Random k-Labelsets for Multilabel Classification

Grigorios Tsoumakas, *Member*, *IEEE*, Ioannis Katakis, and Ioannis Vlahavas, *Member*, *IEEE*

Abstract—A simple yet effective multilabel learning method, called label powerset (LP), considers each distinct combination of labels that exist in the training set as a different class value of a single-label classification task. The computational efficiency and predictive performance of LP is challenged by application domains with large number of labels and training examples. In these cases, the number of classes may become very large and at the same time many classes are associated with very few training examples. To deal with these problems, this paper proposes breaking the initial set of labels into a number of small random subsets, called *labelsets* and employing LP to train a corresponding classifier. The labelsets can be either disjoint or overlapping depending on which of two strategies is used to construct them. The proposed method is called RAkEL (RAndom k labELsets), where k is a parameter that specifies the size of the subsets. Empirical evidence indicates that RAkEL manages to improve substantially over LP, especially in domains with large number of labels and exhibits competitive performance against other high-performing multilabel learning methods.

Index Terms—Categorization, multilabel, ensembles, labelset, classification.

1 Introduction

Traditional single-label classification is concerned with learning from a set of data that are associated with a single label λ from a set of disjoint labels L of size M, with $M > 1$. If $M = 2$, then the learning task is called binary classification, concept learning, or filtering, while if $M > 2$, then it is called multiclass classification.

In several application domains, however, data are associated with a set of labels $Y \subseteq L$. In text categorization for example, a newspaper article concerning the reactions of the Christian church to the release of the "Da Vinci Code" film can be classified into both of the categories *society\religion* and *arts\movies*. Similarly, in semantic scene classification [1], [2], [3], a photograph can belong to more than one conceptual class, such as *sunset* and *beach* at the same time. Other interesting applications of multilabel classification include music categorization into emotions [4], [5], [6], semantic video annotation [7], [8], direct marketing [9], and automated tag suggestion [10], [11].

This paper focuses on the label powerset (LP) multilabel learning method [1], [12], which considers each subset of L, hitherto called *labelset*, that exists in the training set as a different class value of a single-label classification task. LP is an interesting approach to study, as it has the advantage of taking label correlations into consideration. This way it can, in some cases, achieve better performance compared to computationally simpler approaches like binary relevance (BR), which learns a binary model for each label independently of the rest [13].

• *The authors are with the Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece.*
E-mail: {greg, katak, vlahavas}@csd.auth.gr.

Manuscript received 26 Apr. 2009; revised 30 Oct. 2009; accepted 3 Mar. 2010; published online 3 Sept. 2010.
Recommended for acceptance by L. Wang.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2009-04-0377.
Digital Object Identifier no. 10.1109/TKDE.2010.164.

However, LP is challenged by application domains with large number of labels and training examples, due to the typically proportionally large number of labelsets appearing in the training set. The large number of these labelsets (class values for the single-label classifier of LP), raises the computational cost of LP on one hand, and makes its learning task quite hard on the other, as many of these labelsets are usually associated with very few training examples. Moreover, LP can only predict labelsets observed in the training set. This is an important limitation, because new labelsets typically do appear in test sets, which simulate unseen data.

In order to deal with the aforementioned problems of LP, this work proposes randomly breaking the initial set of labels into a number of small-sized labelsets, and employing LP to train a corresponding multilabel classifier. This way, the resulting single-label classification tasks are computationally simpler and the distribution of their class values is less skewed. The proposed method is called RAkEL (RAndom k labELsets) [13], where k is a parameter that specifies the size of the labelsets. Two different strategies for constructing the labelsets are studied. The first one leads to *disjoint*, whereas the second to *overlapping* labelsets.

Empirical evidence indicates that both approaches manage to significantly improve LP, especially in domains with large number of labels. The overlapping strategy achieves higher predictive performance than the disjoint one, as the aggregation of multiple predictions for each label via voting allows the correction of potential uncorrelated errors. Finally, a comparative study against other multilabel learning methods, indicates that RAkEL with overlapping labelsets is highly competitive.

The rest of this paper is structured as follows: The following section presents related work on learning from multilabel data. Section 3 discusses the motivations and rationale for using RAkEL and describes the two alternative strategies for creating the labelsets in detail. Section 4 describes the data sets that are involved in the experiments

and the evaluation measures, while Section 5 presents and discusses the results. Finally, the last section highlights the main points of this work and presents its conclusions.

2 Related Work

Multilabel learning methods can be grouped into two categories [12]: 1) *problem transformation*, and 2) *algorithm adaptation*. Methods of the first group transform the learning task into one or more single-label classification or ranking tasks, for which a large bibliography of learning algorithms exists. The second group of methods extend specific learning algorithms in order to handle multilabel data directly.

2.1 Problem Transformation Methods

The following paragraphs describe a number of problem transformation methods from the literature. For the formal description of these methods, we will use $L = \{\lambda_j : j = 1 \dots M\}$ to denote the finite set of labels in a multilabel learning task and $D = \{(\vec{x_i}, Y_i), i = 1 \dots N\}$ to denote a set of multilabel training examples, where $\vec{x_i}$ is the feature vector and $Y_i \subseteq L$ the set of labels of the ith example.

There exist several transformations that can be used to convert a multilabel data set into a single-label one, where a single-label classifier that outputs a probability distribution over the classes can be applied in order to learn a label ranker [1], [14]. The *copy* transformation replaces each multilabel 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 multilabel example.

Binary relevance 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 $D_{\lambda_j}, j = 1 \dots M$ 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 predicted by the M classifiers.

Label powerset is a simple but effective problem transformation method that works as follows: It considers each unique set of labels that exists in a multilabel training set as one of the classes of a new single-label classification task. Given a new instance, the single-label classifier of LP outputs the most probable class, which actually represents a set of labels.

Ranking by pairwise comparison (RPC) [15] transforms the multilabel data set into $\frac{M(M-1)}{2}$ binary label data sets, one for each pair of labels $(\lambda_i, \lambda_j), 1 \leq i < j \leq M$. Each data set 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. The multilabel pairwise perceptron (MLPP) algorithm [16] is an instantiation of RPC using perceptrons for the binary classification tasks.

Calibrated label ranking (CLR) [17] extends RPC by introducing an additional virtual label, which acts as a natural breaking point of the ranking into a relevant and an irrelevant set of labels. The binary models that learn to discriminate between the virtual label and each of the other labels, correspond to the models of BR. This occurs, because each example that is annotated with a given label is considered as positive for this label and negative for the virtual label, while each example that is not annotated with a label is considered negative for it and positive for the virtual label.

2.2 Algorithm Adaptation Methods

The following paragraphs briefly report a plethora of algorithm adaptation methods grouped by the learning paradigm that they extend.

Decision Trees and Boosting. The C4.5 algorithm was adapted in [18] for the handling of multilabel data. AdaBoost.MH and AdaBoost.MR [19] are two extensions of AdaBoost for multilabel data. A combination of AdaBoost.MH with an algorithm for producing alternating decision trees was presented in [20]. The main motivation was the production of multilabel models that can be understood by humans.

Probabilistic Methods. A probabilistic generative model for multilabel text classification is proposed in [21], according to which, each label generates different words. Based on this model, a multilabel document is produced by a mixture of the word distributions of its labels. A similar word-based mixture model is presented in [22]. A deconvolution approach is proposed in [23], in order to estimate the individual contribution of each label to a given item. The use of conditional random fields is explored in [24], where two graphical models that parameterize label co-occurrences are proposed.

Neural Networks and Support Vector Machines. BP-MLL [25] is an adaptation of the popular back-propagation algorithm for multilabel learning. The main modification to the algorithm is the introduction of a new error function that takes multiple labels into account. This error function is similar to the ranking loss [19]. ML-RBF [26] is a recent approach for adapting radial basis function networks to multilabel data. The multiclass multilabel perceptron (MMP) [27] is a family of online algorithms for label ranking from multilabel 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 is proposed in [28].

