Pattern Recognition 45 (2012) 3084–3104

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

An extensive experimental comparison of methods for multi-label learning

Gjorgji Madjarov [a,b,*], Dragi Kocev [b], Dejan Gjorgjevikj [a], Sašo Džeroski [b]

[a] Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University, Rugjer Boshkovikj 16, 1000 Skopje, Macedonia
[b] Department of Knowledge Technologies, Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia

ARTICLE INFO

Available online 13 March 2012

Keywords:
Multi-label ranking
Multi-label classification
Comparison of multi-label learning methods

ABSTRACT

Multi-label learning has received significant attention in the research community over the past few years: this has resulted in the development of a variety of multi-label learning methods. In this paper, we present an extensive experimental comparison of 12 multi-label learning methods using 16 evaluation measures over 11 benchmark datasets. We selected the competing methods based on their previous usage by the community, the representation of different groups of methods and the variety of basic underlying machine learning methods. Similarly, we selected the evaluation measures to be able to assess the behavior of the methods from a variety of view-points. In order to make conclusions independent from the application domain, we use 11 datasets from different domains. Furthermore, we compare the methods by their efficiency in terms of time needed to learn a classifier and time needed to produce a prediction for an unseen example. We analyze the results from the experiments using Friedman and Nemenyi tests for assessing the statistical significance of differences in performance. The results of the analysis show that for multi-label classification the best performing methods overall are random forests of predictive clustering trees (RF-PCT) and hierarchy of multi-label classifiers (HOMER), followed by binary relevance (BR) and classifier chains (CC). Furthermore, RF-PCT exhibited the best performance according to all measures for multi-label ranking. The recommendation from this study is that when new methods for multi-label learning are proposed, they should be compared to RF-PCT and HOMER using multiple evaluation measures.

1. Introduction

The problem of single-label classification is concerned with learning from examples, where each example is associated with a single label λ_i from a finite set of disjoint labels $\mathcal{L}=\{\lambda_1,\lambda_2,\ldots,\lambda_Q\}, Q>1$. For $Q>2$, the learning problem is referred to as *multi-class classification*. On the other hand, the task of learning a mapping from an example $\mathbf{x}\in\mathcal{X}$ ($\mathcal{X}$ denotes the domain of examples) to a set of labels $\mathcal{Y}\subseteq\mathcal{L}$ is referred to as a *multi-label classification*. In contrast to multi-class classification, alternatives in multi-label classification are not assumed to be mutually exclusive: multiple labels may be associated with a single example, i.e., each example can be a member of more than one class. Labels in the set $\mathcal{Y}$ are called relevant, while the labels in the set $\mathcal{L}\backslash\mathcal{Y}$ are irrelevant for a given example.

Besides the concept of multi-label classification, multi-label learning introduces the concept of *multi-label ranking* [1]. Multi-label ranking can be considered as a generalization of multi-class classification, where instead of predicting only a single label (the top label), it predicts the ranking of all labels. In other words, multi-label ranking is understood as learning a model that associates a query example $\mathbf{x}$ both with a ranking of the complete label set and a bipartition of this set into relevant and irrelevant labels.

The issue of learning from multi-label data has recently attracted significant attention from many researchers, motivated by an increasing number of new applications. The latter include semantic annotation of images and video (news clips, movies clips), functional genomics (gene and protein function), music categorization into emotions, text classification (news articles, web pages, patents, e-mails, bookmarks,…), directed marketing and others. In the last few years, several workshops have been organized and journal special issues edited covering the topic of multi-label learning.

In recent years, many different approaches have been developed to solving multi-label learning problems. Tsoumakas and Katakis [2] summarize them into two main categories: (a) algorithm adaptation methods and (b) problem transformation methods. Algorithm adaptation methods extend specific learning algorithms to handle multi-label data directly. Examples include lazy learning [3–5], neural networks [6,7], boosting [8,9], classification rules [10], decision trees [11,12], etc. Problem transformation methods, on the other hand, transform the

* Corresponding author at: Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University, Rugjer Boshkovikj 16, 1000 Skopje, Macedonia. Tel.: +389 76 359 067.
E-mail addresses: gjorgji.madjarov@finki.ukim.mk (Gj. Madjarov), Dragi.Kocev@ijs.si (D. Kocev), dejan.gjorgjevikj@finki.ukim.mk (D. Gjorgjevikj), Saso.Dzeroski@ijs.si (S. Džeroski).

0031-3203/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2012.03.004

multi-label learning problem into one or more single-label classification problems. The single-label classification problems are solved with a commonly used single-label classification approach and the output is transformed back into a multi-label representation. A common approach to problem transformation is to use class binarization methods, i.e., decompose the problem into several binary sub-problems that can then be solved by using a binary base classifier. The simplest strategies in the multi-label setting are the one-against-all and one-against-one strategies, also referred to as the binary relevance method [2] and pair-wise method [13,14], respectively.

In this study, we extend this categorization of multi-label methods with a third group of methods, namely, ensemble methods. This group of methods consists of methods that use ensembles to make multi-label predictions and their base classifiers belong to either problem transformation or algorithm adaptation methods. Methods that belong to this group are RAkEL [15], ensembles of classifier chains (ECC) [16], random forests of predictive clustering trees [17,18] and random forests of multi-label C4.5 trees [11].

As new methods for multi-label learning are proposed, they are experimentally compared to existing methods. The typical experimental evaluation compares the proposed method to a few existing ones on a few datasets. The methods are compared on performance in terms of one or a few error metrics and the comparison typically shows that the proposed method outperforms the other methods on some of the considered datasets and metrics. It is worth noting that a significant number of metrics has also been proposed for evaluating the performance of multi-label methods, which can concern the classification or ranking variant of the problem.

The number of proposed methods, datasets and metrics for multi-label learning constantly increases. As the research area of multi-label learning matures, there is a strong need for a comprehensive overview of methods and metrics. The need for a wider, extensive, and un-biased experimental comparison of multi-label learning methods is even stronger. It is this need that we address in the present paper.

In this study, we experimentally evaluate 12 methods for multi-label learning using 16 evaluation measures over 11 benchmark datasets. The multi-label methods comprise three algorithm adaptation methods, five problem transformation methods and four ensemble methods. The benchmark datasets are from five application domains: two from image classification, one from gene function prediction, six from text classification, one from music classification and one from video classification. The predictive performance of the methods is assessed using six example-based measures, six label-based measures and four ranking-based measures. Furthermore, we assess the efficiency of the methods by measuring their training and testing times. The large number of methods, datasets and evaluation measures are enabling us to draw some more general conclusions and to perform an un-biased assessment of the predictive performance of the multi-label methods.

The results from our extensive experimental evaluation will facilitate further research on multi-label learning as follows. First, this study will provide the research community with a better insight about the predictive performance of the methods currently available in the literature. Second, this study will identify a few methods that should be further used by the research community as benchmarks to compete against when proposing new methods. Third, this study uses a diverse collection of publicly available datasets that can be reused by other researchers as benchmark datasets for multi-label learning. Finally, this study will highlight the advantages of certain methods for certain types of datasets.

The remainder of this paper is organized as follows. Section 2 defines the tasks of multi-label classification and label ranking and surveys the related work. The state-of-the-art methods for multi-label learning used in the experimental evaluation are presented in Section 3. Section 4 describes the multi-label problems, the evaluation measures and the experimental setup, while Section 5 presents and discusses the experimental results. Finally, the conclusions are given in Section 6.

2. Background

In this section, we present the task of multi-label learning and methods for solving it. We begin by a formal definition of the task of multi-label learning. We then present an overview of the methods for multi-label learning.

2.1. The task of multi-label learning

Multi-label learning is concerned with learning from examples, where each example is associated with multiple labels. These multiple labels belong to a predefined set of labels. Depending on the goal, we can distinguish two types of tasks: multi-label classification and multi-label ranking. In the case of multi-label classification, the goal is to construct a predictive model that will provide a list of relevant labels for a given, previously unseen example. On the other hand, the goal in the task of multi-label ranking is to construct a predictive model that will provide, for each unseen example, a list of preferences (i.e., a ranking) of the labels from the set of possible labels.

We define the task of multi-label learning as follows:

Given:

- an example space $\mathcal{X}$ that consists of tuples of values of primitive data types (boolean, discrete or continuous), i.e., $\forall \mathbf{x_i} \in \mathcal{X}, \mathbf{x_i} = (x_{i_1}, x_{i_2}, \ldots, x_{i_D})$, where D is the size of the tuple (or number of descriptive attributes);
- a label space $\mathcal{L} = \{\lambda_1, \lambda_2, \ldots, \lambda_Q\}$ which is a tuple of Q discrete variables (with values 0 or 1);
- a set of examples E, where each example is a pair of tuples from the example and label space, respectively, i.e., $E = \{(\mathbf{x_i}, \mathcal{Y}_i) | \mathbf{x_i} \in \mathcal{X}, \mathcal{Y}_i \in \mathcal{L}, 1 \le i \le N\}$ and N is the number of examples of E ($N = |E|$); and
- a quality criterion q, which rewards models with high predictive accuracy and low complexity.

If the task at hand is multi-label classification, then the goal is to

Find: a function h: $\mathcal{X} \to 2^{\mathcal{L}}$ such that h maximizes q.

On the other hand, if the task is multi-label ranking, then the goal is to

Find: a function f: $\mathcal{X} \times \mathcal{L} \to \mathcal{R}$, such that f maximizes q, where $\mathcal{R}$ is the ranking of the labels for a given example.

2.2. An overview of methods for multi-label learning

Tsoumakas and Katakis [2] have presented the first overview of methods for multi-label learning where the methods for multi-label learning are divided into two categories: algorithm adaptation and problem transformation methods. There, three problem transformation methods were evaluated on a small empirical study (three datasets). In this study, we perform an extensive experimental evaluation of 12 methods for multi-label learning over 11 benchmark multi-label datasets using 16 evaluation measures. Furthermore, besides the two categories of methods for multi-label learning, we introduce a third category: ensemble methods. In the

remainder of this section, we present the three categories of methods for multi-label learning: algorithm adaptation, problem transformation and ensemble methods.

2.2.1. Algorithm adaptation methods

The multi-label methods that adapt, extend and customize an existing machine learning algorithm for the task of multi-label learning are called algorithm adaptation methods. Here, we present multi-label methods proposed in the literature that are based on the following machine learning algorithms: boosting, *k*-nearest neighbors, decision trees and neural networks. The extended methods are able to directly handle multi-label data.

Boosting: AdaBoost.MH and AdaBoost.MR [8] 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 ranks the correct labels at the top. Furthermore, AdaBoost.MH can also be combined with an algorithm for producing alternating decision trees [9]. The resulting multi-label models of this combination can be interpreted by humans.

k-Nearest neighbors: Several variants for multi-label learning (ML-*k*NN) of the popular *k*-Nearest Neighbors (*k*NN) lazy learning algorithm have been proposed [3–5]. The retrieval of the *k*-nearest neighbors is the same as in the traditional *k* NN algorithm. The main difference is the determination of the label set of a test example. Typically, these algorithms use prior and posterior probabilities of each label within the *k*-nearest neighbors. Cheng et al. [19] have proposed a hybrid method that uses logistic regression and *k*-nearest neighbors.

Decision trees: Clare et al. [11] adapted the C4.5 algorithm for multi-label data (ML-C4.5) by modifying the formula for calculating entropy. Blockeel et al. [12] proposed the concept of predictive clustering trees (PCTs). PCTs have been used for predicting tuples of variables, predicting time series and predicting classes organized into a hierarchy or a directed acyclic graph. However, they can also be used in the context of multi-label learning, where each label is a component of the target tuple.

Neural networks: Neural networks have also been adapted for multi-label classification [6,7]. BP-MLL [7] 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.

Support vector machines: Elisseeff and Weston [20] have proposed a ranking approach for multi-label learning that is based on SVMs. The cost function they use is the average fraction of incorrectly ordered pairs of labels.

2.2.2. Problem transformation methods

The problem transformation methods are multi-label learning methods that transform the multi-label learning problem into one or more single-label classification or regression problems. For smaller single-label problems, there exists a plethora of machine learning algorithms. Problem transformation methods can be grouped into three categories: binary relevance, label power-set and pair-wise methods.

Binary relevance methods: The simplest strategy for problem transformation is to use the one-against-all strategy to convert the multi-label problem into several binary classification problems. This approach is known as the binary relevance method (BR) [2]. A method closely related to the BR method is the Classifier Chain method (CC) proposed by Read et al. [16]. This method involves *Q* binary classifiers linked along a chain. Godbole et al. [21] present algorithms which extend the SVM binary classifiers along two dimensions: training set extension and improvement of margin. With the first approach, the training set is extended with the predictions of the binary classifiers and then a new set of binary classifiers is trained on the extended dataset. For the second extension, Godbole et al. remove very similar negative training examples and remove the negative training examples of a complete class that are similar to the positive class.

Label power-set methods: A second problem transformation method is the label combination method, or label power-set method (LP), which has been the focus of several recent studies [15,22,2]. The basis of these methods is to combine entire label sets into atomic (single) labels to form a single-label problem (i.e., single-class classification problem). For the single-label problem, the set of possible single labels represents all distinct label subsets from the original multi-label representation. In this way, LP based methods directly take into account the label correlations. However, the space of possible label subsets can be very large. To resolve this issue, Read [23] has developed a pruned problem transformation (PPT) method, that selects only the transformed labels that occur more than a predefined number of times. Another label power-set method is HOMER [24], which first constructs a hierarchy of the multiple labels and then constructs a classifier for the label sets in each node of the hierarchy.

Pair-wise methods: A third problem transformation approach to solving the multi-label learning problem is pair-wise or round robin classification with binary classifiers [13,14]. The basic idea here is to use $Q \cdot (Q-1)/2$ classifiers covering all pairs of labels. Each classifier is trained using the samples of the first label as positive examples and the samples of the second label as negative examples. To combine these classifiers, the pairwise classification method naturally adopts the majority voting algorithm. Given a test example, each classifier predicts (i.e., votes for) one of the two labels. After the evaluation of all $Q \cdot (Q-1)/2$ classifiers, the labels are ordered according to their sum of votes. A label ranking algorithm is then used to predict the relevant labels for each example. Besides majority voting in CLR, Park et al. [25] propose a more effective voting algorithm. It computes the class with the highest accumulated voting mass, while avoiding the evaluation of all possible pairwise classifiers. Mencia et al. [26] adapted the QWeighted approach to multi-label learning (QWML).

