially for large values of N and M. The large number of classes, many of which are associated with very few examples, makes the learning process difficult as well.

The pruned problem transformation (PPT) method [30] extends LP in an attempt to deal with the aforementioned problems. It prunes away label sets that occur less times than a small user-defined threshold (e.g. 2 or 3) and optionally replaces their information by introducing disjoint subsets of these label sets that do exist more times than the threshold.

The random k-labelsets (RAKEL) method [31] constructs an ensemble of LP classifiers. Each LP classifier is trained using a different small random subset of the set of labels. This way RAKEL manages to take label correlations into account, while avoiding LP's problems. A ranking of the labels is produced by averaging the

zero-one predictions of each model per considered label. Thresholding is then used to produce a bipartition as well.

Binary relevance (BR) is a popular problem transformation method that learns M binary classifiers, one for each different label in L. It transforms the original data set into M data sets $D_{\lambda_j}, j = 1 \ldots M$ that contain all examples of the original data set, labeled positively if the label set of the original example contained λ_j and negatively otherwise. For the classification of a new instance, BR outputs the union of the labels λ_j that are positively predicted by the M classifiers. Figure 4 shows the four data sets that are constructed by BR when applied to the data set of Figure 1.

Ex.	Label
1	λ_1
2	$\neg\lambda_1$
3	λ_1
4	$\neg\lambda_1$

(a)

Ex.	Label
1	$\neg\lambda_2$
2	$\neg\lambda_2$
3	$\neg\lambda_2$
4	λ_2

(b)

Ex.	Label
1	$\neg\lambda_3$
2	λ_3
3	$\neg\lambda_3$
4	λ_3

(c)

Ex.	Label
1	λ_4
2	λ_4
3	$\neg\lambda_4$
4	λ_4

(d)

Fig. 4: Data sets produced by the BR method

Ranking by pairwise comparison (RPC) [32] transforms the multi-label dataset into $\frac{M(M-1)}{2}$ binary label datasets, one for each pair of labels $(\lambda_i, \lambda_j), 1 \leq i < j \leq M$. Each dataset contains those examples of D that are annotated by at least one of the two corresponding labels, but not both. A binary classifier that learns to discriminate between the two labels, is trained from each of these data sets. Given a new instance, all binary classifiers are invoked, and a ranking is obtained by counting the votes received by each label. Figure 5 shows the data sets that are constructed by RPC when applied to the data set of Figure 1. The multi-label pairwise perceptron (MLPP) algorithm [33] is an instantiation of RPC using perceptrons for the binary classification tasks.

Ex.	Label
1	$\lambda_{1,\neg 2}$
3	$\lambda_{1,\neg 2}$
4	$\lambda_{\neg 1,2}$

(a)

Ex.	Label
1	$\lambda_{1,\neg 3}$
2	$\lambda_{\neg 1,3}$
3	$\lambda_{1,\neg 3}$
4	$\lambda_{\neg 1,3}$

(b)

Ex.	Label
2	$\lambda_{\neg 1,4}$
3	$\lambda_{1,\neg 4}$
4	$\lambda_{\neg 1,4}$

(c)

Ex.	Label
2	$\lambda_{\neg 2,3}$

(d)

Ex.	Label
1	$\lambda_{\neg 2,4}$
2	$\lambda_{\neg 2,4}$

(e)

Ex.	Label
1	$\lambda_{\neg 3,4}$

(f)

Fig. 5: Data sets produced by the RPC method

Calibrated label ranking (CLR) [34] extends RPC by introducing an additional virtual label, which acts as a natural breaking point of the ranking into relevant and irrelevant sets of labels. This way, CLR manages to solve the complete MLR task.

The INSDIF algorithm [35] computes a prototype vector for each label, by averaging all instances of the training set that belong to this label. After that, every

instance is transformed to a bag of M instances, each equal to the difference between the initial instance and one of the prototype vectors. A two level classification strategy is then employed to learn form the transformed data set.

2.2 Algorithm Adaptation

In this section, we briefly discuss a plethora of algorithm adaptation methods grouped by the learning paradigm that they extend.

2.2.1 Decision Trees and Boosting

The C4.5 algorithm was adapted in [6] for the handling of multi-label data. In specific, multiple labels were allowed at the leaves of the tree and the formula of entropy calculation was modified as follows:

$$\text{Entropy}(D) = -\sum_{j=1}^{M} (p(\lambda_j) log p(\lambda_j) + q(\lambda_j) log q(\lambda_j)) \quad (1)$$

where $p(\lambda_j) =$ relative frequency of class λ_j and $q(\lambda_j) = 1 - p(\lambda_j)$.

AdaBoost.MH and AdaBoost.MR [16] are two extensions of AdaBoost for multi-label data. While AdaBoost.MH is designed to minimize Hamming loss, AdaBoost.MR is designed to find a hypothesis which places the correct labels at the top of the ranking.