Lazy and Associative Methods. A number of methods [29], [5], [30], [2], [31] 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. MMAC [32] is an algorithm that follows the paradigm of associative classification, which deals with the construction of classification rule

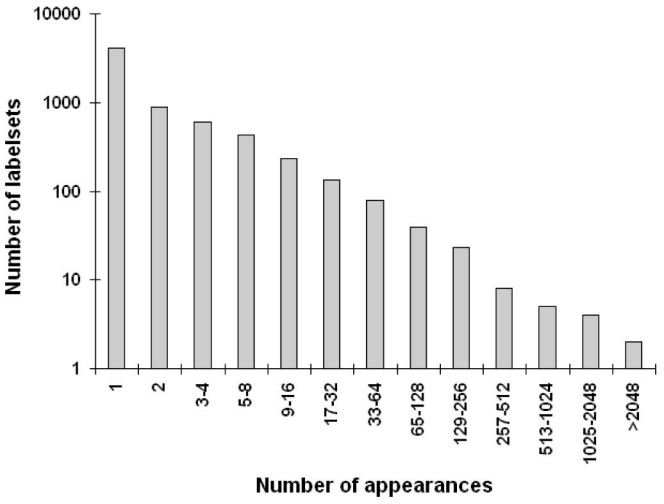

Fig. 1. Histogram of the number of appearances of the 6,555 different labelsets in the mediamill data set. The counts in the y-axis (logarithmic scale with base 10) correspond to the number of labelsets that exhibit the corresponding bin of appearances in the x-axis (logarithmic scale with base 2).

sets using association rule mining. Finally, an approach that combines lazy and associative learning is proposed in [33], where the inductive process is delayed until an instance is given for classification.

3 Random k-Labelsets

Label powerset is a relatively simple method with the advantage of taking label correlations into account. However, as briefly mentioned in the introduction of this paper, it is challenged by domains with large number of labels, M, and training examples, N.

The computational complexity of LP with respect to M and N depends on the computational complexity of the underlying single-label classification algorithm with respect to the number of examples N and the number of classes, which is equal to the number of labelsets that are used as annotations for the instances of the training set. This number is upper bounded by $\min(N, 2^M)$, but is usually much smaller in practice. Columns *bound* and *actual* of Table 2 show the bound of the number of labelsets and their actual number, for the data sets that are used in the experiments. Note that apart from the data set with the fewest number of labels (6), in the rest the bound is equal to N. The actual number of labelsets in a data set ranges from 5 to 44 percent of this bound, as indicated by column *diversity*, which measures the diversity of the labelsets that exist in a data set. Despite being smaller than the bound, the high number of these labelsets can constitute an important scalability problem for LP, especially for large values of N and M.

In addition, the fact that many of these labelsets are associated with very few examples, makes the learning process difficult as well. As an example, consider the *mediamill* data set [8], which is described in more detail in Section 4.1 along with the rest of the data sets that are used in the experiments. Fig. 1 shows a histogram of the number of appearances of the 6,555 different labelsets that exist in this data set. The total number of appearances (examples in the data set) is 43,907. The y-axis (logarithmic scale with base 10) shows the number of labelsets, whose number of appearances falls into the corresponding bin of the x-axis (logarithmic scale with base 2). It can be seen that most of the labelsets are very infrequent. In fact, 4,104 labelsets appear just once in this data set, while those that appear up to eight times account for 92 percent of all 6,555 distinct labelsets.

The main idea in this work is to randomly break a large set of labels into a number of small-sized labelsets, and for each of them train a multilabel classifier using the LP method. For the multilabel classification of an unlabeled instance, the decisions of all LP classifiers are gathered and combined. For simplicity, we only consider labelsets of the same size, k. A labelset $R \subseteq L$ with $k = |R|$ is called k-labelset. Therefore, the proposed approach is dubbed RAkEL (RAndom k labELsets). This paper examines the construction of two different types of labelsets: 1) disjoint (RAkEL$_d$), and 2) overlapping (RAkEL$_o$). In the following two sections, we describe the functionality of both variations of RAkEL in more detail.

RAkEL offers advantages over LP for the following reasons. First of all, the resulting single-label classification tasks are computationally simpler. To see why this occurs, consider the case of the *mediamill* data set again. If we break the set of 101 labels of *mediamill* into a number of labelsets of size $k = 3$, then each LP model will have to predict eight (2^3) classes in the worst case. If we construct 200 such models, training each of them using a one-versus-rest support vector machine, then 1,600 ($200 * 8$) binary models will be built in the worst case, which is much less compared to the 6,555 binary models required by the full LP using the same underlying learning algorithm. On the other hand, using a decision tree learning algorithm underneath will probably lead RAkEL to a larger overall computational cost, as decision tree learners are sublinear with respect to the number of classes. A complexity analysis is presented in Section 3.3.

In addition, the resulting single-label classification tasks are characterized by a much more balanced distribution of class values. Using the same example as above, we can see that the distribution of the eight class values of each simpler problem, will not be as skewed as that of the 6,555 class values that we have seen in Fig. 1. This in turns means an easier single-label learning problem to deal with. Note that this benefit is independent of the single-label learning algorithm used underneath.

Finally, in the case of overlapping labelsets, RAkEL can gather multiple predictions for the same label by the different LP models that include this label in their labelset. As the different LP models are trained on a different output space (different class labels), they offer a diverse view of the task of predicting the value for specific labels. Therefore, combining their output through a voting process, offers the chance of correcting potential uncorrelated errors, and improving the overall performance. In this respect, RAkEL reminds ensemble methods, like ECOC (error correcting output codes) [34], that construct multiple single-label models by manipulating the output space [35].

3.1 RAkEL$_d$

Given a size of labelsets k, RAkEL$_d$ initially partitions L randomly into $m = \lceil M/k \rceil$ disjoint labelsets R_j, $j = 1 \ldots m$,

Input: Set of labels L of size M, training set D, labelset size k
Output: Number of models m, k-labelsets R_i, corresponding LP classifiers h_i
$m = \lceil M/k \rceil$;
for *i=1 to m* **do**
  $r_i \leftarrow \emptyset$;
  for *j=1 to k* **do**
    if $L = \emptyset$ **then**
      break;
    $\lambda_j \leftarrow$ randomly selected label from L;
    $R_i \leftarrow R_i \cup \{\lambda_j\}$;
    $L \leftarrow L \setminus \{\lambda_j\}$;
  train an LP classifier h_i based on D and R_i;

Fig. 2. The training process of $\mathrm{RA}k\mathrm{EL}_d$.

$\bigcap_{j=1}^{m} R_j = \emptyset$. Labelsets R_j, $j = 1 \ldots m-1$ are k-labelsets. If M/k is an integer, then labelset R_m is also a k-labelset, otherwise R_m contains the remaining $M \bmod k$ labels. Then $\mathrm{RA}k\mathrm{EL}_d$ learns m multilabel classifiers h_j, $j = 1 \ldots m$ using LP. Each classifier h_j confronts a single-label classification task having as class values all the subsets of R_j that are found in the training set.

The training set for h_j, denoted as D_j, contains all examples of the original training set annotated with the intersection of their original annotations and R_j: $D_j = \{(\vec{x}_i, Y_i \cap R_j), i = 1 \ldots N\}$. Note that this may lead to the empty set appearing as an annotation for an example. This doesn't mean that these examples are excluded from D_j. The empty set is just another class of the single-label classification task of h_j. Actually, the empty set is an acceptable annotation based on the definition of a multilabel data set ($Y_i \subseteq L$) and most multilabel learners (including LP) handle it without any special consideration. Fig. 2 offers an algorithmic presentation of the training process of $\mathrm{RA}k\mathrm{EL}_d$.