2.2.3. Ensemble methods

The ensemble methods for multi-label learning are developed on top of the common problem transformation or algorithm adaptation methods. The most well known problem transformation ensembles are the RAKEL system by Tsoumakas et al. [15], ensembles of pruned sets (EPS) [27] and ensembles of classifier chains (ECC) [16].

RAKEL constructs each base classifier by considering a small random subset of labels and learning a single-label classifier for the prediction of each element in the power-set of this subset. EPS uses pruning to reduce the computational complexity of label power-set methods, and an example duplication method to reduce the error rate as compared to label power-set and other methods. This method proved to be particularly competitive in terms of efficiency.

ECC are ensemble methods that have classifier chains (CC) as base classifiers. The final prediction is obtained by summing the predictions by label and then applying threshold for selecting the relevant labels. Note that binary methods are occasionally referred to as ensemble methods because they involve multiple binary models. However, none of these models is multi-label itself and therefore we use the term ensemble strictly in the sense of an ensemble of multi-label methods.

Algorithm adaptation ensemble methods are the ensembles whose base classifiers are themselves algorithm adaptation methods.

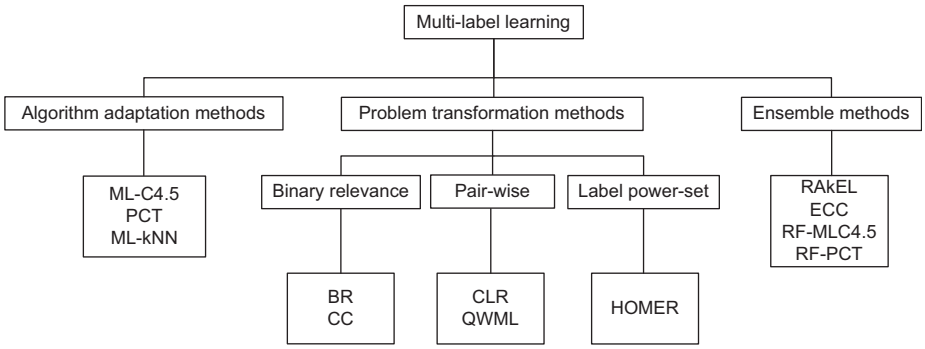

Fig. 1. The multi-label learning methods used in this study divided into groups as discussed in the related work section above.

An example of an algorithm adaptation ensemble method are the ensembles of predictive clustering trees (PCTs) [18]. These ensembles use PCTs for predicting tuples of variables as base classifiers. Each base classifier makes a multi-label prediction and then these predictions are combined by using some voting scheme (e.g., majority or probability distribution voting).

3. Methods for multi-label learning

In this section, we briefly introduce the state-of-the-art methods for multi-label learning that are used in this study. Fig. 1 depicts how these methods are divided into groups using the categorization scheme from the related work section. In this study, we use one label power-set method, two binary relevance and two pair-wise transformation methods, two algorithm adaptation methods, and four ensemble methods. Moreover, one of the ensemble methods is label power-set based, while the other methods are algorithm adaptation based.

We also divide the used multi-label learning approaches based on the type of basic machine learning algorithm they use. The methods use three types of base algorithms: SVMs, decision trees and *k*-nearest neighbors. We show this categorization in Fig. 2.

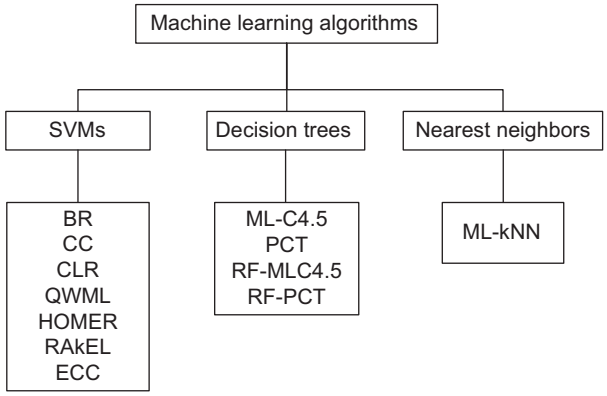

Fig. 2. The multi-label learning methods used in this study divided into groups based on the base machine learning algorithm they use.

3.1. Binary relevance methods

Binary relevance (*BR*) [2] is the well known one-against-all strategy. It addresses the multi-label learning problem by learning one classifier for each label, using all the examples labeled with that label as positive examples and all remaining examples as negative. When making a prediction, each binary classifier predicts whether its label is relevant for the given example or not, resulting in a set of relevant labels. In the ranking scenario, the labels are ordered according to the probability associated to each label by the respective binary classifier.

The *classifier chaining* (*CC*) *method* [16] involves Q binary classifiers as in BR. Classifiers are linked along a chain where the i-th classifier deals with the binary relevance problem associated with label $\lambda_i \in L$, $(1 \leq i \leq Q)$. The feature space of each link in the chain is extended with the 0/1 label associations of all previous links. The ranking and the prediction of the relevant labels in the CC method are the same as in the BR method.

3.2. Pair-wise methods

Calibrated label ranking (*CLR*) [25] is a technique for extending the common pair-wise approach to multi-label learning. It introduces an artificial (calibration) label λ_0, which represents the split-point between relevant and irrelevant labels. The calibration label λ_0 is assumed to be preferred over all irrelevant labels, but all relevant labels are preferred over it. It is represented by the binary relevance classifiers that are introduced as pair-wise classifiers in the context of pair-wise learning. At prediction time (majority voting is usually used), one will get a ranking over $Q+1$ labels (the Q original labels plus the calibration label λ_0). CLR is considered a combination of multi-label classification and ranking.

The *Quick Weighted voting method* for multi-class classification, proposed by Park et al. [25], is a variant of the CLR method that introduces a more effective voting strategy than the majority voting used by the CLR method. Quick weighted voting exploits the fact that during voting some classes can be excluded from the set of possible top rank classes early in the process, when it becomes clear that even if they reach the maximal voting mass in the remaining evaluations they can not exceed the current maximum. Pairwise classifiers are selected depending on a voting loss value, which is the number of votes that a class has not received. The voting loss starts with a value of zero and increases monotonically with the number of performed preference evaluations. The class with the current minimal loss is the best candidate for the top ranked class. If all preferences involving this class have been evaluated (and it still has the lowest loss), it can be concluded that no other class can achieve a better ranking. Thus, the quick weighted algorithm always focuses on classes with low voting loss. The adaptation of quick weighted algorithm for multi-label learning (QWML) [26] is done by repeating the process while all relevant labels are not determined, i.e., until the returned label is the artificial label, which means that all remaining labels will be considered irrelevant.

3.3. Label power-set method

Hierarchy Of Multi-label classifiERs (*HOMER*) [24] is an algorithm for effective and computationally efficient multi-label learning in domains with a large number of labels. HOMER constructs a

hierarchy of multi-label classifiers, each one dealing with a much smaller set of labels compared to Q (the total number of labels) and a more balanced example distribution. This leads to improved predictive performance and also to linear training and logarithmic testing complexities with respect to Q. One of the main processes within HOMER is the even distribution of a set of labels into k disjoint subsets so that similar labels are placed together and dissimilar apart. The best predictive performance is reported using a balanced k means algorithm customized for HOMER [24]. HOMER is a computationally efficient multi-label classification method, specifically designed for large multi-label datasets.

3.4. Algorithm adaptation methods

Multi-Label C4.5 (*ML-C4.5*) [11] is an adaptation of the well known C4.5 algorithm for multi-label learning by allowing multiple labels in the leaves of the tree. Clare et al. [11] modified the formula for calculating entropy (see Eq. (1)) for solving multi-label problems. The modified entropy sums the entropies for each individual class label. The key property of ML-C4.5 is its computational efficiency:

$$entropy(E) = -\sum_{i=1}^{N}(p(c_i)\log p(c_i)+q(c_i)\log q(c_i)) \quad (1)$$

where E is the set of examples, $p(c_i)$ is the relative frequency of class label c_i and $q(c_i)=1-p(c_i)$.

Predictive clustering trees (*PCTs*) [12] are decision trees viewed as a hierarchy of clusters: the top-node corresponds to one cluster containing all data, which is recursively partitioned into smaller clusters while moving down the tree. PCTs are constructed using a standard top-down induction of decision trees algorithm, where the variance and the prototype function can be instantiated according to the task at hand. Namely, PCTs can handle several types of structured outputs: tuples of continuous or discrete variables, time series, classes organized into a hierarchy, tuples of time series and tuples of hierarchies [18]. For the task of predicting tuples of discrete variables, the variance function is computed as the sum of the *Gini indices* [28] of the variables from the target tuple, i.e., $Var(E)=\sum_{i=1}^{T} Gini(E,Y_i)$, $Gini(E,Y_i)=1-\sum_{j=1}^{C_i} p_{c_{ij}}$, where T is the number of target attributes, c_{ij} is the j-th class of target attribute Y_i and C_i is the number of classes of target attribute Y_i. The prototype function returns a vector of probabilities that an example belongs to a given class for each variable from the target tuple. In the case of multi-label learning, it returns a vector of probabilities that an example is labeled with a given label.

Multi-label k-nearest neighbors (*ML-kNN*) [3] is an extension of the popular k-nearest neighbors (kNN) algorithm. First, for each test example, its k-nearest neighbors in the training set are identified. Then, according to statistical information gained from the label sets of these neighboring examples, i.e., the number of neighboring examples belonging to each possible label, the maximum *a posteriori* principle is used to determine the label set for the test example.

3.5. Ensemble methods

The *RAndom k-labELsets* (*RAkEL*) [15] is an ensemble method for multi-label classification. It draws m random subsets of labels with size k from all labels $\mathcal{L}$ and trains a label power-set classifier using each set of labels. A simple voting process determines the final set of labels for a given example. In this way, the proposed algorithm aims to take into account label correlations using single-label classifiers that are applied on subtasks with a manageable number of labels and adequate number of examples per label.

Ensembles of classifier chains (*ECC*) [16] are an ensemble multi-label classification technique that uses classifier chains as a base classifier. ECC trains m CC classifiers $C_1,C_2,\ldots,C_m$. Each C_k is trained with a random chain ordering (of $\mathcal{L}$) and a random subset of $\mathcal{X}$. Hence each C_k model is likely to be unique and able to give different multi-label predictions. These predictions are summed per label so that each label receives a number of votes. A threshold is used to select the most popular labels which form the final predicted multi-label set.

Random forest of predictive clustering trees (*RF-PCT*) [17,18] and *Random forest of ML-C4.5* (*RFML-C4.5*)[1] are ensembles that use PCTs and ML-C4.5 trees, respectively, as base classifiers. The diversity among the base classifiers is obtained by using bagging, and additionally by changing the feature set during learning [29]. More precisely, at each node in the decision trees, a random subset of the input attributes is taken, and the best feature is selected from this subset. The number of attributes that are retained is given by a function f of the total number of input attributes x (e.g., $f(x)=1, f(x)=\sqrt{x}, f(x)=\lfloor 0.1\cdot x+1\rfloor, f(x)=\lfloor \log_2(x)+1\rfloor \ldots$). The predictions of the base classifiers are then combined using some voting scheme (typically, majority or probability distribution vote).

4. Experimental design

In this section, we present the experimental design used to compare the methods for multi-label learning. We first shortly describe the benchmark multi-label datasets. We then give a short overview of the evaluation measures typically applied to asses the predictive performance of methods for multi-label learning. Next, we present the specific setup and the instantiation of the parameters for the used methods for multi-label learning. Finally, we present the procedure for statistical evaluation of the experimental results.

4.1. Datasets

We use 11 different multi-label classification benchmark problems. Parts of the selected problems were used in various studies and evaluations of methods for multi-label learning. In the process of selection of problems, we opted to include benchmark datasets with different scale and from different application domains. Table 1 presents the basic statistics of the datasets. We can note that the datasets vary in size: from 391 up to 60 000 training examples, from 202 up to 27 856 testing examples, from 72 up to 2150 features, from 6 to 983 labels, and from 1.07 to 19.02 average number of labels per example (i.e., label cardinality [30]). From the literature, these datasets come pre-divided into training and testing parts: thus, in the experiments, we use them in their original format. The training part usually comprises around 2/3 of the complete dataset, while the testing part the remaining 1/3 of the dataset.

The datasets come from three domains: biology, multimedia and text categorization. From the biological domain, we have the *yeast* dataset [20]. It is a widely used dataset, where genes are instances in the dataset and each gene can be associated with 14 biological functions (labels).

The datasets that belong to the multimedia domain are: emotions, scene, corel5k and mediamill. *Emotions* [31] is a dataset

[1] We have implemented the random forest of ML-C4.5 trees within the MULAN library for multi-label learning.

Table 1
Description of the benchmark problems in terms of application domain (*domain*), number of training (*#tr.e.*) and test (*#t.e.*) examples, the number of features (D), the total number of labels (Q) and label cardinality (l_c). The problems are ordered by their overall complexity roughly calculated as $\#tr.e. \times D \times Q$.

Dataset	*Domain*	*#tr.e.*	*#t.e.*	D	Q	l_c
emotions [31]	Multimedia	391	202	72	6	1.87
scene [32]	Multimedia	1211	1159	294	6	1.07
yeast [20]	Biology	1500	917	103	14	4.24
medical [16]	Text	645	333	1449	45	1.25
enron [33]	Text	1123	579	1001	53	3.38
corel5k [34]	Multimedia	4500	500	499	374	3.52
tmc2007 [35]	Text	21 519	7077	500	22	2.16
mediamill [36]	Multimedia	30 993	12 914	120	101	4.38
bibtex [37]	Text	4880	2515	1836	159	2.40
delicious [24]	Text	12 920	3185	500	983	19.02
bookmarks [37]	Text	60 000	27 856	2150	208	2.03

where each instance is a piece of music. Each piece of music can be labeled with six emotions: sad-lonely, angry-aggressive, amazed-surprised, relaxing-calm, quiet-still, and happy-pleased. *Scene* [32] is a widely used scene classification dataset. Each scene can be annotated in the following six contexts: beach, sunset, field, fall-foliage, mountain, and urban. The *Corel5k* [34] data set contains Corel images that are segmented using normalized cuts. The segmented regions are then clustered into 499 bins, which are further used to describe the images. Each image can be then assigned several of the 374 possible labels. *Mediamill* [36] originates from the 2005 NIST TRECVID challenge dataset,[2] which contains data about annotated videos. The label space is represented by 101 "annotation concepts", such as explosion, aircraft, face, truck, urban, etc.