A combination of AdaBoost.MH with an algorithm for producing alternating decision trees was presented in [36]. The main motivation was the production of multi-label models that can be understood by humans.

2.2.2 Probabilistic Methods

A probabilistic generative model is proposed in [37], according to which, each label generates different words. Based on this model a multi-label document is produced by a mixture of the word distributions of its labels. A similar word-based mixture model for multi-label text classification is presented in [17]. A deconvolution approach is proposed in [26], in order to estimate the individual contribution of each label to a given item.

The use of conditional random fields is explored in [22], where two graphical models that parameterize label co-occurrences are proposed. The first one, *collective multi-label*, captures co-occurrence patterns among labels, whereas the second one, *collective multi-label with features*, tries to capture the impact that an individual feature has on the co-occurrence probability of a pair of labels.

2.2.3 Neural Networks and Support Vector Machines (SVMs)

BP-MLL [38] is an adaptation of the popular back-propagation algorithm for multi-label learning. The main modification to the algorithm is the introduction of a new error function that takes multiple labels into account.

The multi-class multi-label perceptron (MMP) [39] is a family of online algorithms for label ranking from multi-label data based on the perceptron algorithm. MMP maintains one perceptron for each label, but weight updates for each perceptron are performed so as to achieve a perfect ranking of all labels.

An SVM algorithm that minimizes the ranking loss (see Section 7.2) is proposed in [7]. Three improvements to instantiating the BR method with SVM classifiers are given in [18]. The first two could easily be abstracted in order to be used with any classification algorithm and could thus be considered an extension to BR itself, while the third is specific to SVMs.

The main idea in the first improvement is to extend the original data set with M additional features containing the predictions of each binary classifier. Then a second round of training M new binary classifiers takes place, this time using the extended data sets. For the classification of a new example, the binary classifiers of the first round are initially used and their output is appended to the features of the example to form a meta-example. This meta-example is then classified by the binary classifiers of the second round. Through this extension, the approach takes into consideration the potential dependencies among the different labels. Note here that this improvement is actually a specialized case of applying Stacking [40], a method for the combination of multiple classifiers, on top of BR.

The second improvement, *ConfMat*, consists in removing negative training instances of a complete label if it is very similar to the positive label, based on a confusion matrix that is estimated using any fast and moderately accurate classifier on a held out validation set. The third improvement *BandSVM*, consists in removing very similar negative training instances that are within a threshold distance from the learned hyperplane.

2.2.4 Lazy and Associative Methods

A number of methods [41, 13, 42, 2, 43] are based on the popular k Nearest Neighbors (kNN) lazy learning algorithm. The first step in all these approaches is the same as in kNN, i.e. retrieving the k nearest examples. What differentiates them is the aggregation of the label sets of these examples.

For example, ML-kNN [2], uses the maximum a posteriori principle in order to determine the label set of the test instance, based on prior and posterior probabilities for the frequency of each label within the k nearest neighbors.

MMAC [44] is an algorithm that follows the paradigm of associative classification, which deals with the construction of classification rule sets using association rule mining. MMAC learns an initial set of classification rules through association rule mining, removes the examples associated with this rule set and recursively

learns a new rule set from the remaining examples until no further frequent items are left. These multiple rule sets might contain rules with similar preconditions but different labels on the right hand side. Such rules are merged into a single multi-label rule. The labels are ranked according to the support of the corresponding individual rules.

Finally, an approach that combines lazy and associative learning is proposed in [23], where the inductive process is delayed until an instance is given for classification.

3 Dimensionality Reduction

Several application domains of multi-label learning (e.g. text, bioinformatics) involve data with large number of features. Dimensionality reduction has been extensively studied in the case of single-label data. Some of the existing approaches are directly applicable to multi-label data, while others have been extended for handling them appropriately. We present past and very recent approaches to multi-label dimensionality reduction, organized into two categories: i) *feature selection* and ii) *feature extraction*.

3.1 Feature Selection

The *wrapper* approach to feature selection [45] is directly applicable to multi-label data. Given a multi-label learning algorithm, we can search for the subset of features that optimizes a multi-label loss function (see Section 7) on an evaluation data set.

A different line of attacking the multi-label feature selection problem is to transform the multi-label data set into one or more single-label data sets and use existing feature selection methods, particularly those that follow the *filter* paradigm. One of the most popular approaches, especially in text categorization, uses the BR transformation in order to evaluate the discriminative power of each feature with respect to each of the labels independently of the rest of the labels. Subsequently the obtained scores are aggregated in order to obtain an overall ranking. Common aggregation strategies include taking the maximum or a weighted average of the obtained scores [46]. The LP transformation was used in [14], while the *copy*, *copy-weight*, *select-max*, *select-min* and *ignore* transformations are used in [29].