Given a new multilabel instance $\vec{x}$, the binary predictions $h_i(\vec{x}, \lambda_j)$ of all classifiers h_i for all labels $\lambda_j \in R_i$ are gathered in order to build the final multilabel classification vector (see Fig. 3). Note that it is possible for $\mathrm{RA}k\mathrm{EL}_d$ to predict a labelset that has not appeared in the training set, as its final prediction is assembled from different parts of existing labelsets.

3.2 $\mathrm{RA}k\mathrm{EL}_o$

We first introduce some additional notation. Let the term L^k denote the set of all distinct k-labelsets of L. The size of

Input: Number of models m, new instance $\vec{x}$, k-labelsets R_i, corresponding LP classifiers h_i
Output: Multi-label classification vector $Result$
for *i=1 to m* **do**
  forall $\lambda_j \in R_i$ **do**
    $Result_j \leftarrow h_i(x, \lambda_j)$;

Fig. 3. The classification process of $\mathrm{RA}k\mathrm{EL}_d$.

Input: Set of labels L of size M, training set D, labelset size k, number of models $m \leq \binom{M}{k}$
Output: k-labelsets R_i, corresponding LP classifiers h_i
$S \leftarrow L^k$;
for $i \leftarrow 1$ **to** $\min(m, |L^k|)$ **do**
  $R_i \leftarrow$ a k-labelset randomly selected from S;
  train an LP classifier h_i based on D and R_i;
  $S \leftarrow S \setminus \{R_i\}$;

Fig. 4. The training process of $\mathrm{RA}k\mathrm{EL}_o$.

L^k is given by the binomial coefficient: $|L^k| = \binom{M}{k}$. Given a size of labelsets k and a number of desired classifiers $m \leq |L^k|$, $\mathrm{RA}k\mathrm{EL}_o$ initially selects m k-labelsets R_i, $i = 1 \ldots m$ from the set L^k via random sampling without replacement. Note that in this case the labelsets may overlap, while the overlap is certain when $mk > M$. Then $\mathrm{RA}k\mathrm{EL}_o$ learns m multilabel classifiers h_i, $i = 1 \ldots m$ using LP, as in the case of $\mathrm{RA}k\mathrm{EL}_d$. Fig. 4 presents the training process of $\mathrm{RA}k\mathrm{EL}_o$ in pseudocode.

For the multilabel classification of a new instance $\vec{x}$, each model h_i provides binary predictions $h_i(\vec{x}, \lambda_j)$ for each label λ_j in the corresponding k-labelset R_i. Subsequently, $\mathrm{RA}k\mathrm{EL}_o$ calculates the mean of these predictions for each label $\lambda_j \in L$ and outputs a final positive decision if it is greater than a 0.5 threshold. This intuitive threshold corresponds to the majority voting rule for the fusion of classifier decisions. It has been used for deriving a final decision in the problem transformation method RPC [15] as well. The pseudocode of the classification process of $\mathrm{RA}k\mathrm{ELo}$ is given in Fig. 5, while Table 1 exemplifies it for a run with $k = 3$ and $m = 7$ on a multilabel training set with six labels $\{\lambda_1, \lambda_2, \ldots, \lambda_6\}$. Similarly to $\mathrm{RA}k\mathrm{EL}_d$, $\mathrm{RA}k\mathrm{EL}_o$ can also predict a labelset that is not present in the training set, as its final prediction is obtained through a voting process,

Input: Set of labels L of size M, number of models m, k-labelsets R_i, corresponding LP classifiers h_i, new instance $\vec{x}$
Output: Multi-label classification vector $Result$
for $j \leftarrow 1$ **to** M **do**
  $Sum_j \leftarrow 0$;
  $Votes_j \leftarrow 0$;
for $i \leftarrow 1$ **to** m **do**
  forall *labels* $\lambda_j \in R_i$ **do**
    $Sum_j \leftarrow Sum_j + h_i(\vec{x}, \lambda_j)$;
    $Votes_j \leftarrow Votes_j + 1$;
for $j \leftarrow 1$ **to** M **do**
  $Avg_j \leftarrow Sum_j / Votes_j$;
  if $Avg_j > 0.5$ **then**
    $Result_j \leftarrow 1$;
  else $Result_j \leftarrow 0$;

Fig. 5. The classification process of $\mathrm{RA}k\mathrm{EL}_o$.

TABLE 1
An Example of the Classification Process of $\mathrm{RA}k\mathrm{EL}_o$ Run with $k = 3$ and $m = 7$ on a Multilabel Training Set with Six Labels

		predictions					
model	labelset	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
h_1	$\{\lambda_1,\lambda_2,\lambda_6\}$	1	0	-	-	-	1
h_2	$\{\lambda_2,\lambda_3,\lambda_4\}$	-	1	1	0	-	-
h_3	$\{\lambda_3,\lambda_5,\lambda_6\}$	-	-	0	-	0	1
h_4	$\{\lambda_2,\lambda_4,\lambda_5\}$	-	0	-	0	0	-
h_5	$\{\lambda_1,\lambda_4,\lambda_5\}$	1	-	-	0	1	-
h_6	$\{\lambda_1,\lambda_2,\lambda_3\}$	1	0	1	-	-	-
h_7	$\{\lambda_1,\lambda_4,\lambda_6\}$	0	-	-	1	-	0
	average votes	3/4	1/4	2/3	1/4	1/3	2/3
	final prediction	1	0	1	0	0	1

based on predictions that correspond to different overlapping parts of existing labelsets.

Note that user-specified parameters m (number of classifiers) and k (size of labelsets) determine the expected number of predictions for each label, which is equal to $\frac{km}{M}$. We hypothesize that the larger the expected number of predictions per label, the larger the predictive accuracy of $\mathrm{RA}k\mathrm{EL}_o$, due to the fusion of more predictions. Given that k should be small to avoid the problems of LP, then in order to increase km, one should select a large value for the number of models, m. The empirical study that follows, studies the relationship of m and k and provides guidelines for appropriate values.

3.3 Complexity Analysis

If the complexity of the algorithm employed to learn the transformed single-label classification task is $O(g(C, N, A))$ for a data set with C class values, N examples, and A predictive attributes, then the computational complexity of $\mathrm{RA}k\mathrm{EL}$ is $O(mg(min(N, 2^k), N, A))$, where $m = \lceil M/k \rceil$ in the case of disjoint labelsets. It is linear with respect to the number of LP classifiers and it further depends on the complexity of the single-label classification algorithm.

The number of LP classifiers, m, is linear with respect to M in the case of disjoint labelsets. The empirical study that follows, indicates that in the case of overlapping labelsets a value of m that is linear with respect to M (e.g., $2M$) suffices for reaching a high level of predictive performance. Finally, note that the exponential factor 2^k does not pose a problem, as k will be set to a small value.

4 Experimental Setup

This section provides details on the experimental setup. More specifically, Section 4.1 describes the data sets and Section 4.2 the measures that were used to empirically evaluate the performance of the proposed approach.

4.1 Data Sets

Experiments were conducted on eight multilabel data sets.[1] Table 2 includes basic statistics, such as the number of examples and labels, along with statistics that are relevant to labelsets, such as their bound, actual number, and diversity. Short descriptions of these data sets are given in the following paragraphs.

1. Available at http://mulan.sourceforge.net/datasets.html.

TABLE 2
Multilabel Data Sets and Their Statistics Sorted by Increasing Order of Number of Labels

name	examples	labels	bound of labelsets	actual labelsets	labelset diversity
scene	2407	6	64	15	23%
yeast	2417	14	2417	198	8%
tmc2007	28596	22	28596	1341	5%
medical	978	45	978	94	10%
enron	1702	53	1702	753	44%
mediamill	43907	101	43907	6555	15%
reuters	6000	101	6000	1028	17%
bibtex	7395	159	7395	2856	39%

The *scene* data set contains 2,407 images annotated with up to six concepts such as *beach*, *mountain,* and *field* [1]. Each image is described with 294 visual features.