The domain of text categorization is represented with six datasets: medical, enron, tmc2007, bibtex, delicious and bookmarks. *Medical* [16] is a dataset used in the Medical Natural Language Processing Challenge[3] in 2007. Each instance is a document that contains brief free-text summary of a patient symptom history. The goal is to annotate each document with the probable diseases from the International Classification of Diseases (ICD-9-CM) [38]. *Enron* [33] is a dataset that contains the e-mails from 150 senior Enron officials. The e-mails were categorized into several categories developed by the UCBerkeley Enron Email Analysis Project.[4] The labels can be further grouped into four categories: coarse genre, included/forwarded information, primary topics, and messages with emotional tone. *Tmc2007* [35] contains instances of aviation safety reports that document problems that occurred during certain flights. The labels represent the problems being described by these reports. We use a reduced version of this dataset with the top 500 attributes selected, same as Tsoumakas et al. [15]. *Delicious*, *bibtex* and *bookmarks* are used for automatic tag suggestion. Delicious [24] contains web pages and their tags. The web pages are taken from the del.ico.us social bookmarking site.[5] Note that the label space is greater than the size of the input space ($Q > D$) for this dataset. Bibtex [37] contains metadata for bibtex items, such as the title of the paper, the authors, book title, journal volume, publisher, etc., while bookmarks [37] contains metadata for bookmark items, such as the URL of the web page, an URL hash, a description of the web page, etc.

[2] http://www.science.uva.nl/research/mediamill/challenge/
[3] http://www.computationalmedicine.org/challenge/
[4] http://bailando.sims.berkeley.edu/enron_email.html
[5] http://delicious.com/

4.2. Evaluation measures

Performance evaluation for multi-label learning systems differs from that of classical single-label learning systems. In any multi-label experiment, it is essential to include multiple and contrasting measures because of the additional degrees of freedom that the multi-label setting introduces. In our experiments, we used various evaluation measures that have been suggested by Tsoumakas et al. [30]. Fig. 3 depicts a categorization of the used evaluation measures. Furthermore, we evaluate the algorithms by their efficiency. Namely, we measure the time needed to construct the predictive models (*training time*) and the time needed to obtain a prediction for an unseen example (*testing time*).

The evaluation measures of predictive performance are divided into two groups: *bipartitions-based* and *rankings-based*. The bipartitions-based evaluation measures are calculated based on the comparison of the predicted relevant labels with the ground truth relevant labels. This group of evaluation measures is further divided into *example-based* and *label-based*. The example-based evaluation measures are based on the average differences of the actual and the predicted sets of labels over all examples of the evaluation dataset. The label-based evaluation measures, on the other hand, assess the predictive performance for each label separately and then average the performance over all labels. In our experiments, we used six example-based evaluation measures (*Hamming loss*, *accuracy*, *precision*, *recall*, F_1 *score* and *subset accuracy*) and six label-based evaluation measures (*micro-precision*, *micro-recall*, *micro-*F_1, *macro-precision*, *macro-recall* and *macro-*F_1). Note that these evaluation measures require predictions stating that a given label is present or not (binary 1/0 predictions). However, most predictive models predict a numerical value for each label and the label is predicted as present if that numerical value exceeds some predefined threshold τ. The performance of the predictive model thus directly depends on the selection of an appropriate value of τ. To this end, we applied a threshold calibration method by choosing the threshold that minimizes the difference in label cardinality between the training data and the predictions for the test data [16].

The ranking-based evaluation measures compare the predicted ranking of the labels with the ground truth ranking. We used four ranking-based measures: *one-error*, *coverage*, *ranking loss* and *average precision*. A detailed description of the evaluation measures is given in Appendix A.

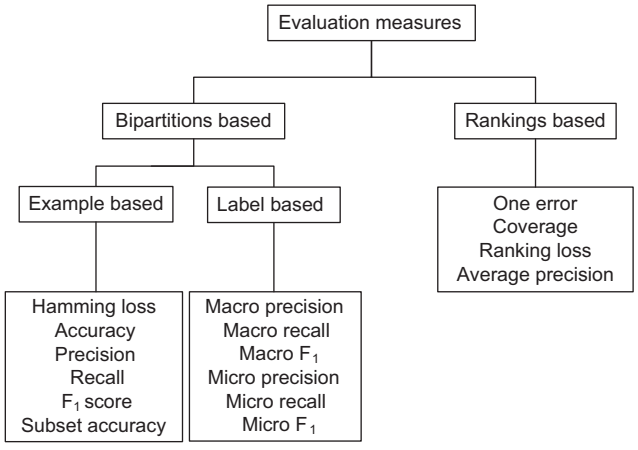

Fig. 3. Categorization of evaluation measures used to assess the predictive performance of methods for multi-label learning.

4.3. Experimental setup

The comparison of the multi-label learning methods was performed using the implementations in the following machine learning systems: MULAN[6] library under the machine learning framework WEKA [39], MEKA[7] extension for the WEKA framework and CLUS[8] system for predictive clustering. The MULAN library was used for BR, CLR, QWML, HOMER, ML-C4.5, RFML-C4.5, ML-*k* NN and RAkEL; the MEKA environment was used for CC and ECC and the CLUS system for PCT and RF-PCT. All experiments were performed on a server with an Intel Xeon processor at 2.50 GHz on 64 GB of RAM with the Fedora 14 operating system. In the remainder of this section, we first state the base classifiers that were used for the multi-label methods and then the parameter instantiations of the methods.

4.3.1. Base classifiers

The methods used in this study use two types of base classifiers for solving the partial binary classification problems in all problem transformation methods and the ensemble methods: SVMs and decision trees (see Fig. 2). For training the SVMs, we used the implementation from the LIBSVM library [40]. In particular, we used SVMs with a radial basis kernel for all problem transformation methods and RAkEL and ECC. The kernel parameter *gamma* and the penalty *C*, for each combination of dataset and method, were determined by 10-fold cross validation using only the training sets. The exception to this is the ensemble method RAkEL where the kernel parameter *gamma* and the penalty *C* were determined by 5-fold cross validation for the *tmc2007* and *mediamill* datasets because of its computational complexity. The values $2^{-15},2^{-13},\ldots,2^{1},2^{3}$ were considered for *gamma* and $2^{-5},2^{-3},\ldots,2^{13},2^{15}$ for the penalty *C*. After determining the best parameters values for each method on every dataset, the classifiers were trained using all available training examples and were evaluated by recognizing all test examples from the corresponding dataset.

We used two implementations of decision trees: ML-C4.5 from MULAN and PCTs from CLUS. The ML-C4.5 and PCT as predictive models were pruned using a pruning method. ML-C4.5 uses a post-pruning strategy based on a confidence factor, while PCTs use a pre-pruning strategy based on the F-test (whether a given split significantly reduces the variance). On the other hand, when they were used as base classifiers in the ensembles (RFML-C4.5 and RF-PCT), the trees were fully grown [41].

4.3.2. Parameter instantiation

The parameters of the methods were instantiated following the recommendations from the literature. In particular, for the ensemble methods based on decision trees (RFML-C4.5 and RF-PCT), the number of models (classifiers) used in the ensemble was 100 as suggested by Bauer and Kohavi [41]. For the size of the feature subsets needed for construction of the base classifiers for RFML-C4.5, we selected the $f(x)=\lfloor \log_2(x)+1 \rfloor$ as recommended by Breiman [29], while for RF-PCT we selected $f(x)=\lfloor 0.1\cdot x+1 \rfloor$ as recommended by Kocev [18]. The number of models in the ECC method was set to 10 as proposed by Read et al. [16]. Next, the number of models in RAkEL was set to $\min(2\cdot Q,100)$ (Q is the number of labels) for all datasets [15], except for the *mediamill*, *delicious* and *bookmarks* datasets, where this parameter was set to 10 as a result of the memory requirements of this method. Besides the number of base classifiers, RAkEL requires one additional parameter: the size of the label-sets *k*. For each dataset, this parameter was set to half the number of labels ($Q/2$). Tsoumakas et al. [15] and Read et al. [16] have shown that this is a reasonable choice, since it provides a balance between computational complexity and predictive performance.

The ML-C4.5 method uses sub-tree raising as a post-pruning strategy with a pruning confidence set to 0.25. Furthermore, the minimal number of examples in the leaves in each model of the RFML-C4.5 was set to 10. PCTs use a pre-pruning strategy that employs the F-test to determine whether a given split results in a significant reduction of variance. The significance level for the F-test was automatically selected from a predefined list of significance levels using 3-fold cross validation. The number of neighbors in the ML-*k*NN method for each dataset was determined from the values 6 to 20 with step 2. HOMER also requires one additional parameter to be configured: the number of clusters. For this parameter, five different values (2–6) were considered in the experiments [24] and we report the best results.

4.4. Statistical evaluation

To assess whether the overall differences in performance across the ten different approaches are statistically significant, we employed the corrected Friedman test [42] and the post-hoc Nemenyi test [43] as recommended by Demšar [44]. The Friedman test is a non-parametric test for multiple hypotheses testing. It ranks the algorithms according to their performance for each dataset separately, thus the best performing algorithm gets the rank of 1, the second best the rank of 2, etc. In case of ties, it assigns average ranks. Then, the Friedman test compares the average ranks of the algorithms and calculates the Friedman statistic χ_F^2, distributed according to the χ_F^2 distribution with $k-1$ degrees of freedom (k being the number of algorithms). Iman and Davenport [45] have shown that the Friedman statistic is undesirably conservative and derive a corrected F-statistic that is distributed according to the F-distribution with $k-1$ and $(k-1)\cdot(N-1)$ degrees of freedom (N being the number of datasets).

If a statistically significant difference in the performance is detected, then next step is a post-hoc test to detect between which algorithms those differences appear. The Nemenyi test is used to compare all the classifiers to each other. In this procedure, the performance of two classifiers is significantly different if their average ranks differ by more than some critical distance. The critical distance depends on the number of algorithms, the number of datasets and the critical value (for a given significance level – p) that is based on the Studentized range statistic and can be found in statistical textbooks (e.g., see [46]).

We present the results from the Nemenyi post-hoc test with average rank diagrams [44]. These are given in Figs. 4–7 and B1–B4. A critical diagram contains an enumerated axis on which the average ranks of the algorithms are drawn. The algorithms are depicted along the axis in such a manner that the best ranking ones are at the right-most side of the diagram. The lines for the average ranks of the algorithms that do not differ significantly (at the significance level of $p=0.05$) are connected with a line.

For the larger datasets, several algorithms did not construct a predictive model within one week under the available resources.[9] These occurrences are marked as DNF (Did Not Finish) in the tables with the results. Considering this, we perform the statistical analysis twice. For the first analysis (Figs. 4–7), we use only the datasets for which all the methods finished and provided results (eight datasets). For the second analysis (Figs. B1–B4), we penalize the algorithms that do not finish by assigning them the

[6] http://mulan.sourceforge.net/
[7] http://meka.sourceforge.net/
[8] http://clus.sourceforge.net

[9] The experiments were performed on a server running Linux, with two Intel Quad-Core Processors running at 2.5 GHz and 64 GB of RAM.

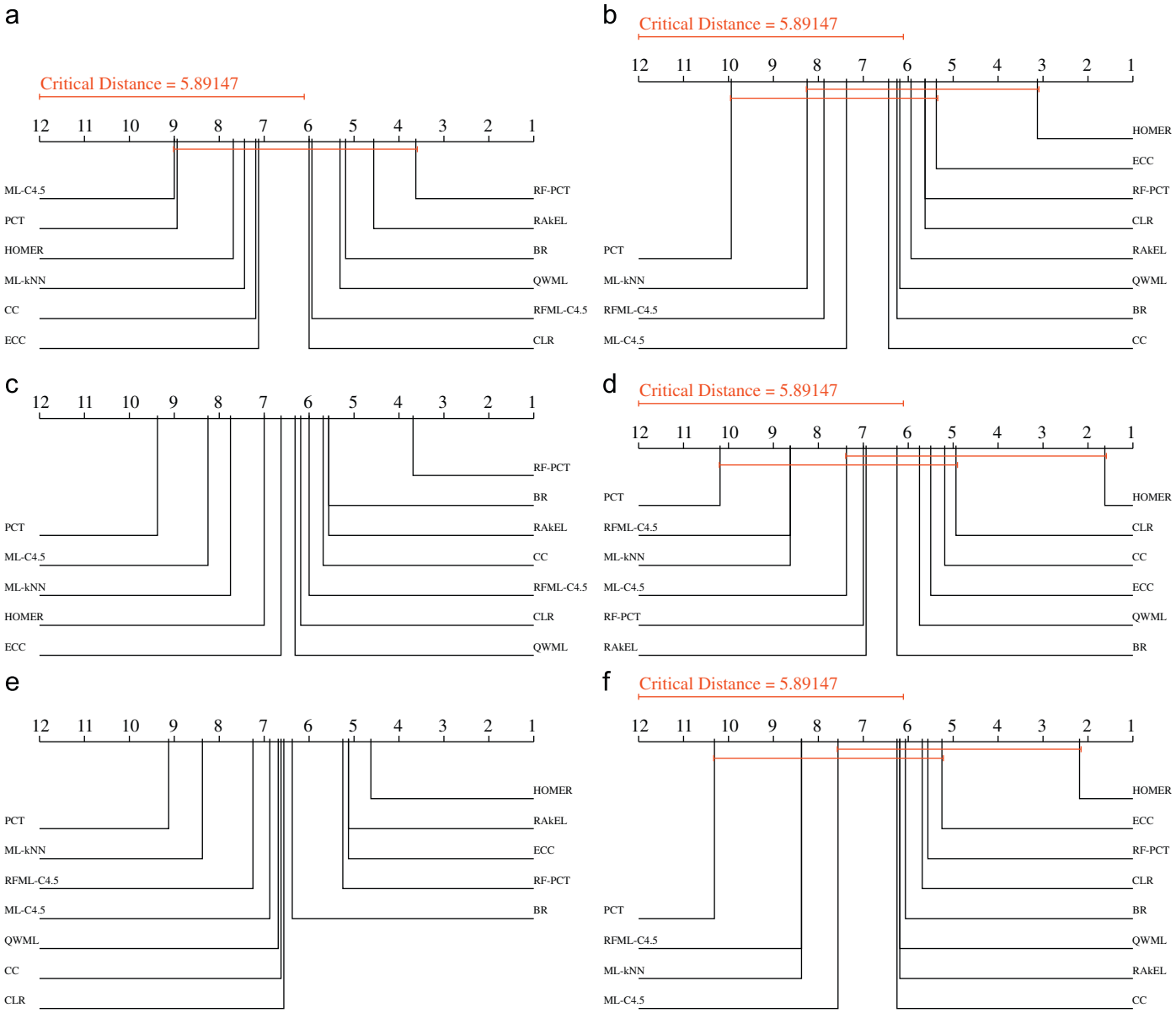

Fig. 4. The critical diagrams for the example-based evaluation measures: the results from the Nemenyi post-hoc test at 0.05 significance level on the datasets for which all algorithms provided results. For *precision* and *subset accuracy* the differences are not statistically significant according to the Friedman test (see Table B7), thus we show only the average ranks of the algorithms: (a) *Hamming loss*; (b) *accuracy*; (c) *precision*; (d) *recall*; (e) *subset accuracy*; and (f) F_1 *score*.

lowest value (i.e., the lowest rank value for the given algorithm-dataset pair) for each evaluation measure.