3.2 Feature Extraction

Feature extraction methods construct new features out of the original ones either using class information (supervised) or not (unsupervised).

Unsupervised methods, such as principal component analysis and latent semantic indexing (LSI) are obviously directly applicable to multi-label data. For example, in [47], the authors directly apply LSI based on singular value decomposition in order to reduce the dimensionality of the text categorization problem.

Supervised feature extraction methods for single-label data, such as linear discriminant analysis (LDA), require modification prior to their application to multi-label data. LDA has been modified to handle multi-label data in [48]. A version of the LSI method that takes into consideration label information (MLSI) was proposed in [49], while a supervised multi-label feature extraction algorithm based on the Hilbert-Schmidt independence criterion was proposed in [50]. In [51] a framework for extracting a subspace of features is proposed. Finally, a hypergraph is employed in [52] for modeling higher-order relations among instances sharing the same label. A spectral learning method is then used for computing a low-dimensional embedding that preserves these relations.

4 Exploiting Label Structure

In certain multi-label domains, such as text mining and bioinformatics, labels are organized into a tree-shaped general-to-specific hierarchical structure. An example of such a structure, called functional catalogue (FunCat) [53], is an annotation scheme for the functional description of proteins from several living organisms. The 1362 functional categories in version 2.1 of FunCat are organized in a tree like structure with up to six levels of increasing specificity. Many more hierarchical structures exist for textual data, such as the MeSH[1] for medical articles and the ACM computing classification system[2] for computer science articles. Taking into account such structures when learning from multi-label data is important, because it can lead to improved predictive performance and time complexity.

A general-to-specific tree structure of labels implies that an example cannot be associated with a label λ if it isn't associated with its parent label $\text{par}(\lambda)$. In other words, the set of labels associated with an example must be a union of the labels found along zero or more paths starting at the root of the hierarchy. Some applications may require such paths to end at a leaf, but in the general case they can be partial.

Given a label hierarchy, a straightforward approach to learning a multi-label classifier is to train a binary classifier for each non-root label λ of this hierarchy, using as training data those examples of the full training set that are annotated with $\text{par}(\lambda)$. During testing, these classifiers are called in a top-down manner, calling a classifier for λ only if the classifier for $\text{par}(\lambda)$ has given a positive output. We call this the *hierarchical binary relevance* (HBR) method.

[1] www.nlm.nih.gov/mesh/

[2] www.acm.org/class/

An online learning algorithm that follows the HBR approach, using a regularized least squares estimator at each node, is presented in [54]. Better results were found compared to an instantiation of HBR using perceptrons. Other important contributions of [54] are the definition of a hierarchical loss function (see Section 7.1.1) and a thorough theoretical analysis of the proposed algorithm. An approach that follows the training process of HBR but uses a bottom-up procedure during testing is presented in [9].

The HBR approach can be reformulated in a more generalized fashion as the training of a multi-label (instead of binary) classifier in all non-leaf (instead of non-root) nodes [55, 56]. TreeBoost.MH [55] uses Adaboost.MH (see Section 2.2) at each non-leaf node. Experimental results indicate that not only is TreeBoost.MH more efficient in training and testing than Adaboost.MH, but that it also improves predictive accuracy.

Two different approaches for exploiting tree-shaped hierarchies are [8, 19]. Predictive clustering trees are used in [8], while a large margin method for structured output prediction is used in [19].

The directed acyclic graph (DAG) is a more general type of structure, where a node can have multiple parents. This is the case for the Gene Ontology (GO) [57], which covers several domains of molecular and cellular biology. A Bayesian framework for combining a hierarchy of support vector machines based on the GO is proposed in [10]. An extension of the work in [8] for handling DAG label structures is presented in [58].

5 Scaling Up

Problems with large number of labels can be found in several domains. For example, the *Eurovoc*[3] taxonomy contains approximately 4000 descriptors European for documents, while in collaborative tagging systems such as *delicious*[4], the user assigned tags can be hundreds of thousands.

The high dimensionality of the label space may challenge a multi-label learning algorithm in many ways. Firstly, the number of training examples annotated with each particular label will be significantly less than the total number of examples. This is similar to the class imbalance problem in single-label data [59]. Secondly, the computational cost of training a multi-label model may be strongly affected by the number of labels. There are simple algorithms, such as BR with linear complexity with respect to M, but there are others, such as LP, whose complexity is worse. Thirdly, although the complexity of using a multi-label model for prediction is linear with respect to M in the best case, this may still be inefficient for applications requiring fast response times. Finally, methods that need to maintain a large number of models in memory, may fail to scale up to such domains.