The *yeast* data set [28] contains microarray expressions and phylogenetic profiles for 2,417 yeast genes. Each gene is annotated with a subset of 14 functional categories (e.g., *metabolism*, *energy*, etc) from the top level of the functional catalogue (FunCat).

The *tmc2007* data set is based on the data of the competition organized by the text mining workshop of the seventh SIAM international conference on data mining.[2] The original data contained 28,596 aviation safety reports in free text form, annotated with one or more out of 22 problem types that appear during flights [36]. Text representation follows the boolean bag-of-words model. Feature selection was then in order to reduce the computational cost of training. We used the χ^2 feature ranking method separately for each label in order to obtain a ranking of all features for that label. We then selected the top 500 features based on the their maximum rank over all labels. A similar approach was found to have high performance in previous experimental work on textual data sets [37].

The *medical* data set[3] is based on the data made available during the Computational Medicine Center's 2007 Medical Natural Language Processing Challenge.[4] The data set consists of 978 clinical free text reports labeled with one or more out of 45 disease codes.

The *enron* data set is based on a collection of email messages exchanged between the Enron Corporation employees, which was made available during a legal investigation. It contains 1,702 email messages that were categorized into 53 topic categories, such as *company strategy*, *humor,* and *legal advice*, by the UC Berkeley Enron Email Analysis Project.[5]

The *mediamill* data set was part of the Mediamill challenge for automated detection of semantic concepts in 2006 [8]. It contains 43,907 video frames annotated with 101 concepts (e.g., *military*, *desert*, and *basketball*, etc). The specific data set we used corresponds to experiment 1 (visual feature extraction) as described in [8]. Each video frame is characterized by a set of 120 visual features.

The *bibtex* data set [10] is based on the data of the ECML/PKDD 2008 discovery challenge. It contains 7,395 bibtex

2. http://www.cs.utk.edu/tmw07/.
3. Originally obtained from http://www.cs.waikato.ac.nz/~jmr30/.
4. http://www.computationalmedicine.org/challenge/index.php.
5. http://bailando.sims.berkeley.edu/enron_email.html.

entries from the BibSonomy social bookmark and publication sharing system, annotated with a subset of the tags assigned by BibSonomy users (e.g., *statistics*, *quantum*, and *datamining*). The title and abstract of bibtex entries were used to construct features using the boolean bag-of-words model.

The *reuters* (rcv1) data set is a well-known benchmark for text classification methods. We have used an existing subset of this data set (subset1) that contains 6,000 news articles assigned into one or more out of 101 topics. In order to reduce the feature space, we have kept only words that appear at least 50 times in the corpus leading to a reduced feature space of 1,654 features. An extensive description of this data set can be found in [38].

4.2 Evaluation Measures

The evaluation of multilabel learning methods requires different measures than those used in the case of single-label data. A unified presentation and categorization of existing evaluation measures for multilabel classification is given in [13]. The evaluation in this work is based on the popular and indicative micro F_1 and macro F_1 measures.

The F_1 measure is the harmonic mean of precision and recall and is a popular evaluation measure in the research area of information retrieval. Formally, given the number of true positives (tp), true negatives (tn), false positives (fp), and false negatives (fn), F_1 is defined as follows:

$$F_1 = \frac{2 * tp}{2 * tp + fp + fn}. \quad (1)$$

Micro F_1 and macro F_1 are the microaveraged and macroaveraged versions of F_1, respectively. Microaveraging as well as macroaveraging [39] are ways to calculate binary evaluation measures across several labels. Consider a binary evaluation measure $B(tp, tn, fp, fn)$. 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 microaveraged and macroaveraged versions of B, are calculated as follows:

$$B_{\text{micro}} = B\left(\sum_{\lambda=1}^{M} tp_\lambda, \sum_{\lambda=1}^{M} fp_\lambda, \sum_{\lambda=1}^{M} tn_\lambda, \sum_{\lambda=1}^{M} fn_\lambda\right), \quad (2)$$

$$B_{\text{macro}} = \frac{1}{M}\sum_{\lambda=1}^{M} B(tp_\lambda, fp_\lambda, tn_\lambda, fn_\lambda). \quad (3)$$

5 Results and Discussion

This section presents the results obtained from our empirical study and concludes on the applicability and performance of RAkEL. The first part studies the performance of RAkEL$_d$ and RAkEL$_o$ with respect to their parameters and compares them with LP and each other. Then they are compared against baseline and high-performing multilabel classifiers. Finally, three different single-label classification algorithms are used to study their effect on the performance of RAkEL$_o$.

In all experiments, RAkEL$_d$ and RAkEL$_o$ are run 10 times using different seed values for the initialization of the pseudorandom number generator that guides the selection of the labelsets, in order to obtain more representative results on one hand and assess the effect of the stochastic component of the algorithm on the other.

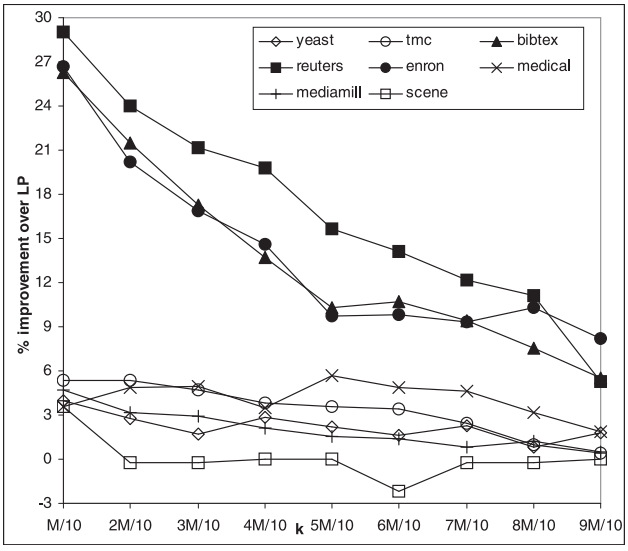

Fig. 6. Percentage of Micro F_1 Measure Improvement of RAkEL$_d$ over LP with Respect to k.

5.1 Empirical Evaluation of RAkEL

In this section, the evaluation is based on the micro F_1 measure, estimated via the holdout method using the original train and test subsets provided with the releases of the eight data sets.

The first two parts of this section examine the performance of the two RAkEL variations with respect to their parameters, which for RAkEL$_d$ is the size of the labelset, k, while for RAkEL$_o$ is both k and the number of models, m. Instead of absolute micro F_1 values, the percentage of improvement over LP is shown. This improves the legibility of the figures and simplifies the interpretation of the results across several different data sets. The last part compares the absolute micro F_1 performance of RAkEL$_d$ and RAkEL$_o$ using specific parameters against LP and each other.

The C4.5 decision tree learning algorithm was used as the base-level single-label classification algorithm of LP and the LP classifiers of RAkEL. We used the implementation of C4.5 within Weka [40] and our implementations of LP and RAkEL within Mulan [13].

5.1.1 Evaluation of RAkEL$_d$

Fig. 6 presents the percentage of improvement of RAkEL$_d$ over LP in terms of micro F_1 measure with respect to the size of the labelset (k) in all data sets. The values of k correspond to different fractions of the total number of labels (from $k = \lceil M/10 \rceil$ to $k = \lceil 9M/10 \rceil$), so that the effect of k is studied for a broad range of values and at the same time results are comparable across all data sets. When $k = M$, RAkEL$_d$ becomes equivalent to LP.

A first observation is that RAkEL$_d$ provides a substantial improvement over the LP classifier for all data sets with the exception of *scene*, where it leads to worse performance for $k = \lceil 6M/10 \rceil = 4$. However, this was an expected result due to the small number of labels in this data set (6).