5. Results and discussion

In this section, we present the results from the experimental evaluation. For each type of evaluation measure, we present and discuss the critical diagrams from the tests for statistical significance using the datasets on which all algorithms provided predictive models. We give complete results over all evaluation measures and all critical diagrams (including those for all datasets) in Appendix B.

5.1. Results on the example-based measures

The example-based evaluation measures include *Hamming loss*, *accuracy*, *precision*, *recall*, F_1 *score* and *subset accuracy*. The results of the statistical evaluation are given in Fig. 4, while the complete results are given in Tables B1–B7, and Fig. B1. Considering the results that include the datasets for which all algorithms finished (Fig. 4), we can make several conclusions. The first conclusion that draws our attention is that HOMER performs best as evaluated by *recall*, while RF-PCT performs best according to *precision*. This means that the predictions made by HOMER are more complete: the original relevant labels were correctly predicted as relevant labels (small number of false negatives results in high *recall*). However, the lower *precision* means that besides the labels that were originally relevant, HOMER predicts non-relevant labels as relevant (larger number of false positives results in low *precision*). The situation is somewhat reversed when looking at the predictions from RF-PCT. The predictions of RF-PCT are more exact: the labels predicted as relevant were truly relevant in the original examples (small number of false positives results in high *precision*). However, RF-PCT is leaving out some of the relevant labels when making

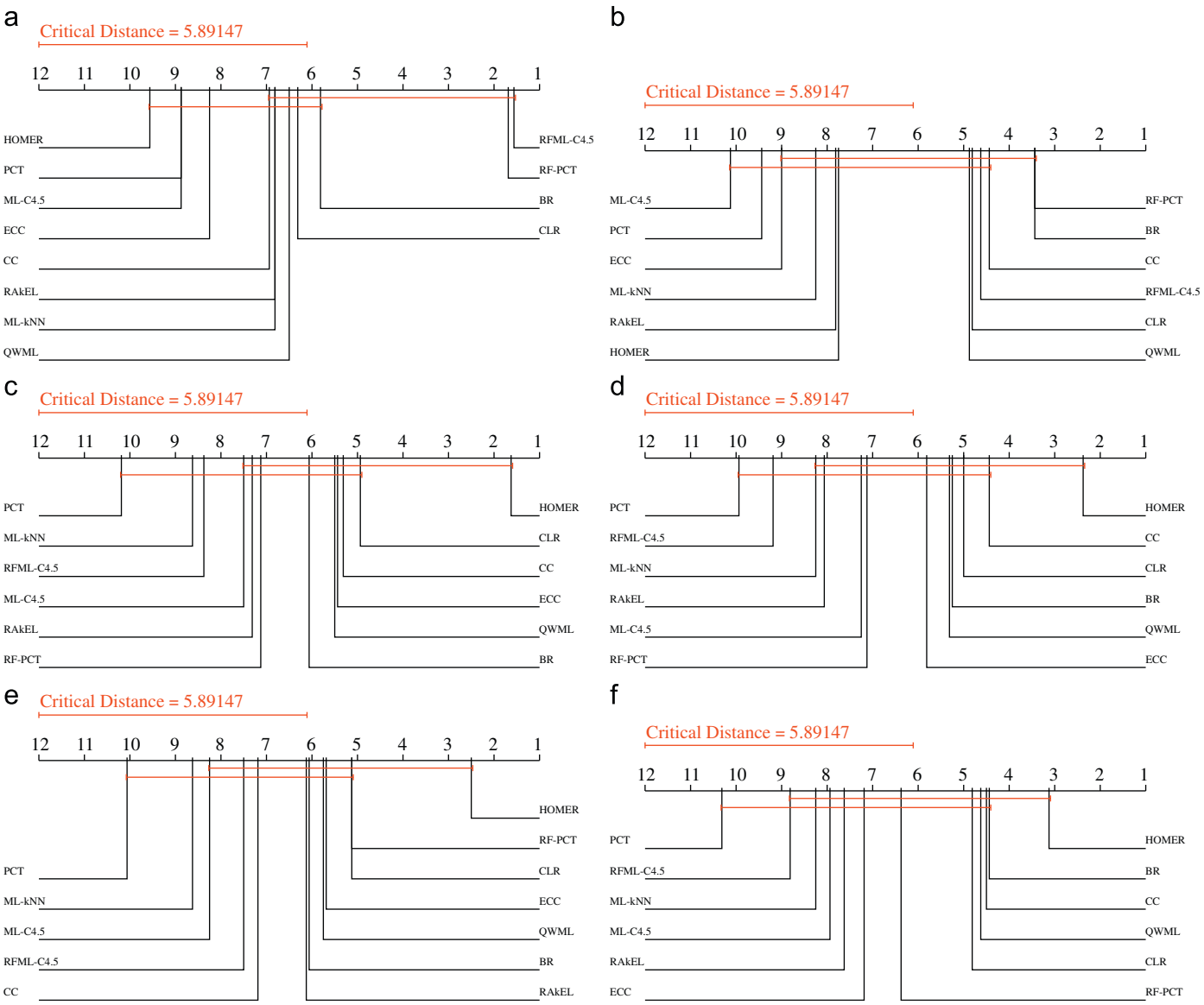

Fig. 5. The critical diagrams for the label-based evaluation measures: the results from the Nemenyi post-hoc test at 0.05 significance level on the datasets for which all algorithms provided results: (a) *micro-precision*; (b) *macro-precision*; (c) *micro-recall*; (d) *macro-recall*; (e) *micro-F_1*; and (f) *macro-F_1*.

predictions (larger number of false negatives results in high *recall*).

We further analyze the performance of the methods across all six evaluation measures: the best performing methods on all measures are either RF-PCT or HOMER. We can further note that RF-PCT is the best performing method, closely followed by HOMER, BR and CC. The RF-PCT method performs best according to *Hamming loss* and *precision*, third best according to *accuracy* and *F_1 score*. HOMER is the best performing as evaluated by *subset accuracy*, *accuracy*, *recall* and *F_1 score*. The HOMER method has poor performance as evaluated by *Hamming loss* and *precision* (10-th and 9-th position on the critical diagrams, respectively), while on these two measures RF-PCT performs the best. We hypothesize that the low performance of HOMER according to *Hamming loss* is because the procedure for construction of HOMER's hierarchical structure does not optimize *Hamming loss*.

The differences in predictive performance are rarely significant at the significance level of 0.05. HOMER and RF-PCT are often significantly better than single PCT, or single ML-C4.5 trees. From an ensemble learning point of view, this means that the RF-PCTs lift the predictive performance of a single PCT even when the target concept is a set of labels, similarly as for simple regression and classification. On the other hand, the increase in predictive performance is not constant for the ECC and RF-MLC4.5 ensembles: according to some evaluation measure, the single models perform even better on average than the corresponding ensembles. For this we have two hypotheses: first, CC are stable classifiers and ensemble can't much improve over their predictive performance. Second, RF-MLC4.5 did not perform competitively because it selects feature subsets with a logarithmic size compared to the complete set of features. Considering that the domains we used in this study (and other multi-label domains) have a large number of features (typically larger than 500), the logarithmic function under-samples the feature space and is missing some useful information that can contribute to better classification.

We next focus the discussion on the different types of base machine learning algorithms. First, we can note that the multi-label variant of *k*-nearest neighbors (ML-*k*NN) performs poor by across all evaluation measures. Next, the SVM-based methods perform better for the smaller datasets, while tree-based methods for the larger datasets. This is because the Gaussian kernel can

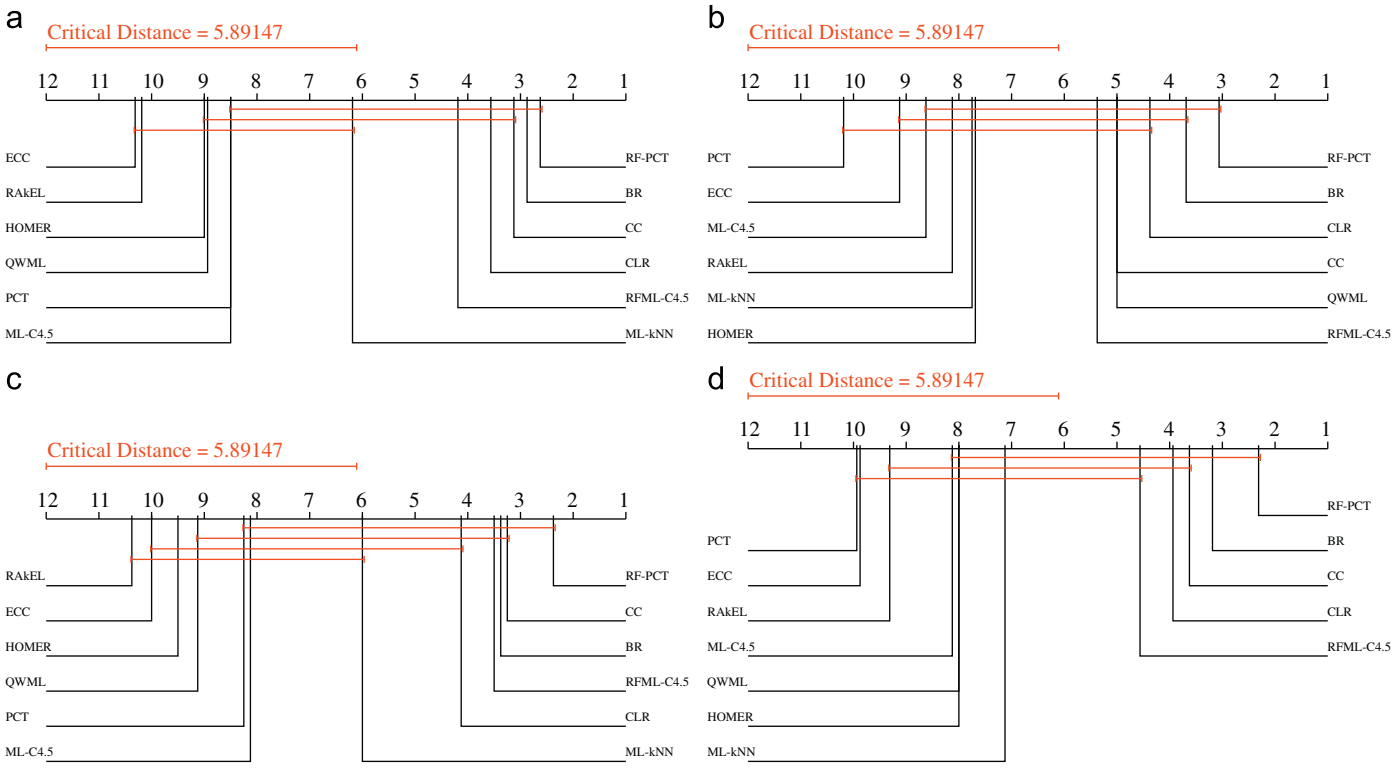

Fig. 6. The critical diagrams for the ranking-based evaluation measures: the results from the Nemenyi post-hoc test at 0.05 significance level on the datasets for which all algorithms provided results: (a) *ranking loss*; (b) *one-error*; (c) *coverage*; and (d) *average precision*.