[3] europa.eu/eurovoc/

[4] delicious.com

HOMER [56] constructs a Hierarchy Of Multilabel classifiERs each one dealing with a much smaller set of labels compared to M and a more balanced example distribution. This leads to improved predictive performance along with linear training and logarithmic testing complexities withs respect to M. At a first step, HOMER automatically organizes labels into a tree-shaped hierarchy. This is accomplished by recursively partitioning the set of labels into a number of nodes using a balance clustering algorithm. It then builds one multi-label classifier at each node apart from the leafs, following the HBR approach described in the previous Section. The multi-label classifiers predict one or more meta-labels μ, each one corresponding to the disjunction of a child node's labels. Figure 6 presents a sample tree of multi-label classifiers constructed by HOMER for a domain with 8 labels.

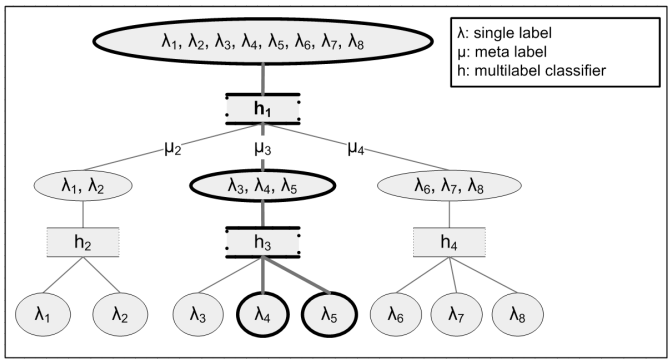

Fig. 6: Sample hierarchy for a multi-label domain with 8 labels.

To deal with the memory problem of RPC, an extension of MLPP with reduced space complexity in the presence of large number of labels is described in [60].

6 Statistics and Datasets

In some applications the number of labels of each example is small compared to M, while in others it is large. This could be a parameter that influences the performance of the different multi-label methods. We here introduce the concepts of label cardinality and label density of a data set.

Label cardinality of a dataset D is the average number of labels of the examples in D:

$$\text{Label-Cardinality} = \frac{1}{N}\sum_{i=1}^{N}|Y_i|$$

Label density of D is the average number of labels of the examples in D divided by M:

$$\text{Label-Density} = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i|}{M} \tag{2}$$

Label cardinality is independent of the number of labels M in the classification problem, and is used to quantify the number of alternative labels that characterize the examples of a multi-label training data set. Label density takes into consideration the number of labels in the domain. Two data sets with the same label cardinality but with a great difference in the number of labels (different label density) might not exhibit the same properties and cause different behavior to the multi-label learning methods. The number of distinct label sets is also important for many algorithm transformation methods that operate on subsets of labels.

Table 3 presents some benchmark datasets[5] from various domains among with their corresponding statistics and source reference. The statistics of the Reuters (rcv1v2) dataset are averages over the 5 subsets.

Table 3: Multilabel datasets and their statistics

name	domain	instances	nominal	numeric	labels	cardinality	density	distinct	source
delicious	text (web)	16105	500	0	983	19.020	0.019	15806	[28]
emotions	music	593	0	72	6	1.869	0.311	27	[14]
genbase	biology	662	1186	0	27	1.252	0.046	32	[61]
mediamill	multimedia	43907	0	120	101	4.376	0.043	6555	[5]
rcv1v2 (avg)	text	6000	0	47234	101	2.6508	0.026	937	[62]
scene	multimedia	2407	0	294	6	1.074	0.179	15	[1]
yeast	biology	2417	0	103	14	4.237	0.303	198	[7]
tmc2007	text	28596	49060	0	22	2.158	0.098	1341	[63]

7 Evaluation Measures

The evaluation of methods that learn from multi-label data requires different measures than those used in the case of single-label data. This section presents the various measures that have been proposed in the past for the evaluation of i) bipartitions and ii) rankings with respect to the ground truth of multi-label data.

For the definitions of these measures we will consider an evaluation data set of multi-label examples (x_i, Y_i), $i = 1 \ldots N$, where $Y_i \subseteq L$ is the set of true labels and $L = \{\lambda_j : j = 1 \ldots M\}$ is the set of all labels. Given instance x_i, the set of labels that are predicted by an MLC method is denoted as Z_i, while the rank predicted by an LR method for a label λ is denoted as $r_i(\lambda)$. The most relevant label, receives the highest rank (1), while the least relevant one, receives the lowest rank (M).

[5] All datasets are available for download at http://mlkd.csd.auth.gr/multilabel.html

7.1 Bipartitions

Some of the measures that evaluate bipartitions are calculated based on the average differences of the actual and the predicted sets of labels over all examples of the evaluation data set. Others decompose the evaluation process into separate evaluations for each label, which they subsequently average over all labels. We call the former *example-based* and the latter *label-based* evaluation measures.

7.1.1 Example-based

The Hamming loss [16] is defined as follows:

$$\text{Hamming-Loss} = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \triangle Z_i|}{M}$$

where $\triangle$ stands for the symmetric difference of two sets, which is the set-theoretic equivalent of the exclusive disjunction (XOR operation) in Boolean logic.