Concerning the effect of k on the predictive performance of RAkEL$_d$, we could argue that in general smaller values of

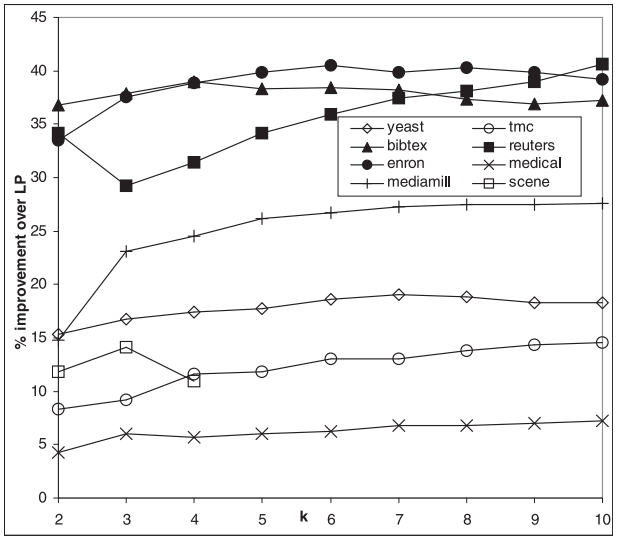

Fig. 7. Percentage of Micro F_1 Improvement of $\mathrm{RA}k\mathrm{EL}_o$ ($m = 2M$) over LP with Respect to k.

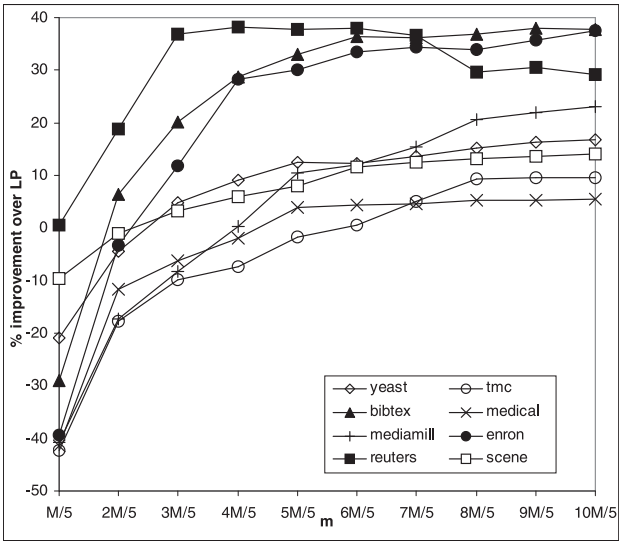

Fig. 8. Percentage of Micro F_1 Measure Improvement for $\mathrm{RA}k\mathrm{EL}$ ($k = 3$) over LP with Respect to m.

k usually lead to better results. This confirms the hypothesis made earlier that splitting the initial multilabel problem into a number of simpler and smaller subproblems will improve the performance of LP. On the other hand, greater values of k allow the LP models of $\mathrm{RA}k\mathrm{EL}_d$ to take larger labelsets (and potentially more correlations) into account. This is a plausible explanation for the fact that $\mathrm{RA}k\mathrm{EL}_d$'s performance is not always degrading with respect to k. Nevertheless, values of k that are close to M perform worst and approximate the performance of LP.

Concerning the performance of $\mathrm{RA}k\mathrm{EL}_d$ with respect to the number of labels (M), we notice that the greatest improvement is achieved in data sets with large number of labels such as *reuters* (101 labels), *bibtex* (159 labels), and *enron* (53 labels), where the performance of LP is poor (micro F_1: 0.0979, 0.2897, 0.3953, respectively). An exception to this rule is *mediamill* (101 labels), where the improvement is similar to data sets with small M. This can be explained firstly by the higher performance of LP in this data set (micro F_1: 0.4539). Second, even the smallest k value in the graph ($\lceil M/10 \rceil$) is quite large for *mediamill* (equal to 10). If we select smaller values for k we will observe greater improvement over LP. For example $k = 2$ leads to an 11.23 percent improvement and $k = 3$ leads to an 8.9 percent improvement.

In conclusion, we could state that setting k to small values is expected to lead to substantial better results compared to LP, especially in data sets with large number of labels.

5.1.2 Evaluation of $\mathrm{RA}k\mathrm{EL}_o$

Fig. 7 presents the percentage of improvement of $\mathrm{RA}k\mathrm{EL}_o$ over LP in terms of micro F_1 measure with respect to the size of the labelset (k) in all data sets. Based on the conclusions of the previous section, we use small values for k (from 2 to 10). The number of models (m) is set to $2M$, so that each label appears in the output of $\mathrm{RA}k\mathrm{EL}_o$'s models approximately $2k$ times irrespectively of the number of labels in the data set. Note that the *scene* plot can not be extended more than $k = 4$ because the small number of labels (6) limits the number of different labelsets that can be created.

We first observe that $\mathrm{RA}k\mathrm{EL}_o$ outperforms LP for all values of k in all data sets. As k increases, the performance of $\mathrm{RA}k\mathrm{EL}_o$ exhibits an increasing trend in most of the data sets, in contrast to what we have seen for $\mathrm{RA}k\mathrm{EL}_d$ in the previous section. The reason is that in this experiment, bigger values of k lead to more votes for each label, as the number of models is constant ($2M$). In turn, more votes lead to more accurate estimates of the true value of each label. Finally, as in $\mathrm{RA}k\mathrm{EL}_d$, the improvement in performance is in general greater in data sets with larger number of labels.

Fig. 8 presents the percentage of micro F_1 improvement of $\mathrm{RA}k\mathrm{EL}_o$ over LP using $k = 3$ with respect to various values of m in all data sets. In order to improve the legibility of the figure, the values of m correspond to different fractions of the total number of labels (from $\lceil M/5 \rceil$ to $2M$). What we observe is that as the number of classifiers increases, so does the performance of the ensemble. As before, this is due to the fact that bigger values of m lead to more votes for each label. An important finding that holds for all data sets is that after a certain number of models, the performance of $\mathrm{RA}k\mathrm{EL}_o$ does not exhibit substantial improvement. In most cases M is a good approximation of this number.

In conclusion, increasing either m or k leads $\mathrm{RA}k\mathrm{EL}_o$ to improved performance. The main reason for this behavior is that the number of votes received for each label is proportional to km. Since the complexity of $\mathrm{RA}k\mathrm{EL}$ grows exponentially with respect to k, but only linearly with respect to m, it is more efficient to increase the value of m instead of k. As a guideline, we suggest using a small value for k (e.g., $k = 3$), and a value that is between M and $2M$ for m.

One thing to note, however, is that the process of selecting the labels is random, and as such, it could lead to the case, where none of the labelsets include one or more from the original set of labels. In this case, the algorithm will not be able to make a rational decision about the value of this (or these) particular label(s). In this respect, increasing m further, reduces the probability of a label receiving no votes from the models.

TABLE 3
Comparison of RAkEL$_d$ and RAkEL$_o$

	LP		RAkEL$_d$ (k = 3)			RAkEL$_o$ (k = 3, m = 2M)		
	micro F_1	classes	micro F_1	Imp. over LP	classes	micro F_1	Imp. over LP	classes
scene	58.75%	14	58.76± 0.49%	0.02%	10.8	67.02± 0.69%	14.08%	66.3
yeast	52.65%	164	54.11± 0.63%	2.78%	33.7	61.50± 0.62%	16.81%	207.3
tmc2007	78.79%	1172	83.03± 0.37%	5.38%	53.6	85.76± 2.53%	8.84%	317.4
medical	74.54%	77	78.81± 0.53%	5.72%	58.3	78.69± 1.04%	5.56%	351.8
enron	39.53%	545	49.94± 1.29%	26.33%	91.8	54.39± 0.62%	37.59%	554.7
mediamill	45.39%	4913	49.43± 0.25%	8.90%	176.7	55.80± 0.39%	22.94%	1054.7
reuters	9.79%	523	12.40± 0.16%	26.64%	151.4	12.71± 0.18%	29.80%	922.9
bibtex	28.97%	2058	39.07± 0.33%	34.85%	255.8	39.95± 0.27%	37.89%	1542.8

5.1.3 Disjoint versus Overlapping Labelsets

Table 3 presents the micro F_1 value of LP, RAkEL$_d$ ($k = 3$), and RAkEL$_o$ ($k = 3$, $m = 2M$). For RAkEL$_d$ and RAkEL$_o$ the percentage of improvement in micro F_1 over LP is also presented.