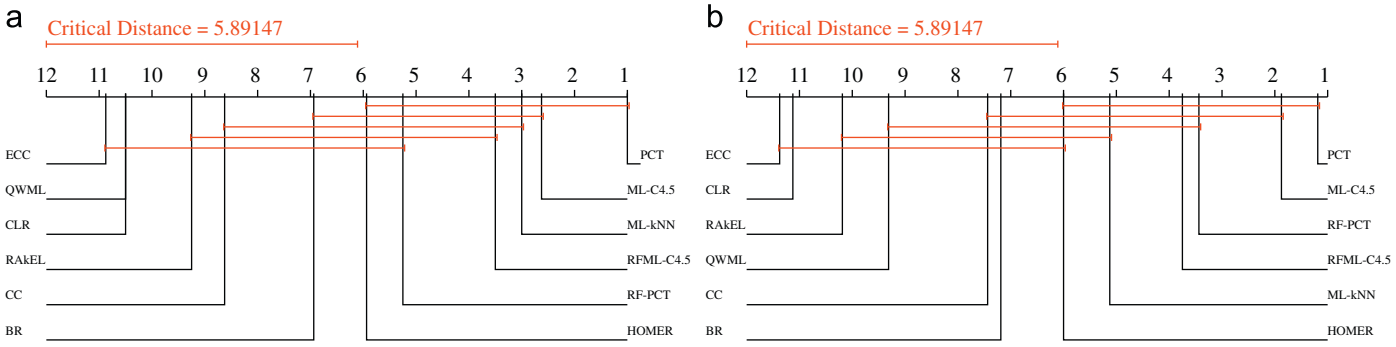

Fig. 7. The critical diagrams for the efficiency measures: the results from the Nemenyi post-hoc test at 0.05 significance level on the datasets for which all algorithms provided results: (a) *training time* and (b) *testing time*.

handle very well the smaller number of examples: when the number of examples increases, the performance of the kernel approaches the performance of a linear kernel. Furthermore, the SVM-based methods are better for the domains with larger numbers of features. For instance, in text classification an example is a document typically represented as a bag-of-words, where each feature can play a crucial role in making a correct prediction. The SVMs exploit the information from all the features, while the decision trees use only a (small) subset of features and may miss some crucial information.

Finally, we discuss the addition of all datasets in the statistical analysis (shown in Fig. B1). This analysis shows that RF-PCT and BR have improved predictive performance at the expense of the methods that did not finish. RF-PCT is again the best performing method overall, followed by BR, HOMER and CC. RF-PCT is best according to *Hamming loss*, *precision* and *subset accuracy*, while HOMER is best according to *accuracy*, F_1 *score* and *recall*. RF-PCT is second best on *accuracy* and F_1 *score* and has an improved performance according to *recall*.

5.2. Results on the label-based measures

The label-based evaluation measures include *micro-precision*, *micro-recall*, *micro-*F_1, *macro-precision*, *macro-recall* and *macro-*F_1. The results from the statistical evaluation are given in Fig. 5, while complete results are given in Tables B8–B14, and Fig. B2. First, we focus on the results and the statistical analysis on the datasets for which all methods have finished. As for the example-based measures, the best performing methods are RF-PCT, HOMER, BR and CC. HOMER performs best according to four evaluation measures: *macro-*F_1, *macro-recall*, *micro-*F_1 and *micro-recall* and it performs worst of all methods according to *micro-precision* and 7-th according to *macro-precision*. RF-PCT is the best performing using *macro-precision* and, according to the critical diagrams, performs statistically significantly better than HOMER for the *micro precision* evaluation measure.

We next discuss the performance of the ensembles and the single models. Again, as for the example-based measures, RF-PCT is better than single PCT over all evaluation measures. On the

other hand, this is not the case for RF-MLC4.5 and ECC. The reasons for this are the same as for the example-based measures: CC is a stable classifier and the logarithmic size of the feature subset for RF-MLC4.5 is under-sampling the feature space.

The behavior of the base machine learning algorithms remains the same as for the example-based measures. ML-k NN again has very poor predictive performance across all evaluation measures. SVMs are better for the smaller datasets and decision trees for the larger datasets.

Finally, the addition of the datasets for which some (but not all) of the methods finished did not change the results much (compare Fig. 5 with Fig. B2). The updated results improved the average performance of RF-PCT, BR and CC at the expense of the methods not able to produce results. However, the relative average performance remained the same as for the subset of datasets: HOMER is best according to *macro-F_1*, *macro-recall*, *micro-F_1* and *micro-recall*, while RF-PCT is best according to *micro-precision* and *macro-precision*. Moreover, RF-PCT is statistically significantly better than HOMER according to *micro-precision*.

5.3. Results on the ranking-based measures

The ranking-based measures include *one-error*, *ranking loss*, *coverage* and *average precision*. The results from the statistical evaluation are given in Fig. 6, while the complete results are given in Tables B15–B19 and Fig. B3. We first focus on the results for the datasets for which all methods have finished. The best performing method is RF-PCT, followed by BR, CC and CLR. RF-PCT is the best performing method on all four evaluation measures, BR is second best on three measures (*one-error*, *ranking loss* and *average precision*) and third on *coverage*. Considering ensemble learning, the ensembles based on decision trees perform better than the corresponding single models. However, the ensembles of CC perform worse than a single CC. The ranking-based measures indicate that the SVM-based methods perform better for smaller datasets, while tree-based measures perform better on larger datasets.

Let us further compare RF-PCT with HOMER using the ranking measures. The statistical evaluation of the performance reveals that RF-PCT is statistically significantly better than HOMER at the significance level of 0.05 according to *coverage* and *ranking loss* (see Fig. 6). Furthermore, the statistical evaluation using all datasets (Fig. B3) shows that RF-PCT is statistically significantly better than HOMER on all evaluation measures except *one–error*. The other results from the statistical analysis using all datasets are similar to the results on the subset of datasets.

5.4. Results on the efficiency measures

We finally discuss the efficiency of the proposed methods in terms of training and testing time. The results are given in Figs. 7 and B4, and Tables B20–B22. They show that the tree-based methods are more efficient than the SVM-based methods. Namely, PCT is the most efficient method, followed by ML-C4.5 and ML-k NN. PCTs are faster to construct than ML-C4.5 because of the pruning strategy they employ: the former used pre-pruning and the latter post-pruning.

We further discuss the methods that exhibited the best predictive performance according to the other evaluation measures: RF-PCT and HOMER. RF-PCT is better than HOMER on the time needed to produce a prediction for an unseen example (*testing time*) and on the time needed for learning a classifier (*training time*) for both analyses: the first includes only the datasets with complete results and the second includes all datasets. Moreover, HOMER did not produce results for the *bookmarks* dataset.

6. Conclusions

In this study, we present an extensive experimental evaluation of methods for multi-label learning. The topic of multi-label learning has lately received significant research effort. It has also attracted much attention from the research community, in the form of journal special issues and workshops at major conferences. This has resulted in a variety of methods for addressing the task of multi-label learning. However, a wider experimental comparison of these methods is still lacking in the literature.

We evaluate the most popular methods for multi-label learning using a wide range of evaluation measures on a variety of datasets. Below we explain the dimensions of the extensive experimental evaluation. First, we selected 12 multi-label methods that were recently proposed in the literature. The selected methods are divided into three main groups: algorithm adaptation (three methods), problem transformation (five methods) and ensembles (four methods). The methods use three types of basic machine learning algorithms: SVMs (seven methods), decision trees (four methods) and k-nearest neighbors (one method). Second, we used 16 different evaluation measures that are typically used in the context of multi-label learning. The variety of evaluation measures is necessary to provide a view on algorithm performance from different perspectives. The evaluation measures are divided in three groups: example-based (six measures), label-based (six measures) and ranking-based (four measures). Furthermore, we assess the efficiency of the methods by measuring the time needed to learn the classifier and the time needed to produce a prediction for an unseen example. Third, we evaluate the methods on 11 multi-label benchmark datasets from five application domains: text classification (six datasets), image classification (two datasets), gene function prediction (one dataset), music classification (one dataset) and video classification (one dataset). We then analyze the results from the experiments using Friedman and Nemenyi tests for assessing the statistical significance of the differences in performance. We present the results from the statistical tests using critical diagrams.

The results of the experimental comparison revealed that the best performing methods are RF-PCT and HOMER, followed by BR and CC. For each performance measure, the best algorithm was either RF-PCT or HOMER. The example-based measures, which are most widely used for multi-label classification, show that RF-PCT is best according to *precision* and has average performance on *recall*. On the other hand, HOMER is best according to *recall*, while having poor performance on *precision*. This means that the predictions from RF-PCT are more exact than the ones from HOMER, while the predictions from HOMER are more complete than the ones from RF-PCT. Considering the basic machine learning algorithms underlying the compared approaches, the SVM based methods are better on datasets with a large number of features and a smaller number of examples, since they can exploit the information from all of the features, while the decision trees exploit only a subset of the features.

The label-based measures showed behavior similar to that of the example-based measures. However, the gap between HOMER and RF-PCT on *recall* and *precision* is now much bigger. Namely, RF-PCT is statistically significantly better than HOMER on the two precision-based measures.

The ranking-based measures offer a different perspective on the results. RF-PCT was the best performing method, followed by CC and BR. On these measures HOMER exhibited poor performance. RF-PCT is statistically significantly better than HOMER on two evaluation measures (*coverage* and *ranking loss*) using the datasets for which there are results from all methods and statistically significantly better on all four evaluation measures using all datasets. Furthermore, these measures emphasize the

advantages of the SVM-based methods on the smaller datasets and the tree-based methods on the larger datasets.

Considering efficiency, the tree-based methods are generally faster to train a classifier and produce a prediction for an unseen example than the SVM-based methods. We further compare the efficiency of the RF-PCT and HOMER methods. The results show that RF-PCT is faster than HOMER on testing time (on average 194.6 times) and on training time (on average 8.4 times).

The experimental comparison can be extended by including more methods for multi-label learning. For example, one can also include bagging of PCTs: ensemble method that has competitive performance to random forests of PCTs [18]. The comparison can be also extended by including other evaluation measures. One evaluation measure that can be easily adapted for multi-label setting is the precision-recall curve (and the area under the precision-recall curve thereof) [47]. This will offer a better insight to the trade-off between the precision and recall performance of a given method for multi-label learning.

All in all, the final recommendation considering the performance and the efficiency of the evaluated methods is that RF-PCT, HOMER, BR and CC should be used as benchmark methods for multi-label learning. Over all evaluation measures these methods performed best. Furthermore, RF-PCT and HOMER exhibited the best predictive performance and better efficiency than the rest of the methods.

Appendix A. Evaluation measures

In this section, we present the measures that are used to evaluate the predictive performance of the compared methods in our experiments. In the definitions below, $\mathcal{Y}_i$ denotes the set of true labels of example $\mathbf{x_i}$ and $h(\mathbf{x_i})$ denotes the set of predicted labels for the same examples. All definitions refer to the multi-label setting.

A.1. Example based measures

Hamming loss evaluates how many times an example-label pair is misclassified, i.e., label not belonging to the example is predicted or a label belonging to the example is not predicted. The smaller the value of *hamming_loss*(h), the better the performance. The performance is perfect when *hamming_loss*(h) $=0$. This metric is defined as

$$hamming\_loss(h)=\frac{1}{N}\sum_{i=1}^{N}\frac{1}{Q}\left|h(\mathbf{x_i})\Delta\mathcal{Y}_i\right| \tag{A.1}$$

where Δ stands for the symmetric difference between two sets, N is the number of examples and Q is the total number of possible class labels.

Accuracy for a single example $\mathbf{x_i}$ is defined by the Jaccard similarity coefficients between the label sets $h(\mathbf{x_i})$ and $\mathcal{Y}_i$. Accuracy is micro-averaged across all examples:

$$accuracy(h)=\frac{1}{N}\sum_{i=1}^{N}\frac{\left|h(\mathbf{x_i})\cap\mathcal{Y}_i\right|}{\left|h(\mathbf{x_i})\bigcup\mathcal{Y}_i\right|} \tag{A.2}$$

Precision is defined as

$$precision(h)=\frac{1}{N}\sum_{i=1}^{N}\frac{\left|h(\mathbf{x_i})\cap\mathcal{Y}_i\right|}{\left|\mathcal{Y}_i\right|} \tag{A.3}$$

Recall is defined as

$$recall(h)=\frac{1}{N}\sum_{i=1}^{N}\frac{\left|h(\mathbf{x_i})\cap\mathcal{Y}_i\right|}{\left|h(\mathbf{x_i})\right|} \tag{A.4}$$

F_1 *score* is the harmonic mean between precision and recall and is defined as

$$F_1=\frac{1}{N}\sum_{i=1}^{N}\frac{2\times\left|h(\mathbf{x_i})\cap\mathcal{Y}_i\right|}{\left|h(\mathbf{x_i})\right|+\left|\mathcal{Y}_i\right|} \tag{A.5}$$

F_1 is an example based metric and its value is an average over all examples in the dataset. F_1 reaches its best value at 1 and worst score at 0.

Subset accuracy or classification accuracy is defined as follows:

$$subset\_accuracy(h)=\frac{1}{N}\sum_{i=1}^{N}I(h(\mathbf{x_i})=\mathcal{Y}_i) \tag{A.6}$$

where $I(true)=1$ and $I(false)=0$. 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.

A.2. Label based measures

Macro-precision (precision averaged across all labels) is defined as

$$macro\_precision=\frac{1}{Q}\sum_{j=1}^{Q}\frac{tp_j}{tp_j+fp_j} \tag{A.7}$$

where tp_j and fp_j are the number of true positives and false positives for the label λ_j considered as a binary class.

Macro-recall (recall averaged across all labels) is defined as

$$macro\_recall=\frac{1}{Q}\sum_{j=1}^{Q}\frac{tp_j}{tp_j+fn_j} \tag{A.8}$$

where tp_j and fp_j are defined as for the macro-precision and fn_j is the number of false negatives for the label λ_j considered as a binary class.

Macro-F_1 is the harmonic mean between precision and recall, where the average is calculated per label and then averaged across all labels. If p_j and r_j are the precision and recall for all $\lambda_j\in h(\mathbf{x_i})$ from $\lambda_j\in\mathcal{Y}_i$, the macro-$F_1$ is

$$macro\_F_1=\frac{1}{Q}\sum_{j=1}^{Q}\frac{2\times p_j\times r_j}{p_j+r_j} \tag{A.9}$$

Micro-precision (precision averaged over all the example/label pairs) is defined as

$$micro\_precision=\frac{\sum_{j=1}^{Q}tp_j}{\sum_{j=1}^{Q}tp_j+\sum_{j=1}^{Q}fp_j} \tag{A.10}$$

where tp_j, fp_j are defined as for macro-precision.

Micro-recall (recall averaged over all the example/label pairs) is defined as

$$micro\_recall=\frac{\sum_{j=1}^{Q}tp_j}{\sum_{j=1}^{Q}tp_j+\sum_{j=1}^{Q}fn_j} \tag{A.11}$$

where tp_j and fn_j are defined as for macro-recall.