The hierarchical loss [54] is a modified version of the Hamming loss that takes into account an existing hierarchical structure of the labels. It examines the predicted labels in a top-down manner according to the hierarchy and whenever the prediction for a label is wrong, the subtree rooted at that node is not considered further in the calculation of the loss. Let $\text{anc}(\lambda)$ be the set of all the ancestor nodes of λ. The hierarchical loss is defined as follows:

$$\text{H-Loss} = \frac{1}{N}\sum_{i=1}^{N}|\{\lambda : \lambda \in Y_i \triangle Z_i, \text{anc}(\lambda) \cap (Y_i \triangle Z_i) = \emptyset\}|$$

Classification accuracy [20] or subset accuracy [22] is defined as follows:

$$\text{ClassificationAccuracy} = \frac{1}{N}\sum_{i=1}^{N} I(Z_i = Y_i)$$

where $I(\text{true}) = 1$ and $I(\text{false}) = 0$. This is a very strict evaluation measure as it requires the predicted set of labels to be an exact match of the true set of labels.

The following measures are used in [18]:

$$\text{Precision} = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap Z_i|}{|Z_i|} \qquad \text{Recall} = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap Z_i|}{|Y_i|}$$

$$F_1 = \frac{1}{N}\sum_{i=1}^{N}\frac{2|Y_i \cap Z_i|}{|Z_i| + |Y_i|} \qquad \text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|}$$

7.1.2 Label-based

Any known measure for binary evaluation can be used here, such as accuracy, area under the ROC curve, precision and recall. The calculation of these measures for all labels can be achieved using two averaging operations, called *macro-averaging* and *micro-averaging* [64]. These operations are usually considered for averaging precision, recall and their harmonic mean (F-measure) in Information Retrieval tasks.

Consider a binary evaluation measure $B(tp,tn,fp,fn)$ that is calculated based on the number of true positives (tp), true negatives (tn), false positives (fp) and false negatives (fn). Let tp_λ, fp_λ, tn_λ and fn_λ be the number of true positives, false positives, true negatives and false negatives after binary evaluation for a label λ. The macro-averaged and micro-averaged versions of B, are calculated as follows:

$$B_{\text{macro}} = \frac{1}{M}\sum_{\lambda=1}^{M} B(tp_\lambda, fp_\lambda, tn_\lambda, fn_\lambda)$$

$$B_{\text{micro}} = B\left(\sum_{\lambda=1}^{M} tp_\lambda, \sum_{\lambda=1}^{M} fp_\lambda, \sum_{\lambda=1}^{M} tn_\lambda, \sum_{\lambda=1}^{M} fn_\lambda\right)$$

Note that micro-averaging has the same result as macro-averaging for some measures, such as accuracy, while it differs for other measures, such as precision, recall and area under the ROC curve. Note also that the average (macro/micro) accuracy and Hamming loss sum up to 1, as Hamming loss is actually the average binary classification error.

7.2 Ranking

One-error evaluates how many times the top-ranked label is not in the set of relevant labels of the instance:

$$\text{1-Error} = \frac{1}{N}\sum_{i=1}^{N} \delta(\arg\min_{\lambda \in L} r_i(\lambda))$$

where

$$\delta(\lambda) = \begin{cases} 1 & \text{if } \lambda \notin Y_i \\ 0 & \text{otherwise} \end{cases}$$

Coverage evaluates how far we need, on average, to go down the ranked list of labels in order to cover all the relevant labels of the example.

$$\text{Cov} = \frac{1}{N}\sum_{i=1}^{N} \max_{\lambda \in Y_i} r_i(\lambda) - 1$$

Ranking loss expresses the number of times that irrelevant labels are ranked higher than relevant labels:

$$\text{R-Loss} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|Y_i||\overline{Y_i}|}|\{(\lambda_a,\lambda_b) : r_i(\lambda_a) > r_i(\lambda_b), (\lambda_a,\lambda_b) \in Y_i \times \overline{Y_i}\}|$$

where $\overline{Y_i}$ is the complementary set of Y_i with respect to L.

Average precision evaluates the average fraction of labels ranked above a particular label $\lambda \in Y_i$ which actually are in Y_i.

$$\text{AvgPrec} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|Y_i|}\sum_{\lambda \in Y_i}\frac{|\{\lambda' \in Y_i : r_i(\lambda') \leq r_i(\lambda)\}|}{r_i(\lambda)}$$

8 Related Tasks

Jin and Ghahramani [65] call *multiple-label problems*, the semi-supervised classification problems where each example is associated with more than one classes, but only one of those classes is the true class of the example. This task is not that common in real-world applications as the one we are studying.