We observe that RAkEL$_o$ provides a higher improvement over LP than RAkEL$_d$. This fact confirms the assumption that ensemble voting will further enhance the overall performance. An exception to this observation is the *medical* data set, where RAkEL$_o$ provides a slightly smaller improvement. However, in this data set there is a very small number of distinct labelsets (94), despite the relatively large number of labels (45). This explains both the good performance of LP and the minor improvements of RAkEL$_o$ due to the small diversity of its ensemble of LP classifiers.

In Table 3, the number of classes resulting from the transformation process of the three methods is presented. For RAkEL$_d$ and RAkEL$_o$ the classes of all LP models are summed. The number of classes can be considered as an estimator of computational requirements. We observe that RAkEL$_d$ presents less classes than LP in all data sets. Note the substantial reduction in number of classes in data sets *tmc2007*, *enron*, *mediamill*, *reuters*, and *bibtex*. RAkEL$_o$, as expected, presents greater number of classes compared to RAkEL$_d$ due to the additional classfiers. However, in data sets *tmc2007*, *mediamill*, and *bibtex* it presents lower number of classes than LP.

5.2 Comparison with Other Methods

In this section, the evaluation is based on both the micro F_1 and the macro F_1 measures, estimated via 10 repeated holdout experiments, each using a random 66 percent of each data set for training and the rest for evaluation. So, in this case RAkEL$_d$ and RAkEL$_o$ are run a total of 100 times each, as 10 different seeds are used for each different holdout experiment. To calculate the performance of RAkEL for a specific holdout experiment, we average the values obtained from these 10 additional internal executions.

RAkEL is compared against the simple baseline methods BR and LP, as well as against three high-performing multilabel methods that have been found to perform better than a number of other multilabel methods in a variety of data sets. The first one is a multilabel version of the k nearest neighbors algorithm, called MLkNN [2]. The second one is a multilabel version of the back-propagation algorithm for training multilayer perceptrons, called BPMLL [25]. The last one is the pairwise comparison method, called calibrated label ranking (CLR) [17].

The C4.5 decision tree learning algorithm was used as the base-level single-label classification algorithm of BR, LP, the LP classifiers of RAkEL, and the binary classifiers of CLR. For RAkEL$_d$ we set k to 3 and for RAkEL$_o$ we set k to 3 and m to $2M$ in all data sets. Note that these are generic settings based on the conclusions of the previous section, and definitely not the optimal ones as also shown in the previous section. For MLkNN, the number of neighbors is set to 10 and the smoothing factor is set to 1 as recommended in [2]. For BPMLL, the learning rate is set to 0.05, the number of epochs is set to 100 and the number of hidden units is set to 20 percent of the input units, as recommended in [25]. We used the implementation of C4.5 within Weka [40] and our implementations of BR, LP, RAkEL, CLR, MLkNN, and BPMLL within Mulan [13] for unified experiments and evaluation.

Tables 4 and 6 present the average and standard deviation of the micro F_1 and macro F_1 measure, respectively, for all method-data set pairs. Tables 5 and 7 present the rank of each method in terms of micro F_1 and macro F_1, respectively, in each data set, along with the average rank of each method. Following the suggestions in [41], we compare the different methods according to their average rank.

TABLE 4
Comparative Results in Terms of Micro F_1

	BR	LP	RAkEL$_d$	RAkEL$_o$	MLkNN	BPMLL	CLR
scene	62.36± 1.01%	60.05± 1.14%	59.87± 0.82%	69.58± 1.53%	72.29± 1.08%	48.18± 5.19%	62.82± 0.92%
yeast	57.67± 1.89%	53.04± 1.03%	54.26± 0.58%	61.89± 0.74%	63.93± 1.06%	63.11± 1.47%	61.69± 1.29%
tmc2007	68.80± 0.28%	61.87± 0.30%	67.78± 0.32%	71.78± 0.22%	63.21± 0.52%	71.23± 0.23%	70.73± 0.32%
medical	79.34± 1.02%	74.37± 1.87%	79.21± 0.89%	79.34± 1.22%	66.23± 1.77%	60.40± 5.81%	78.67± 1.14%
enron	51.93± 1.93%	41.33± 1.00%	50.69± 0.48%	55.35± 0.94%	44.42± 3.00%	56.01± 1.15%	54.65± 1.24%
mediamill	55.35± 0.22%	49.21± 0.27%	54.28± 0.14%	60.81± 0.16%	58.15± 0.27%	49.30± 0.58%	58.51± 0.24%
reuters	37.12± 0.47%	29.88± 0.71%	36.31± 0.38%	38.21± 0.13%	20.10± 1.55%	05.99± 0.33%	32.74± 0.81%
bibtex	39.58± 0.74%	29.32± 0.69%	39.13± 0.67%	40.36± 0.71%	20.56± 0.94%	45.81± 0.45%	32.54± 0.70%

TABLE 5
Ranking of Methods According to Their Micro F_1 Performance (See Table 4)

	BR	LP	RAkEL$_d$	RAkEL$_o$	MLkNN	BPMLL	CLR
scene	4	5	6	2	1	7	3
yeast	5	7	6	3	1	2	4
tmc	4	7	5	1	6	2	3
medical	1	5	3	2	6	7	4
enron	4	7	5	2	6	1	3
mediamill	4	7	5	1	3	6	2
reuters	2	5	3	1	6	7	4
bibtex	3	6	4	2	7	1	5
average rank	3.38	6.13	4.63	1.75	4.50	4.13	3.50

We observe that RAkEL$_o$ exhibits the highest average rank both in the micro F_1 and the macro F_1 measures. RAkEL$_d$ presents the second worst average rank in terms of micro-F_1 (outperforming only LP) but the third best average rank in terms of macro F_1 outperforming LP, MLkNN, BPMLL, and CLR. It is interesting to notice that the baseline BR presents the second best average rank in both metrics.

In addition, the Wilcoxon signed-rank test ($\mathrm{a} = 0.05$) was applied in order to examine if RAkEL$_o$ (or even RAkEL$_d$) have a statistical significant advantage over the rest of the methods. Over all data sets, RAkEL$_o$ proved to outperform significantly BR, LP, RAkEL$_d$, and CLR in terms of micro F_1. On the other hand, RAkEL$_d$, proved significantly better than LP only. In terms of macro F_1, RAkEL$_o$ proved to be significantly better than LP, RAkEL$_d$, MLkNN, and CLR. RAkEL$_d$ significantly outperformed LP only, as for micro F_1.

5.3 The Effect of the Classification Algorithm

This section studies the effect of the single-label classification algorithm that is used to train the LP models of RAkEL$_o$ on the performance of RAkEL$_o$. We compare the performance of RAkEL$_o$ with $k = 3$ and $m = 2M$ using three different base-level learning algorithms: 1) the C4.5 algorithm that was used in the experiments so far, 2) a naive Bayes (NB) algorithm, and 3) a support vector machine (SVM) learning algorithm (one-versus-rest). We used the implementations of these algorithms that are available within Weka [40] (The Weka port of LibSVM [42] was used for the SVM). The SVM was set with polynomial kernel. We tuned the regularization (C) and degree (d) parameters of the SVM by searching the space defined by the values $d = \{1, 2, 3\}$ and $C = \{1, 10, 100\}$. As in Section 5.1, the evaluation here is also based on the micro F_1 measure, estimated via the holdout method using the original train and test subsets provided with the releases of the eight data sets.