Micro-F_1 is the harmonic mean between *micro-precision* and *micor-recall*. Micro-F_1 is defined as

$$micro\_F_1=\frac{2\times micro\_precision\times micro\_recall}{micro\_precision+micro\_recall} \tag{A.12}$$

A.3. Ranking based measures

One error evaluates how many times the top-ranked label is not in the set of relevant labels of the example. The metric *one_error*(f) takes values between 0 and 1. The smaller the value

of *one_error(f)*, the better the performance. This evaluation metric is defined as

$$one\_error(f) = \frac{1}{N}\sum_{i=1}^{N} \left[\!\!\left[\left[\arg\max_{\lambda \in \mathcal{Y}} f(\mathbf{x_i},\lambda)\right] \notin \mathcal{Y}_i \right]\!\!\right] \quad (A.13)$$

where $\lambda \in \mathcal{L} = \{\lambda_1,\lambda_2,\ldots,\lambda_Q\}$ and $[\![\pi]\!]$ equals 1 if π holds and 0 otherwise for any predicate π. Note that, for single-label classification problems, the One Error is identical to ordinary classification error.

Coverage evaluates how far, on average, we need to go down the list of ranked labels in order to cover all the relevant labels of the example. The smaller the value of *coverage(f)*, the better the performance:

$$coverage(f) = \frac{1}{N}\sum_{i=1}^{N} \max_{\lambda \in \mathcal{Y}_i} rank_f(\mathbf{x_i},\lambda) - 1 \quad (A.14)$$

where $rank_f(\mathbf{x_i},\lambda)$ maps the outputs of $f(\mathbf{x_i},\lambda)$ for any $\lambda \in \mathcal{L}$ to $\{\lambda_1,\lambda_2,\ldots,\lambda_Q\}$ so that $f(\mathbf{x_i},\lambda_m) > f(\mathbf{x_i},\lambda_n)$ implies $rank_f(\mathbf{x_i},\lambda_m) < rank_f(\mathbf{x_i},\lambda_n)$. The smallest possible value for *coverage(f)* is l_c, i.e., the label cardinality of the given dataset.

Ranking loss evaluates the average fraction of label pairs that are reversely ordered for the particular example given by

$$ranking\ loss(f) = \frac{1}{N}\sum_{i=1}^{N} \frac{|D_i|}{|\mathcal{Y}_i|\ |\overline{\mathcal{Y}}_i|} \quad (A.15)$$

where $D_i = \{(\lambda_m,\lambda_n) | f(\mathbf{x_i},\lambda_m) \le f(\mathbf{x_i},\lambda_n), (\lambda_m,\lambda_n) \in \mathcal{Y}_i \times \overline{\mathcal{Y}}_i\}$, while $\overline{\mathcal{Y}}$ denotes the complementary set of $\mathcal{Y}$ in $\mathcal{L}$. The smaller the value of *ranking_loss(f)*, the better the performance, so the performance is perfect when $ranking\_loss(f) = 0$.

Average precision is the average fraction of labels ranked above an actual label $\lambda \in \mathcal{Y}_i$ that actually are in $\mathcal{Y}_i$. The performance is perfect when $avg\_precision(f) = 1$; the larger the value of *avg_precision(f)*, the better the performance. This metric is defined as

$$avg\_precision(f) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{|\mathcal{Y}_i|} \sum_{\lambda \in \mathcal{Y}_i} \frac{|\mathcal{L}_i|}{rank_f(\mathbf{x_i},\lambda)} \quad (A.16)$$

where $\mathcal{L}_i = \{\lambda' | rank_f(\mathbf{x_i},\lambda') \le rank_f(\mathbf{x_i},\lambda), \lambda' \in \mathcal{Y}_i\}$ and $rank_f(\mathbf{x_i},\lambda)$ is defined as in coverage above.

Appendix B. Complete results from the experimental evaluation

In this section, we present the complete results from the experimental evaluation. We present the results based on the evaluation measures. We first present the results for the example-based evaluation measures. We then show the results for label-based evaluation measures. We next give the results for ranking-based evaluation measure. Finally, we present the efficiency of the methods by their training and testing times.

Table B1
The performance of the multi-label learning approaches in terms of the *Hamming loss* measure. DNF (*Did Not Finish*) stands for the algorithms that did not construct a predictive model within one week under the available resources.

Dataset	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-*k*NN	RAkEL	ECC	RFML-C4.5	RF-PCT
emotions	0.257	0.256	0.257	0.254	0.361	0.247	0.267	0.294	0.282	0.281	0.198	**0.189**
scene	0.079	0.082	0.080	0.081	0.082	0.141	0.129	0.099	**0.077**	0.085	0.116	0.094
yeast	**0.190**	0.193	**0.190**	0.191	0.207	0.234	0.219	0.198	0.192	0.207	0.205	0.197
medical	0.077	0.077	0.017	**0.012**	**0.012**	0.013	0.023	0.017	**0.012**	0.014	0.022	0.014
enron	**0.045**	0.064	0.048	0.048	0.051	0.053	0.058	0.051	**0.045**	0.049	0.047	0.046
corel5k	0.017	0.017	0.012	0.012	0.012	0.010	**0.009**	**0.009**	**0.009**	**0.009**	**0.009**	**0.009**
tmc2007	0.013	0.013	0.014	0.014	0.015	0.093	0.075	0.058	0.021	0.026	0.037	**0.011**
mediamill	0.032	0.032	0.043	0.043	0.038	0.044	0.034	0.031	0.035	0.035	0.030	**0.029**
bibtex	**0.012**	**0.012**	**0.012**	**0.012**	0.014	0.016	0.014	0.014	DNF	0.013	0.014	0.013
delicious	**0.018**	**0.018**	DNF	DNF	0.022	0.019	0.019	**0.018**	DNF	DNF	0.018	0.018
bookmarks	DNF	DNF	DNF	DNF	DNF	**0.009**	**0.009**	**0.009**	DNF	DNF	**0.009**	**0.009**

Table B2
The performance of the multi-label learning approaches in terms of the *accuracy* measure. DNF (*Did Not Finish*) stands for the algorithms that did not construct a predictive model within one week under the available resources.

Dataset	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-*k*NN	RAkEL	ECC	RFML-C4.5	RF-PCT
emotions	0.361	0.356	0.361	0.373	0.471	**0.536**	0.448	0.319	0.419	0.432	0.488	0.519
scene	0.689	0.723	0.686	0.683	0.717	0.569	0.538	0.629	0.734	**0.735**	0.388	0.541
yeast	0.520	0.527	0.524	0.523	**0.559**	0.480	0.440	0.492	0.531	0.546	0.453	0.478
medical	0.206	0.211	0.656	0.658	0.713	**0.730**	0.228	0.528	0.673	0.611	0.250	0.591
enron	0.446	0.334	0.459	0.388	**0.478**	0.418	0.196	0.319	0.428	0.462	0.374	0.416
corel5k	0.030	0.030	**0.195**	**0.195**	0.179	0.002	0.000	0.014	0.000	0.001	0.005	0.009
tmc2007	0.891	0.899	0.889	0.889	0.888	0.110	0.436	0.574	0.852	0.808	0.663	**0.914**
mediamill	0.403	0.390	0.095	0.095	0.413	0.052	0.354	0.421	0.337	0.349	0.423	**0.441**
bibtex	0.348	**0.352**	0.334	0.338	0.330	0.108	0.046	0.129	DNF	0.186	0.060	0.166
delicious	0.136	0.137	DNF	DNF	**0.207**	0.001	0.001	0.102	DNF	DNF	0.151	0.146
bookmarks	DNF	DNF	DNF	DNF	DNF	**0.237**	0.133	0.202	DNF	DNF	0.176	0.204

Table B3
The performance of the multi-label learning approaches in terms of the *precision* measure. DNF (*Did Not Finish*) stands for the algorithms that did not construct a predictive model within one week under the available resources.

Dataset	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-*k*NN	RAkEL	ECC	RFML-C4.5	RF-PCT
emotions	0.550	0.551	0.538	0.548	0.509	0.606	0.577	0.502	0.564	0.580	0.625	**0.644**
scene	0.718	0.758	0.714	0.711	0.746	0.592	0.565	0.661	0.768	**0.770**	0.403	0.565
yeast	0.722	0.727	0.719	0.718	0.663	0.620	0.705	0.732	0.715	0.667	0.738	**0.744**
medical	0.211	0.217	0.695	0.697	0.762	**0.797**	0.285	0.575	0.730	0.662	0.284	0.635
enron	0.703	0.464	0.650	0.624	0.616	0.623	0.415	0.587	0.708	0.652	0.690	**0.709**
corel5k	0.042	0.042	**0.329**	0.326	0.317	0.005	0.000	0.035	0.000	0.002	0.018	0.030
tmc2007	0.941	0.944	0.937	0.937	0.926	0.146	0.659	0.738	0.928	0.872	0.874	**0.977**
mediamill	0.731	0.741	0.201	0.203	0.597	0.056	0.694	0.724	0.705	0.690	0.765	**0.772**
bibtex	**0.515**	0.508	0.488	0.496	0.472	0.123	0.140	0.254	DNF	0.324	0.159	0.292
delicious	0.443	0.399	DNF	DNF	0.369	0.001	0.001	0.424	DNF	DNF	0.472	**0.512**
bookmarks	DNF	DNF	DNF	DNF	DNF	**0.271**	0.133	0.218	DNF	DNF	0.182	0.218

Table B4
The performance of the multi-label learning approaches in terms of the *recall* measure. DNF (*Did Not Finish*) stands for the algorithms that did not construct a predictive model within one week under the available resources.

Dataset	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-*k*NN	RAkEL	ECC	RFML-C4.5	RF-PCT
emotions	0.409	0.397	0.410	0.429	**0.775**	0.703	0.534	0.377	0.491	0.533	0.545	0.582
scene	0.711	0.726	0.712	0.709	0.744	0.582	0.539	0.655	0.740	**0.771**	0.388	0.541
yeast	0.591	0.600	0.601	0.600	**0.714**	0.608	0.490	0.549	0.615	0.673	0.491	0.523
medical	0.735	0.754	0.795	**0.801**	0.760	0.740	0.227	0.547	0.679	0.642	0.251	0.599
enron	0.497	0.507	0.557	0.453	**0.610**	0.487	0.229	0.358	0.469	0.560	0.398	0.452
corel5k	0.055	0.056	**0.264**	0.264	0.250	0.002	0.000	0.014	0.000	0.001	0.005	0.009
tmc2007	0.928	0.934	0.929	0.929	**0.943**	0.111	0.478	0.664	0.880	0.903	0.677	0.920
mediamill	0.450	0.424	0.101	0.101	**0.563**	0.052	0.379	0.470	0.353	0.372	0.456	0.476
bibtex	0.373	0.378	0.364	0.366	**0.389**	0.111	0.046	0.132	DNF	0.187	0.060	0.167
delicious	0.155	0.157	DNF	DNF	**0.303**	0.001	0.001	0.112	DNF	DNF	0.176	0.160
bookmarks	DNF	DNF	DNF	DNF	DNF	**0.244**	0.137	0.207	DNF	DNF	0.181	0.208

Table B5
The performance of the multi-label learning approaches in terms of the *subset accuracy* measure. DNF (*Did Not Finish*) stands for the algorithms that did not construct a predictive model within one week under the available resources.

Dataset	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-*k*NN	RAkEL	ECC	RFML-C4.5	RF-PCT
emotions	0.129	0.124	0.144	0.149	0.163	0.277	0.223	0.084	0.208	0.168	0.272	**0.307**
scene	0.639	0.685	0.633	0.630	0.661	0.533	0.509	0.573	**0.694**	0.665	0.372	0.518
yeast	0.190	**0.239**	0.195	0.192	0.213	0.158	0.152	0.159	0.201	0.215	0.129	0.152
medical	0.000	0.000	0.486	0.480	0.610	**0.646**	0.177	0.462	0.607	0.526	0.216	0.538
enron	**0.149**	0.000	0.117	0.097	0.145	0.140	0.002	0.062	0.136	0.131	0.124	0.131
corel5k	0.000	0.000	0.010	**0.012**	0.002	0.000	0.000	0.000	0.000	0.001	0.008	0.000
tmc2007	0.772	0.787	0.767	0.768	0.765	0.078	0.215	0.305	0.734	0.608	0.421	**0.816**
mediamill	0.080	0.080	0.044	0.044	0.053	0.049	0.065	0.110	0.060	0.065	0.104	**0.122**
bibtex	0.194	**0.202**	0.183	0.186	0.165	0.095	0.004	0.056	DNF	0.109	0.011	0.098
delicious	0.004	0.006	DNF	DNF	0.001	0.001	0.001	0.003	DNF	DNF	**0.018**	0.007
bookmarks	DNF	DNF	DNF	DNF	DNF	**0.209**	0.129	0.187	DNF	DNF	0.167	0.189

Table B6
The performance of the multi-label learning approaches in terms of the F_1 *score* measure. DNF (*Did Not Finish*) stands for the algorithms that did not construct a predictive model within one week under the available resources.

Dataset	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-*k*NN	RAkEL	ECC	RFML-C4.5	RF-PCT
emotions	0.469	0.461	0.465	0.481	0.614	**0.651**	0.554	0.431	0.525	0.556	0.583	0.611
scene	0.714	0.742	0.713	0.710	0.745	0.587	0.551	0.658	0.754	**0.771**	0.395	0.553
yeast	0.650	0.657	0.655	0.654	**0.687**	0.614	0.578	0.628	0.661	0.670	0.589	0.614
medical	0.328	0.337	0.742	0.745	0.761	**0.768**	0.253	0.560	0.704	0.652	0.267	0.616
enron	0.582	0.484	0.600	0.525	**0.613**	0.546	0.295	0.445	0.564	0.602	0.505	0.552
corel5k	0.047	0.048	**0.293**	0.292	0.280	0.003	0.000	0.021	0.000	0.001	0.008	0.014
tmc2007	0.934	0.939	0.933	0.933	0.934	0.126	0.554	0.699	0.904	0.887	0.763	**0.948**
mediamill	0.557	0.539	0.134	0.135	0.579	0.054	0.490	0.570	0.471	0.483	0.572	**0.589**
bibtex	0.433	**0.434**	0.417	0.421	0.426	0.117	0.069	0.174	DNF	0.237	0.087	0.212
delicious	0.230	0.225	DNF	DNF	**0.343**	0.001	0.001	0.017	DNF	DNF	0.256	0.244
bookmarks	DNF	DNF	DNF	DNF	DNF	**0.257**	0.135	0.213	DNF	DNF	0.181	0.213

Table B7
The *p*-values of the assessment of performance of the multi-label learning approaches by the Friedman test using the example-based evaluation measures. *Subset* shows the calculated *p*-values for the datasets on which all algorithms finished. *All* shows the calculated *p*-values for all datasets including those which did not finished.