Multiple-instance or *multi-instance* learning is a variation of supervised learning, where labels are assigned to bags of instances [66]. In certain applications, the training data can be considered as both multi-instance and multi-label [67]. In image classification for example, the different regions of an image can be considered as multiple-instances, each of which can be labeled with a different concept, such as *sunset* and *sea*. Several methods have been recently proposed for addressing such data [68, 69].

In *Multitask learning* [70] we try to solve many similar tasks in parallel usually using a shared representation. Taking advantage of the common characteristics of these tasks a better generalization can be achieved. A typical example is to learn to identify hand written text for different writers in parallel. Training data from one writer can aid the construction of better predictive models for other authors.

References

1. Boutell, M., Luo, J., Shen, X., Brown, C.: Learning multi-label scene classification. Pattern Recognition **37** (2004) 1757–1771
2. Zhang, M.L., Zhou, Z.H.: Ml-knn: A lazy learning approach to multi-label learning. Pattern Recognition **40** (2007) 2038–2048
3. Yang, S., Kim, S.K., Ro, Y.M.: Semantic home photo categorization. Circuits and Systems for Video Technology, IEEE Transactions on **17** (2007) 324–335
4. Qi, G.J., Hua, X.S., Rui, Y., Tang, J., Mei, T., Zhang, H.J.: Correlative multi-label video annotation. In: MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia, New York, NY, USA, ACM (2007) 17–26

5. Snoek, C.G.M., Worring, M., van Gemert, J.C., Geusebroek, J.M., Smeulders, A.W.M.: The challenge problem for automated detection of 101 semantic concepts in multimedia. In: MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia, New York, NY, USA, ACM (2006) 421–430
6. Clare, A., King, R.: Knowledge discovery in multi-label phenotype data. In: Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2001), Freiburg, Germany (2001) 42–53
7. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Advances in Neural Information Processing Systems 14. (2002)
8. Blockeel, H., Schietgat, L., Struyf, J., Dz?eroski, S., Clare, A.: Decision trees for hierarchical multilabel classification: A case study in functional genomics. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **4213 LNAI** (2006) 18–29
9. Cesa-Bianchi, N., Gentile, C., Zaniboni, L.: Hierarchical classification: combining bayes with svm. In: ICML '06: Proceedings of the 23rd international conference on Machine learning. (2006) 177–184
10. Barutcuoglu, Z., Schapire, R.E., Troyanskaya, O.G.: Hierarchical multi-label prediction of gene function. Bioinformatics **22** (2006) 830–836
11. Li, T., Ogihara, M.: Detecting emotion in music. In: Proceedings of the International Symposium on Music Information Retrieval, Washington D.C., USA (2003) 239–240
12. Li, T., Ogihara, M.: Toward intelligent music information retrieval. IEEE Transactions on Multimedia **8** (2006) 564–574
13. Wieczorkowska, A., Synak, P., Ras, Z.: Multi-label classification of emotions in music. In: Proceedings of the 2006 International Conference on Intelligent Information Processing and Web Mining (IIPWM'06). (2006) 307–315
14. Trohidis, K., Tsoumakas, G., Kalliris, G., Vlahavas, I.: Multilabel classification of music into emotions. In: Proc. 9th International Conference on Music Information Retrieval (ISMIR 2008), Philadelphia, PA, USA, 2008. (2008)
15. Zhang, Y., Burer, S., Street, W.N.: Ensemble pruning via semi-definite programming. Journal of Machine Learning Research **7** (2006) 1315–1338
16. Schapire, R.E. Singer, Y.: Boostexter: a boosting-based system for text categorization. Machine Learning **39** (2000) 135–168
17. Ueda, N., Saito, K.: Parametric mixture models for multi-labeled text. Advances in Neural Information Processing Systems 15 (2003) 721–728
18. Godbole, S., Sarawagi, S.: Discriminative methods for multi-labeled classification. In: Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2004). (2004) 22–30
19. Rousu, J., Saunders, C., Szedmak, S., Shawe-Taylor, J.: Kernel-based learning of hierarchical multilabel classification methods. Journal of Machine Learning Research **7** (2006) 1601–1626
20. Zhu, S., Ji, X., Xu, W., Gong, Y.: Multi-labelled classification using maximum entropy method. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in Information Retrieval. (2005) 274–281
21. Mencia, E.L., Furnkranz, J.: Efficient pairwise multilabel classification for large scale problems in the legal domain. In: 12th European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD 2008, Antwerp, Belgium (2008)
22. Ghamrawi, N., McCallum, A.: Collective multi-label classification. In: Proceedings of the 2005 ACM Conference on Information and Knowledge Management (CIKM '05), Bremen, Germany (2005) 195–200
23. Veloso, A., Wagner, M.J., Goncalves, M., Zaki, M.: Multi-label lazy associative classification. In: Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2007). Volume LNAI 4702., Warsaw, Poland, Springer (2007) 605–612
24. Katakis, I., Tsoumakas, G., Vlahavas, I.: Multilabel text classification for automated tag suggestion. In: Proceedings of the ECML/PKDD 2008 Discovery Challenge, Antwerp, Belgium (2008)