TABLE 7
Ranking of Methods According to Their Macro F_1 Performance (See Table 6)

	BR	LP	RAkEL$_d$	RAkEL$_o$	MLkNN	BPMLL	CLR
scene	4	5	6	2	1	7	3
yeast	5	6	3	2	7	1	4
tmc	4	6	5	2	7	1	3
medical	2	5	3	1	7	6	4
enron	4	5	3	2	7	1	6
mediamill	3	4	2	1	5	7	6
reuters	1	4	3	2	6	7	5
bibtex	2	5	4	3	7	1	6
average rank	3.13	5.00	3.63	1.88	5.88	3.88	4.63

TABLE 8
Micro F_1 of RAkEL$_o$ Using Three Different Single-Label Classification Algorithms

dataset	NB	C4.5	SVM
scene	62.96± 0.38%	67.02± 0.69%	**72.59**± 0.21%
yeast	57.00± 0.65%	61.50± 0.62%	**64.88**± 0.37%
tmc2007	60.42± 1.86%	**85.76**± 2.53%	71.74± 0.35%
medical	46.54± 0.43%	**78.69**± 1.04%	78.05± 0.52%
enron	33.02± 0.72%	54.39± 0.62%	**55.81**± 0.32%
mediamill	17.42± 0.34%	**55.80**± 0.39%	47.37± 0.13%
reuters	8.88± 0.60%	**12.71**± 0.18%	3.39± 0.03%
bibtex	22.31± 0.10%	39.95± 0.27%	**41.66**± 0.11%

Table 8 presents the results of the experiments. The best performance at each data set is indicated with bold typeface. In general, C4.5 and SVM perform better than NB in almost all data sets. C4.5 and SVM achieve the best performance in four data sets each. This phenomenon can be explained by considering that SVMs are more accurate classifiers but, on the other hand, decision trees are well suited for training ensembles of classifiers as RAkEL$_o$ does.

6 Conclusions

This paper has presented a new multilabel classification method, called RAkEL, that learns an ensemble of LP classifiers, each one targeting a different small random subset of the set of labels. The motivation was the computational efficiency and predictive performance problems of the simple and effective standard LP method, when faced with domains with large number of labels and training examples.

We examined both disjoint and overlapping subsets and found that both lead to improved results over the standard LP method, especially in domains with many labels. We also found that overlapping subsets lead to better results compared to disjoint ones, due to the classifier fusion process that takes place for each label. The results of comparing the predictive performance of the proposed approach with three high-performing algorithm adaptation

TABLE 6
Comparative Results in Terms of Macro F_1

	BR	LP	RAkEL$_d$	RAkEL$_o$	MLkNN	BPMLL	CLR
scene	63.41± 0.91%	61.04± 1.16%	60.90± 0.88%	70.26± 1.64%	72.63± 1.37	51.29± 5.26%	64.23± 0.89%
yeast	38.29± 0.59%	37.36± 1.09%	38.84± 0.50%	40.66± 0.77%	36.34± 0.79	42.85± 1.02%	38.52± 0.96%
tmc2007	57.82± 0.52%	50.37± 0.47%	57.04± 0.54%	59.90± 0.41%	41.61± 1.08	61.88± 0.69%	58.59± 0.67%
medical	35.67± 2.27%	33.38± 1.94%	35.62± 2.07%	36.98± 1.81%	20.26± 1.36	22.83± 3.04%	34.63± 1.65%
enron	14.21± 0.74%	13.66± 1.43%	14.45± 0.55%	14.65± 0.63%	7.43± 1.18	17.04± 1.18%	13.10± 0.70%
mediamill	18.64± 0.61%	17.90± 0.33%	19.78± 0.38%	21.26± 0.50%	14.18± 0.23	9.86± 0.24%	11.81± 0.33%
reuters	21.81± 0.79%	16.81± 0.87%	21.12± 0.60%	21.25± 0.45%	8.40± 0.68	5.22± 0.08%	11.77± 0.57%
bibtex	27.12± 0.66%	20.39± 0.79%	26.40± 0.59%	26.77± 0.58%	6.42± 0.41	31.73± 0.32%	16.68± 0.53%

methods were in favor of the proposed approach (using overlapping subsets).

RAkEL could be more generally thought of as a new approach for creating an ensemble of multilabel classifiers by manipulating the label space using randomization. In this sense, RAkEL could be independent of the underlying method for multilabel learning, which in this paper is LP. However, we should note that only multilabel learning methods that strongly depend on the specific set of labels used to annotate each example, such as LP and PPT [43] (an extension of LP), are good candidates for this generalized version of RAkEL. BR for example wouldn't benefit at all, while MLkNN would only slightly be affected, as it is heavily based on the feature space (the nearest neighbors will always be the same for all different labelsets).

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for the valuable comments. We would also like to thank Dr. Ioannis Partalas for his implementation of the Wilcoxon signed-rank test. This work was partially supported by a PENED program (EPAN M.8.3.1, No. 03EΔ73), jointly funded by EU and the Greek General Secretariat of Research and Technology.

REFERENCES

[1] M. Boutell, J. Luo, X. Shen, and C. Brown, "Learning Multi-Label Scene Classification," *Pattern Recognition,* vol. 37, no. 9, pp. 1757-1771, 2004.

[2] M.-L. Zhang and Z.-H. Zhou, "Ml-Knn: A Lazy Learning Approach to Multi-Label Learning," *Pattern Recognition,* vol. 40, no. 7, pp. 2038-2048, 2007.

[3] C. Wang, S. Yan, L. Zhang, and H.-J. Zhang, "Multi-label Sparse Coding for Automatic Image Annotation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '09),* 2009.

[4] T. Li and M. Ogihara, "Toward Intelligent Music Information Retrieval," *IEEE Trans. Multimedia,* vol. 8, no. 3, pp. 564-574, 2006.

[5] A. Wieczorkowska, P. Synak, and Z. Ras, "Multi-Label Classification of Emotions in Music," *Proc. Int'l Conf. Intelligent Information Processing and Web Mining (IIPWM '06),* pp. 307-315, 2006.

[6] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas, "Multi-label Classification of Music into Emotions," *Proc. Ninth Int'l Conf. Music Information Retrieval (ISMIR '08),* 2008.

[7] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, T. Mei, and H.-J. Zhang, "Correlative Multi-Label Video Annotation," *Proc. MULTIMEDIA '07: 15th Int'l Conf. Multimedia,* pp. 17-26, 2007.

[8] C.G.M. Snoek, M. Worring, J.C. van Gemert, J.-M. Geusebroek, and A.W.M. Smeulders, "The Challenge Problem for Automated Detection of 101 Semantic Concepts in Multimedia," *Proc. MULTIMEDIA '06: 14th Ann. ACM Int'l Conf. Multimedia,* pp. 421-430, 2006.

[9] Y. Zhang, S. Burer, and W.N. Street, "Ensemble Pruning via Semi-Definite Programming," *J. Machine Learning Research,* vol. 7, pp. 1315-1338, 2006.

[10] I. Katakis, G. Tsoumakas, and I. Vlahavas, "Multilabel Text Classification for Automated Tag Suggestion," *Proc. European Conf. Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD '08) Discovery Challenge,* 2008.

[11] Y. Song, L. Zhang, and L.C. Giles, "A Sparse Gaussian Processes Classification Framework for Fast Tag Suggestions," *Proc. CIKM '08: 17th ACM Conf. Information and Knowledge Management,* pp. 93-102, 2008.

[12] G. Tsoumakas and I. Katakis, "Multi-Label Classification: An Overview," *Int'l J. Data Warehousing and Mining,* vol. 3, no. 3, pp. 1-13, 2007.

[13] G. Tsoumakas and I. Vlahavas, "Random K-Labelsets: An Ensemble Method for Multilabel Classification," *Proc. 18th European Conf. Machine Learning (ECML '07),* pp. 406-417, Sept. 2007.