Evaluation measure	All	Subset
Hamming loss	0.0895	0.047
Accuracy	0.077	0.037
Precision	0.117	0.19
Recall	5×10^{-4}	7.9×10^{-5}
F_1 score	5×10^{-3}	0.0013
Subset accuracy	0.555	0.343

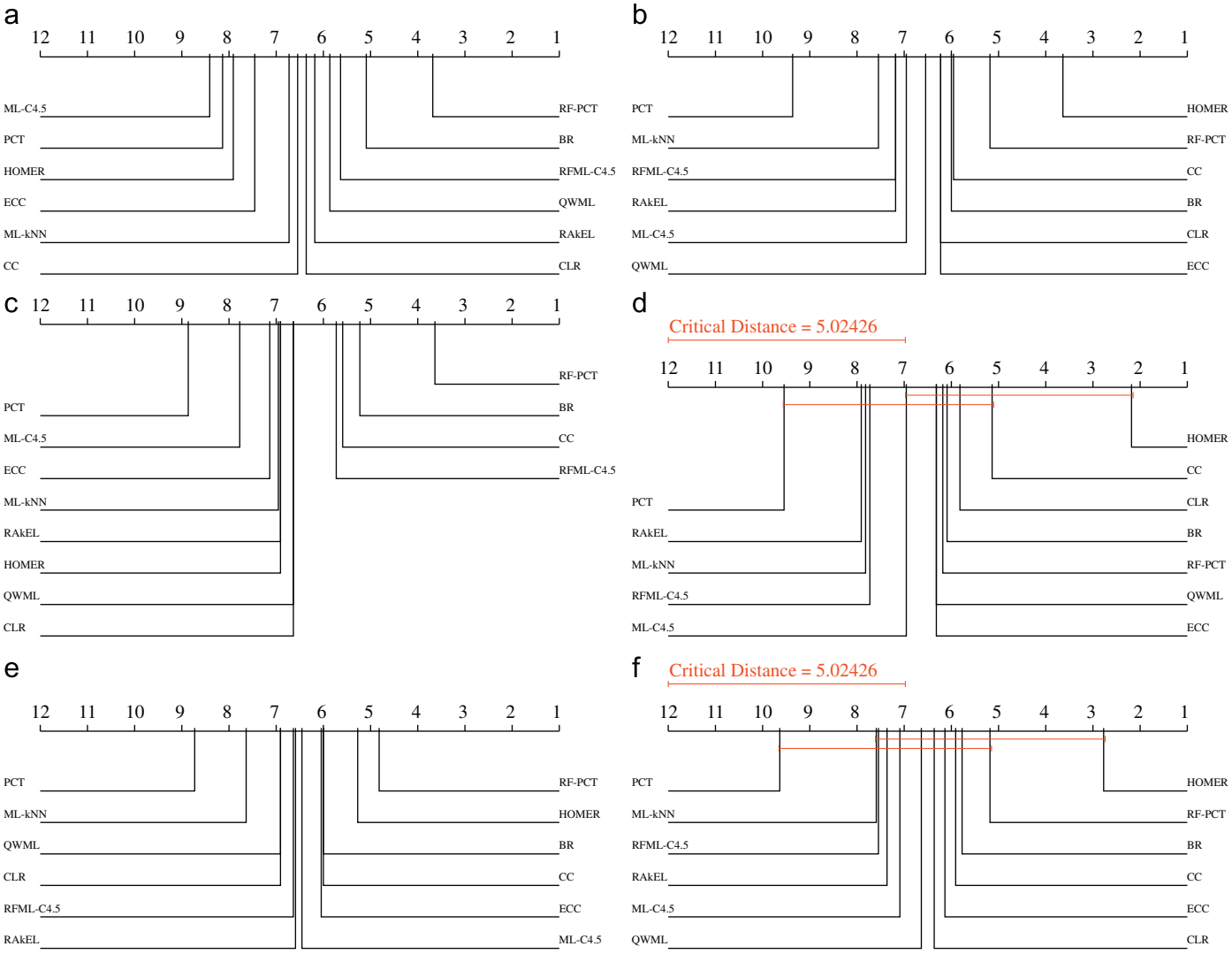

Fig. B1. The critical diagrams for the example-based evaluation measures: the results from the Nemenyi post-hoc test at 0.05 significance level on all the datasets. For *Hamming loss*, *precision*, *accuracy* and *subset accuracy* the differences are not statistically significant according to the Friedman test (see Table B7), thus we show only the average ranks of the algorithms: (a) *Hamming loss*; (b) *accuracy*; (c) *precision*; (d) *recall*; (e) *subset accuracy*; and (f) F_1 *score*.

Table B8
The performance of the multi-label learning approaches in terms of the *micro-precision* measure. DNF (*Did Not Finish*) stands for the algorithms that did not construct a predictive model within one week under the available resources.

Dataset	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-*k*NN	RAkEL	ECC	RFML-C4.5	RF-PCT
emotions	0.684	0.698	0.685	0.680	0.471	0.607	0.607	0.584	0.586	0.579	**0.783**	**0.783**
scene	0.843	0.814	0.835	0.832	0.804	0.619	0.512	0.691	0.831	0.773	**0.960**	0.930
yeast	0.733	0.726	0.729	0.727	0.647	0.618	0.698	0.736	0.720	0.662	0.747	**0.755**
medical	0.225	0.229	0.669	0.667	0.807	0.796	0.826	0.807	0.881	0.834	0.884	**0.885**
enron	0.721	0.492	0.652	0.687	0.597	0.613	0.601	0.684	0.743	0.642	**0.768**	0.738
corel5k	0.061	0.061	0.338	0.339	0.308	0.160	0.000	0.730	0.000	0.333	**0.750**	0.696
tmc2007	0.947	0.948	0.940	0.941	0.922	0.940	0.689	0.757	0.938	0.869	0.963	**0.992**
mediamill	0.742	0.753	0.582	0.580	0.569	0.597	0.743	0.739	0.725	0.708	0.788	**0.798**
bibtex	0.753	0.744	0.734	0.736	0.547	0.359	**1.000**	0.819	DNF	0.948	0.940	0.957
delicious	0.658	0.660	DNF	DNF	0.396	0.000	0.000	0.651	DNF	DNF	0.589	**0.695**
bookmarks	DNF	DNF	DNF	DNF	DNF	0.632	**0.947**	0.850	DNF	DNF	0.878	0.895

Table B9
The performance of the multi-label learning approaches in terms of the *macro-precision* measure. DNF (*Did Not Finish*) stands for the algorithms that did not construct a predictive model within one week under the available resources.

Dataset	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-*k*NN	RAkEL	ECC	RFML-C4.5	RF-PCT
emotions	0.721	0.581	0.677	0.660	0.464	0.602	0.628	0.518	0.547	0.531	**0.828**	0.802
scene	0.844	0.817	0.835	0.832	0.807	0.635	0.682	0.784	0.835	0.785	**0.963**	0.919
yeast	0.628	0.602	0.614	0.614	0.471	0.377	0.479	0.600	0.480	0.391	0.533	**0.674**
medical	**0.399**	0.391	0.288	0.285	0.287	0.263	0.018	0.267	0.269	0.266	0.190	0.269
enron	0.258	**0.260**	0.205	0.242	0.241	0.142	0.023	0.170	0.222	0.249	0.245	0.233
corel5k	0.052	0.053	**0.059**	**0.059**	0.044	0.004	0.000	0.031	0.000	0.001	0.007	0.015
tmc2007	0.972	0.972	0.964	0.965	0.954	0.925	0.386	0.780	0.973	0.938	0.994	**0.997**
mediamill	0.112	0.144	0.140	0.133	0.107	0.046	0.401	0.308	0.025	0.037	0.397	**0.441**
bibtex	0.528	**0.539**	0.503	0.490	0.391	0.128	0.006	0.192	DNF	0.121	0.080	0.127
delicious	0.299	0.303	DNF	DNF	0.154	0.000	0.000	0.134	DNF	DNF	**0.422**	0.293
bookmarks	DNF	DNF	DNF	DNF	DNF	0.292	0.018	0.414	DNF	DNF	0.388	**0.522**

Table B10
The performance of the multi-label learning approaches in terms of the *micro-recall* measure. DNF (*Did Not Finish*) stands for the algorithms that did not construct a predictive model within one week under the available resources.

Dataset	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-*k*NN	RAkEL	ECC	RFML-C4.5	RF-PCT
emotions	0.406	0.393	0.409	0.431	**0.782**	0.712	0.539	0.376	0.489	0.531	0.551	0.589
scene	0.694	0.708	0.695	0.692	0.727	0.570	0.521	0.634	0.721	**0.751**	0.572	0.523
yeast	0.587	0.588	0.595	0.595	**0.702**	0.603	0.492	0.543	0.602	0.655	0.491	0.521
medical	0.725	0.739	0.782	**0.787**	0.742	0.720	0.227	0.522	0.600	0.624	0.237	0.569
enron	0.464	0.472	0.532	0.438	**0.585**	0.440	0.246	0.353	0.435	0.532	0.366	0.422
corel5k	0.057	0.057	**0.258**	**0.258**	0.248	0.002	0.000	0.015	0.000	0.001	0.005	0.009
tmc2007	0.917	0.924	0.920	0.920	**0.932**	0.073	0.454	0.621	0.847	0.869	0.651	0.902
mediamill	0.415	0.385	0.066	0.066	**0.537**	0.004	0.351	0.432	0.315	0.333	0.418	0.435
bibtex	0.328	0.335	0.322	0.328	**0.353**	0.053	0.057	0.118	DNF	0.142	0.066	0.131
delicious	0.143	0.144	DNF	DNF	0.297	0.000	0.000	0.101	DNF	DNF	0.174	0.151
bookmarks	DNF	DNF	DNF	DNF	DNF	**0.170**	0.135	0.135	DNF	DNF	0.112	0.136

Table B11
The performance of the multi-label learning approaches in terms of the *macro-recall* measure. DNF (*Did Not Finish*) stands for the algorithms that did not construct a predictive model within one week under the available resources.

Dataset	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-*k*NN	RAkEL	ECC	RFML-C4.5	RF-PCT
emotions	0.378	0.364	0.381	0.398	**0.775**	0.702	0.533	0.334	0.462	0.508	0.532	0.569
scene	0.703	0.716	0.704	0.701	0.734	0.573	0.529	0.647	0.727	**0.757**	0.381	0.533
yeast	0.355	0.357	0.361	0.361	**0.466**	0.375	0.269	0.308	0.352	0.388	0.257	0.286
medical	0.423	**0.428**	0.307	0.324	0.282	0.249	0.022	0.163	0.183	0.179	0.040	0.176
enron	0.120	0.146	0.139	0.120	**0.163**	0.107	0.030	0.075	0.097	0.129	0.082	0.100
corel5k	0.023	0.023	0.039	0.039	**0.041**	0.005	0.000	0.006	0.000	0.001	0.001	0.002
tmc2007	0.915	**0.924**	0.914	0.914	0.897	0.085	0.235	0.418	0.739	0.772	0.297	0.769
mediamill	0.049	0.044	0.028	0.028	0.074	0.002	0.029	**0.088**	0.020	0.023	0.065	0.080
bibtex	0.250	**0.257**	0.236	0.238	0.247	0.034	0.006	0.049	DNF	0.044	0.013	0.043
delicious	0.072	0.075	DNF	DNF	**0.103**	0.000	0.000	0.039	DNF	DNF	0.092	0.060
bookmarks	DNF	DNF	DNF	DNF	DNF	**0.098**	0.016	0.070	DNF	DNF	0.048	0.072

Table B12
The performance of the multi-label learning approaches in terms of the *micro-*F_1 measure. DNF (*Did Not Finish*) stands for the algorithms that did not construct a predictive model within one week under the available resources.

Dataset	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-*k*NN	RAkEL	ECC	RFML-C4.5	RF-PCT
emotions	0.509	0.503	0.512	0.528	0.588	0.655	0.571	0.457	0.533	0.554	0.647	0.672
scene	0.761	0.757	0.758	0.756	0.764	0.593	0.516	0.661	**0.772**	0.762	0.717	0.669
yeast	0.652	0.650	0.655	0.654	**0.673**	0.610	0.577	0.625	0.656	0.658	0.593	0.617
medical	0.343	0.350	0.721	0.722	**0.773**	0.756	0.356	0.634	0.714	0.714	0.374	0.693
enron	0.564	0.482	0.585	0.535	**0.591**	0.512	0.349	0.466	0.548	0.582	0.496	0.537
corel5k	0.059	0.059	**0.293**	**0.293**	0.275	0.004	0.000	0.030	0.000	0.002	0.010	0.018
tmc2007	0.932	0.936	0.930	0.930	0.927	0.135	0.547	0.682	0.890	0.869	0.777	**0.945**
mediamill	0.533	0.509	0.118	0.119	0.553	0.007	0.477	0.545	0.440	0.453	0.546	**0.563**
bibtex	0.457	**0.462**	0.448	0.454	0.429	0.093	0.108	0.206	DNF	0.247	0.123	0.230
delicious	0.234	0.236	DNF	DNF	**0.339**	0.000	0.000	0.175	DNF	DNF	0.269	0.248
bookmarks	DNF	DNF	DNF	DNF	DNF	**0.268**	0.236	0.232	DNF	DNF	0.199	0.236

Table B13
The performance of the multi-label learning approaches in terms of the *macro-F_1* measure. DNF (*Did Not Finish*) stands for the algorithms that did not construct a predictive model within one week under the available resources.