25. Boleda, G., im Walde, S.S., Badia, T.: Modelling polysemy in adjective classes by multi-label classification. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Prague (2007) 171–180
26. Streich, A.P., Buhmann, J.M.: Classification of multi-labeled data: A generative approach. In: 12th European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD 2008, Antwerp, Belgium (2008)
27. Brinker, K., Furnkranz, J., Hullermeier, E.: A unified model for multilabel classification and ranking. In: Proceedings of the 17th European Conference on Artificial Intelligence (ECAI '06), Riva del Garda, Italy (2006) 489–493
28. Tsoumakas, G., Katakis, I.: Multi-label classification: An overview. International Journal of Data Warehousing and Mining **3** (2007) 1–13
29. Chen, W., Yan, J., Zhang, B., Chen, Z., Yang, Q.: Document transformation for multi-label feature selection in text categorization. In: Proc. 7th IEEE International Conference on Data Mining, Los Alamitos, CA, USA, IEEE Computer Society (2007) 451–456
30. Read, J.: A pruned problem transformation method for multi-label classification. In: Proc. 2008 New Zealand Computer Science Research Student Conference (NZCSRS 2008). (2008) 143–150
31. Tsoumakas, G., Vlahavas, I.: Random k-labelsets: An ensemble method for multilabel classification. In: Proceedings of the 18th European Conference on Machine Learning (ECML 2007), Warsaw, Poland (2007) 406–417
32. Hullermeier, E., Furnkranz, J., Cheng, W., Bringer, K.: Label ranking by learning pairwise preferences. Artificial Intelligence (2008)
33. Loza Mencia, E., Fu"rnkranz, J.: Pairwise learning of multilabel classifications with perceptrons. In: 2008 IEEE International Joint Conference on Neural Networks (IJCNN-08), Hong Kong (2008) 2900–2907
34. Fu"rnkranz, J., Hu"llermeier, E., Mencia, E.L., Brinker, K.: Multilabel classification via calibrated label ranking. Machine Learning (2008)
35. Zhang, M.L., Zhou, Z.H.: Multi-label learning by instance differentiation. In: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, Vancouver, Britiths Columbia, Canada, AAAI Press (2007) 669–674
36. de Comite, 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 2003), Leipzig, Germany (2003) 35–49
37. McCallum, A.: Multi-label text classification with a mixture model trained by em. In: Proceedings of the AAAI' 99 Workshop on Text Learning. (1999)
38. Zhang, M.L., Zhou, Z.H.: Multi-label neural networks with applications to functional genomics and text categorization. IEEE Transactions on Knowledge and Data Engineering **18** (2006) 1338–1351
39. Crammer, K., Singer, Y.: A family of additive online algorithms for category ranking. Journal of Machine Learning Research **3** (2003) 1025–1058
40. Wolpert, D.: Stacked generalization. Neural Networks **5** (1992) 241–259
41. Luo, X., Zincir-Heywood, A.: 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
42. Brinker, K., Hullermeier, E.: Case-based multilabel ranking. In: Proceedings of the 20th International Conference on Artificial Intelligence (IJCAI '07), Hyderabad, India (2007) 702–707
43. Spyromitros, E., Tsoumakas, G., Vlahavas, I.: An empirical study of lazy multilabel classification algorithms. In: Proc. 5th Hellenic Conference on Artificial Intelligence (SETN 2008). (2008)
44. Thabtah, F., Cowling, P., Peng, Y.: Mmac: A new multi-class, multi-label associative classification approach. In: Proceedings of the 4th IEEE International Conference on Data Mining, ICDM '04. (2004) 217–224
45. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence **97** (1997) 273–324

46. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In Fisher, D.H., ed.: Proceedings of ICML-97, 14th International Conference on Machine Learning, Nashville, US, Morgan Kaufmann Publishers, San Francisco, US (1997) 412–420
47. Gao, S., Wu, W., Lee, C.H., Chua, T.S.: A MFoM learning approach to robust multiclass multi-label text categorization. In: Proceedings of the 21st international conference on Machine learning (ICML '04), Banff, Alberta, Canada (2004) 42
48. Park, C.H., Lee, M.: On applying linear discriminant analysis for multi-labeled problems. Pattern Recogn. Lett. **29** (2008) 878–887
49. Yu, K., Yu, S., Tresp, V.: Multi-label informed latent semantic indexing. In: SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, Salvador, Brazil, ACM Press (2005) 258–265
50. Zhang, Y., Zhou, Z.H.: Multi-label dimensionality reduction via dependence maximization. In: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, AAAI Press (2008) 1503–1505
51. Ji, S., Tang, L., Yu, S., Ye, J.: Extracting shared subspace for multi-label classification. In: Proceedings of the 14th SIGKDD International Conferece on Knowledge Discovery and Data Mining, Las Vegas, USA (2008)
52. Sun, L., Ji, S., Ye, J.: Hypergraph spectral learning for multi-label classification. In: Proceedings of the 14th SIGKDD International Conferece on Knowledge Discovery and Data Mining, Las Vegas, USA (2008)
53. Ruepp, A., Zollner, A., Maier, D., Albermann, K., Hani, J., Mokrejs, M., Tetko, I., Güldener, U., Mannhaupt, G., Münsterkötter, M., Mewes, H.W.: The funcat, a functional annotation scheme for systematic classification of proteins from whole genomes. Nucleic Acids Res **32** (2004) 5539–5545
54. Cesa-Bianchi, N., Gentile, C., Zaniboni, L.: Incremental algorithms for hierarchical classification. Journal of Machine Learning Research **7** (2006) 31–54
55. Esuli, A., Fagni, T., Sebastiani, F.: Boosting multi-label hierarchical text categorization. Information Retrieval **11** (2008) 287–313
56. Tsoumakas, G., Katakis, I., Vlahavas, I.: Effective and efficient multilabel classification in domains with large number of labels. In: Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08). (2008) 30–44
57. Harris, M.A., Clark, J., Ireland, A., Lomax, J., Ashburner, M., Foulger, R., Eilbeck, K., Lewis, S., Marshall, B., Mungall, C., Richter, J., Rubin, G.M., Blake, J.A., Bult, C., Dolan, M., Drabkin, H., Eppig, J.T., Hill, D.P., Ni, L., Ringwald, M., Balakrishnan, R., Cherry, J.M., Christie, K.R., Costanzo, M.C., Dwight, S.S., Engel, S., Fisk, D.G., Hirschman, J.E., Hong, E.L., Nash, R.S., Sethuraman, A., Theesfeld, C.L., Botstein, D., Dolinski, K., Feierbach, B., Berardini, T., Mundodi, S., Rhee, S.Y., Apweiler, R., Barrell, D., Camon, E., Dimmer, E., Lee, V., Chisholm, R., Gaudet, P., Kibbe, W., Kishore, R., Schwarz, E.M., Sternberg, P., Gwinn, M., Hannick, L., Wortman, J., Berriman, M., Wood, V., de La, Tonellato, P., Jaiswal, P., Seigfried, T., White, R.: The gene ontology (go) database and informatics resource. Nucleic Acids Res **32** (2004)
58. Vens, C., Struyf, J., Schietgat, L., Džeroski, S., Blockeel, H.: Decision trees for hierarchical multi-label classification. Machine Learning (2008)
59. Chawla, N.V., Japkowicz, N., Kotcz, A.: Editorial: special issue on learning from imbalanced data sets. SIGKDD Explorations **6** (2004) 1–6
60. Loza Mencia, E., Fu"rnkranz, J.: Efficient pairwise multilabel classification for large scale problems in the legal domain. In: 12th European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD 2008, Antwerp, Belgium (2008) 50–65
61. Diplaris, S., Tsoumakas, G., Mitkas, P., Vlahavas, I.: Protein classification with multiple algorithms. In: Proceedings of the 10th Panhellenic Conference on Informatics (PCI 2005), Volos, Greece (2005) 448–456
62. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: Rcv1: A new benchmark collection for text categorization research. J. Mach. Learn. Res. **5** (2004) 361–397
63. Srivastava, A., Zane-Ulman, B.: Discovering recurring anomalies in text reports regarding complex space systems. In: IEEE Aerospace Conference. (2005)

64. Yang, Y.: An evaluation of statistical approaches to text categorization. Journal of Information Retrieval **1** (1999) 67–88
65. Jin, R., Ghahramani, Z.: Learning with multiple learning. In: Proceedings of Neural Information Processing Systems 2002 (NIPS 2002), Vancouver, Canada (2002)
66. Maron, O., p Erez, T.A.L.: A framework for multiple-instance learning. In: Advances in Neural Information Processing Systems 10, MIT Press (1998) 570–576
67. Zhou, Z.H.: Mining ambiguous data with multi-instance multi-label representation. In: Proceedings of the 3rd International Conference on Advanced Data Mining and Applications (ADMA'07). Springer (2007) 1
68. Zhou, Z.H., Zhang, M.L.: Multi-instance multi-label learning with application to scene classification. In Schölkopf, B., Platt, J.C., Hoffman, T., eds.: NIPS, MIT Press (2006) 1609–1616
69. Zha, Z.J., Hua, X.S., Mei, T., Wang, J., Qi, G.J., Wang, Z.: Joint multi-label multi-instance learning for image classification. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. (2008) 1–8
70. Caruana, R.: Multitask learning. Machine Learning **28** (1997) 41–75