[14] W. Chen, J. Yan, B. Zhang, Z. Chen, and Q. Yang, "Document Transformation for Multi-Label Feature Selection in Text Categorization," *Proc. Seventh IEEE Int'l Conf. Data Mining,* pp. 451-456, 2007.

[15] E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Bringer, "Label Ranking by Learning Pairwise Preferences," *Artificial Intelligence,* vol. 172, no. 16-17, pp. 1897-1916, Nov. 2008.

[16] E. Loza Mencia and J. Fürnkranz, "Pairwise Learning of Multilabel Classifications with Perceptrons," *Proc. IEEE Int'l Joint Conf. Neural Networks (IJCNN '08),* pp. 2900-2907, 2008.

[17] J. Fürnkranz, E. Hüllermeier, E.L. Mencia, and K. Brinker, "Multilabel Classification via Calibrated Label Ranking," *Machine Learning,* vol. 73, no. 2, pp. 133-153, Nov. 2008.

[18] A. Clare and R. King, "Knowledge Discovery in Multi-label Phenotype Data," *Proc. Fifth European Conf. Principles of Data Mining and Knowledge Discovery (PKDD '01),* pp. 42-53, 2001.

[19] Y. Schapire and R.E. Singer, "Boostexter: A Boosting-Based System for Text Categorization," *Machine Learning,* vol. 39, no. 2/3, pp. 135-168, 2000.

[20] F. de Comite, R. Gilleron, and M. Tommasi, "Learning Multi-Label Alternating Decision Trees from Texts and Data," *Proc. Third Int'l Conf. Machine Learning and Data Mining in Pattern Recognition (MLDM '03),* pp. 35-49, July 2003.

[21] A. McCallum, "Multi-Label Text Classification with a Mixture Model Trained by Em," *Proc. Am. Assoc. for Artificial Intelligence (AAAI '99) Workshop Text Learning,* 1999.

[22] N. Ueda and K. Saito, "Parametric Mixture Models for Multi-Labeled Text," *Advances in Neural Information Processing Systems,* vol. 15, pp. 721-728, 2003.

[23] A.P. Streich and J.M. Buhmann, "Classification of Multi-Labeled Data: A Generative Approach," *Proc. 12th European Conf. Principles and Practice of Knowledge Discovery in Databases (PKDD '08),* 2008.

[24] N. Ghamrawi and A. McCallum, "Collective Multi-Label Classification," *Proc. 14th ACM Int'l Conf. Information and Knowledge Management (CIKM '05),* pp. 195-200, 2005.

[25] M.-L. Zhang and Z.-H. Zhou, "Multi-Label Neural Networks with Applications to Functional Genomics and Text Categorization," *IEEE Trans. Knowledge and Data Eng.,* vol. 18, no. 10, pp. 1338-1351, Oct. 2006.

[26] M.-L. Zhang, "Ml-Rbf: Rbf Neural Networks for Multi-label Learning," *Neural Processing Letters,* vol. 29, no. 2, pp. 61-74, 2009.

[27] K. Crammer and Y. Singer, "A Family of Additive Online Algorithms for Category Ranking," *J. Machine Learning Research,* vol. 3, pp. 1025-1058, 2003.

[28] A. Elisseeff and J. Weston, "A Kernel Method for Multi-Labelled Classification," *Proc. Advances in Neural Information Processing Systems 14,* 2002.

[29] X. Luo and A. Zincir-Heywood, "Evaluation of Two Systems on Multi-Class Multi-Label Document Classification," *Proc. 15th Int'l Symp. Methodologies for Intelligent Systems,* pp. 161-169, 2005.

[30] K. Brinker and E. Hullermeier, "Case-Based Multilabel Ranking," *Proc. 20th Int'l Joint Conf. Artificial Intelligence (IJCAI '07),* pp. 702-707, Jan. 2007.

[31] E. Spyromitros, G. Tsoumakas, and I. Vlahavas, "An Empirical Study of Lazy Multilabel Classification Algorithms," *Proc. Fifth Hellenic Conf. Artificial Intelligence (SETN '08),* 2008.

[32] F. Thabtah, P. Cowling, and Y. Peng, "Mmac: A New Multi-class, Multi-Label Associative Classification Approach," *Proc. Fourth IEEE Int'l Conf. Data Mining, (ICDM '04),* pp. 217-224, 2004.

[33] A. Veloso, M.J. Wagner, M. Goncalves, and M. Zaki, "Multi-Label Lazy Associative Classification," *Proc. 11th European Conf. Principles and Practice of Knowledge Discovery in Databases (PKDD '07),* pp. 605-612, Sept. 2007.

[34] T.G. Dietterich and G. Bakiri, "Solving Multiclass Learning Problems via Error-Correcting Output Codes," *J. Artificial Intelligence Research,* vol. 2, pp. 263-286, 1995.

[35] T.G. Dietterich, "Ensemble Methods in Machine Learning," *Proc. First Int'l Workshop Multiple Classifier Systems,* pp. 1-15, 2000.

[36] A. Srivastava and B. Zane-Ulman, "Discovering Recurring Anomalies Inc Text Reports Regarding Complex Space Systems," *Proc. IEEE Aerospace Conf.,* 2005.

[37] M. Rogati and Y. Yang, "High-Performing Feature Selection for Text Classification," *Proc. CIKM '02: Eleventh Int'l Conf. Information and Knowledge Management*, pp. 659-661, 2002.
[38] D.D. Lewis, Y. Yang, T.G. Rose, and F. Li, "Rcv1: A New Benchmark Collection for Text Categorization Research," *J. Machine Learning Research*, vol. 5, pp. 361-397, 2004.
[39] Y. Yang, "An Evaluation of Statistical Approaches to Text Categorization," *J. Information Retrieval*, vol. 1, pp. 67-88, 1999.
[40] I.H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2005.
[41] J. Demsar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *J. Machine Learning Research*, vol. 7, pp. 1-30, 2006.
[42] C.-C. Chang and C.-J. Lin, *LIBSVM: A Library for Support Vector Machines*, Software http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001.
[43] J. Read, "A Pruned Problem Transformation Method for Multi-Label Classification," *Proc. New Zealand Computer Science Research Student Conf. (NZCSRS '08)*, pp. 143-150, 2008.

Grigorios Tsoumakas received the BSc degree in Informatics from Aristotle University of Thessaloniki (AUTH) in 1999, the MSc degree in Artificial Intelligence from the University of Edinburgh in 2000, and the PhD degree in Informatics from AUTH in 2005. He is a Lecturer at the Department of Informatics of AUTH. His research interests include various aspects of machine learning, knowledge discovery and data mining, including ensemble methods, distributed data mining, text classification, and multilabel learning. He is a member of the IEEE and the IEEE Computer Society.

Ioannis Katakis received the BSc degree in Informatics in 2004, the MSc degree in Information Systems in 2007, and the PhD degree in Informatics in 2009 from Aristotle University of Thessaloniki. His research interests include machine learning and data mining and in particular text stream classification and multilabel text classification. He is a member of the Machine Learning and Knowledge Discovery group (MLKD, http://mlkd.csd.auth.gr).

Ioannis Vlahavas received the PhD degree in logic programming systems from the Aristotle University of Thessaloniki in 1988. He is a professor at the Department of Informatics at the Aristotle University of Thessaloniki. During the first half of 1997 he has been a visiting scholar at the Department of CS at Purdue University. He specializes in logic programming, knowledge based, and AI systems and he has published more than 200 papers and book chapters, and coauthored eight books in these areas. He teaches logic programming, AI, expert systems, and DSS. He has been involved in more than 27 research and development projects, leading most of them. He was the chairman of the 2nd Hellenic Conference on AI and the host of the 2nd International Summer School on AI Planning. He is leading the Logic Programming and Intelligent Systems Group (LPIS Group, lpis.csd.auth.gr) (more information at www.csd.auth.gr/vlahavas). He is a member of the IEEE and the IEEE Computer Society.