Dataset	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-*k*NN	RAkEL	ECC	RFML-C4.5	RF-PCT
emotions	0.440	0.420	0.443	0.458	0.570	0.630	0.568	0.385	0.488	0.500	0.620	**0.650**
scene	0.765	0.762	0.762	0.759	0.768	0.596	0.593	0.692	**0.777**	0.770	0.514	0.658
yeast	0.392	0.390	0.392	0.394	**0.447**	0.370	0.293	0.336	0.359	0.350	0.283	0.322
medical	0.361	**0.371**	0.281	0.286	0.282	0.250	0.020	0.192	0.210	0.203	0.058	0.207
enron	0.143	0.153	0.149	0.143	**0.167**	0.115	0.026	0.087	0.115	0.140	0.102	0.122
corel5k	0.021	0.021	**0.042**	**0.042**	0.036	0.008	0.000	0.010	0.000	0.001	0.001	0.004
tmc2007	0.942	**0.947**	0.938	0.938	0.924	0.124	0.263	0.493	0.826	0.834	0.371	0.857
mediamill	0.056	0.052	0.037	0.037	0.073	0.003	0.031	**0.113**	0.019	0.022	0.088	0.112
bibtex	0.307	**0.316**	0.291	0.292	0.266	0.045	0.006	0.065	DNF	0.052	0.016	0.055
delicious	0.096	0.100	DNF	DNF	0.103	0.000	0.000	0.051	DNF	DNF	**0.142**	0.083
bookmarks	DNF	DNF	DNF	DNF	DNF	**0.119**	0.017	0.096	DNF	DNF	0.065	0.101

Table B14
The *p*-values of the assessment of performance of the multi-label learning approaches by the Friedman test using the label-based evaluation measures. *Subset* shows the calculated *p*-values for the datasets on which all algorithms finished. *All* shows the calculated *p*-values for all datasets including those which did not finished.

Evaluation measure	All	Subset
Macro-precision	3.5×10^{-7}	4.8×10^{-7}
Macro-recall	2.8×10^{-4}	1.1×10^{-4}
Macro-F_1	3.1×10^{-4}	9.8×10^{-5}
Micro-precision	3.7×10^{-9}	3.4×10^{-8}
Micro-recall	3.6×10^{-4}	7.3×10^{-5}
Micro-F_1	0.011	0.0022

Table B15
The performance of the multi-label learning approaches in terms of the *ranking loss* measure. DNF (*Did Not Finish*) stands for the algorithms that did not construct a predictive model within one week under the available resources.

Dataset	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-*k*NN	RAkEL	ECC	RFML-C4.5	RF-PCT
emotions	0.246	0.245	0.264	0.331	0.297	0.210	0.270	0.283	0.281	0.310	0.153	**0.151**
scene	**0.060**	0.064	0.065	0.103	0.119	0.169	0.174	0.093	0.104	0.103	0.079	0.072
yeast	0.164	0.170	**0.163**	0.296	0.205	0.225	0.199	0.172	0.259	0.224	0.173	0.167
medical	0.021	**0.019**	0.028	0.027	0.090	0.048	0.104	0.045	0.159	0.152	0.028	0.024
enron	0.084	0.083	**0.078**	0.177	0.183	0.120	0.114	0.093	0.283	0.238	0.083	0.079
corel5k	0.117	0.118	**0.100**	0.245	0.352	0.479	0.140	0.130	0.673	0.749	0.122	0.117
tmc2007	**0.003**	**0.003**	0.005	0.039	0.028	0.043	0.100	0.031	0.031	0.032	0.007	0.006
mediamill	0.061	0.062	0.092	0.101	0.177	0.073	0.063	0.055	0.236	0.258	**0.047**	**0.047**
bibtex	0.068	0.067	**0.065**	0.207	0.255	0.260	0.255	0.217	DNF	0.394	0.126	0.093
delicious	0.114	0.117	DNF	DNF	0.379	0.174	0.172	0.129	DNF	DNF	0.140	**0.106**
bookmarks	DNF	DNF	DNF	DNF	DNF	0.194	0.258	0.181	DNF	DNF	0.129	**0.104**

Table B16
The performance of the multi-label learning approaches in terms of the *one-error* measure. DNF (*Did Not Finish*) stands for the algorithms that did not construct a predictive model within one week under the available resources.

Dataset	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-*k*NN	RAkEL	ECC	RFML-C4.5	RF-PCT
emotions	0.386	0.376	0.391	0.391	0.411	0.347	0.386	0.406	0.396	0.426	0.277	**0.262**
scene	**0.180**	0.204	0.190	0.193	0.216	0.394	0.389	0.242	0.197	0.213	0.232	0.210
yeast	0.236	0.268	**0.229**	0.233	0.248	0.312	0.264	0.234	0.254	0.249	0.250	0.248
medical	0.135	**0.123**	0.168	0.165	0.216	0.198	0.612	0.279	0.312	0.315	0.243	0.174
enron	0.237	0.238	0.231	0.269	0.314	0.309	0.392	0.280	0.290	0.247	**0.219**	0.221
corel5k	0.660	0.674	**0.588**	0.592	0.652	0.762	0.776	0.706	0.758	0.992	0.644	0.608
tmc2007	0.029	0.026	0.033	0.033	0.050	0.145	0.306	0.190	0.047	0.052	0.071	**0.006**
mediamill	0.188	0.193	0.586	0.560	0.219	0.194	0.220	0.182	0.234	0.242	0.171	**0.159**
bibtex	0.346	**0.342**	0.388	0.380	0.466	0.529	0.783	0.576	DNF	0.666	0.544	0.433
delicious	0.354	0.367	DNF	DNF	0.509	0.411	0.592	0.416	DNF	DNF	0.368	**0.332**
bookmarks	DNF	DNF	DNF	DNF	DNF	0.643	0.817	0.639	DNF	DNF	0.607	**0.541**

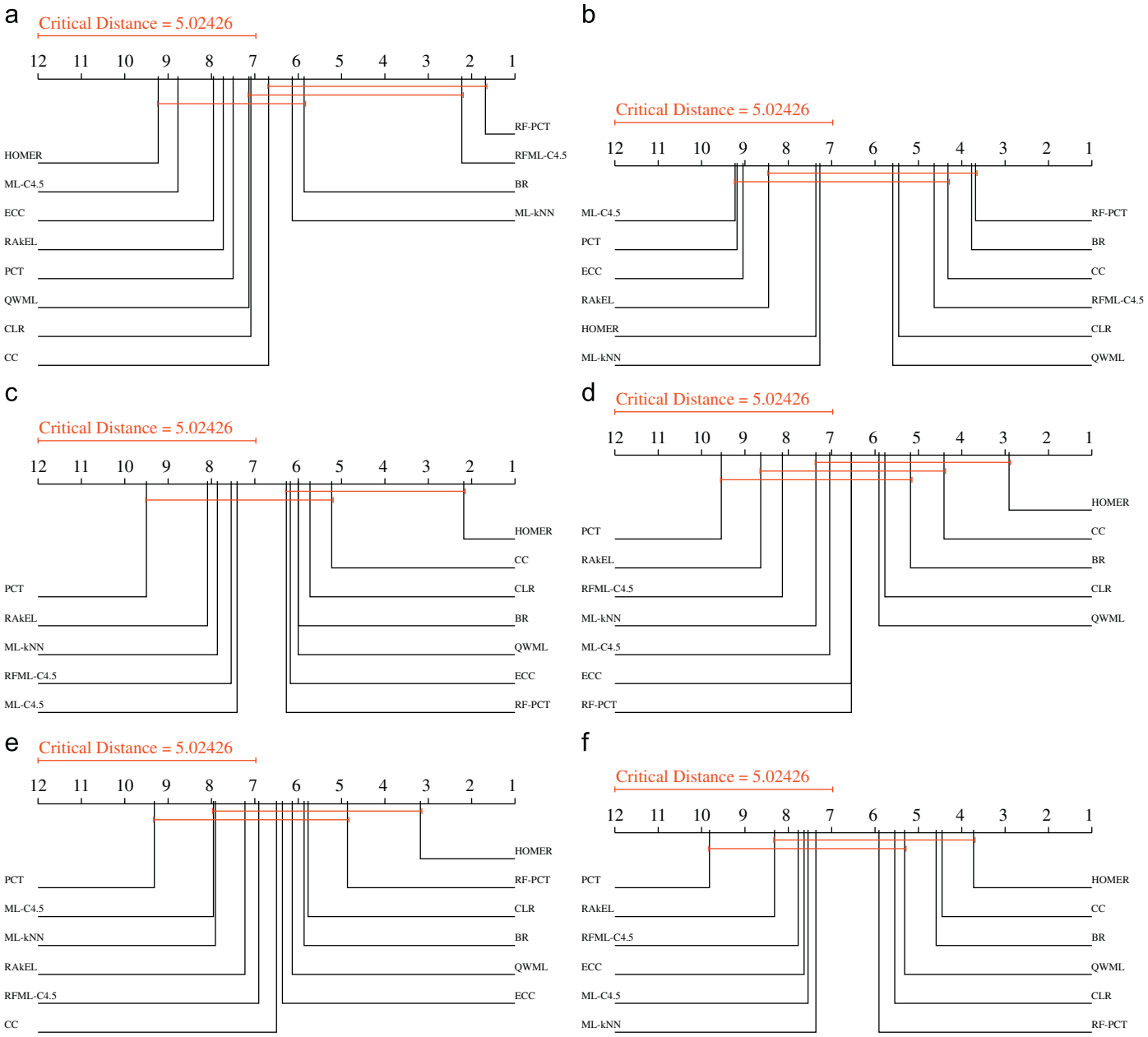

Fig. B2. The critical diagrams for the label-based evaluation measures: the results from the Nemenyi post-hoc test at 0.05 significance level on all the datasets: (a) *micro-precision*; (b) *macro-precision*; (c) *micro-recall*; (d) *macro-recall*; (e) *micro-F_1*; and (f) *macro-F_1*.

Table B17
The performance of the multi-label learning approaches in terms of the *coverage* measure. DNF (*Did Not Finish*) stands for the algorithms that did not construct a predictive model within one week under the available resources.

Dataset	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-*k*NN	RAkEL	ECC	RFML-C4.5	RF-PCT
emotions	2.307	2.317	2.386	2.807	2.634	2.069	2.356	2.490	2.465	2.619	**1.801**	1.827
scene	**0.399**	0.417	0.423	0.631	0.739	0.945	0.964	0.569	0.635	0.625	0.495	0.461
yeast	6.330	6.439	6.286	8.659	7.285	7.105	6.705	6.414	7.983	7.153	6.276	**6.179**
medical	1.610	**1.471**	2.036	1.832	5.324	3.033	5.813	2.844	8.520	7.994	1.889	1.619
enron	12.530	12.437	**11.763**	22.746	24.190	17.010	14.920	13.181	30.509	27.760	12.485	12.074
corel5k	104.800	105.428	**91.506**	206.880	250.800	279.900	115.676	113.046	340.398	348.160	110.356	107.412
tmc2007	1.311	1.302	1.363	2.796	2.369	2.671	4.572	2.155	2.498	2.494	1.416	**1.219**
mediamill	20.481	20.333	24.247	28.982	47.046	22.096	20.456	18.719	56.617	58.865	**16.868**	16.926
bibtex	20.926	21.078	**18.540**	57.343	65.626	58.016	58.599	56.266	DNF	87.841	32.580	25.854
delicious	530.126	537.388	DNF	DNF	933.956	620.155	691.622	589.898	DNF	DNF	624.572	**504.999**
bookmarks	DNF	DNF	DNF	DNF	DNF	58.353	73.780	54.528	DNF	DNF	40.903	**34.185**

Table B18
The performance of the multi-label learning approaches in terms of the *average precision* measure. DNF (*Did Not Finish*) stands for the algorithms that did not construct a predictive model within one week under the available resources.

Dataset	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-*k*NN	RAkEL	ECC	RFML-C4.5	RF-PCT
emotions	0.721	0.724	0.718	0.679	0.698	0.759	0.713	0.694	0.713	0.687	**0.812**	**0.812**
scene	**0.893**	0.881	0.886	0.864	0.848	0.751	0.745	0.851	0.862	0.856	0.862	0.874
yeast	0.768	0.755	**0.768**	0.698	0.740	0.706	0.724	0.758	0.715	0.734	0.749	0.757
medical	0.896	**0.901**	0.864	0.862	0.786	0.823	0.522	0.784	0.676	0.684	0.817	0.868
enron	0.693	0.695	**0.699**	0.604	0.604	0.629	0.546	0.635	0.522	0.576	0.680	0.698
corel5k	0.303	0.293	**0.352**	0.311	0.222	0.196	0.208	0.266	0.088	0.014	0.314	0.334
tmc2007	0.978	0.981	0.972	0.938	0.945	0.842	0.700	0.844	0.939	0.935	0.945	**0.996**
mediamill	0.686	0.672	0.450	0.492	0.583	0.669	0.654	0.703	0.492	0.453	0.728	**0.737**
bibtex	0.597	**0.599**	0.579	0.498	0.407	0.392	0.212	0.349	DNF	0.228	0.418	0.525
delicious	0.351	0.343	DNF	DNF	0.231	0.321	0.206	0.326	DNF	DNF	0.359	**0.395**
bookmarks	DNF	DNF	DNF	DNF	DNF	0.378	0.213	0.381	DNF	DNF	0.423	**0.480**

Table B19
The *p*-values of the assessment of performance of the multi-label learning approaches by the Friedman test using the ranking-based evaluation measures. *Subset* shows the calculated p-values for the datasets on which all algorithms finished. *All* shows the calculated *p*-values for all datasets including those which did not finished.

Evaluation measure	All	Subset
One error	2.2×10^{-7}	5.3×10^{-6}
Coverage	1×10^{-18}	2.3×10^{-16}
Ranking loss	1×10^{-18}	1.2×10^{-16}
Average precision	6.5×10^{-14}	2×10^{-11}

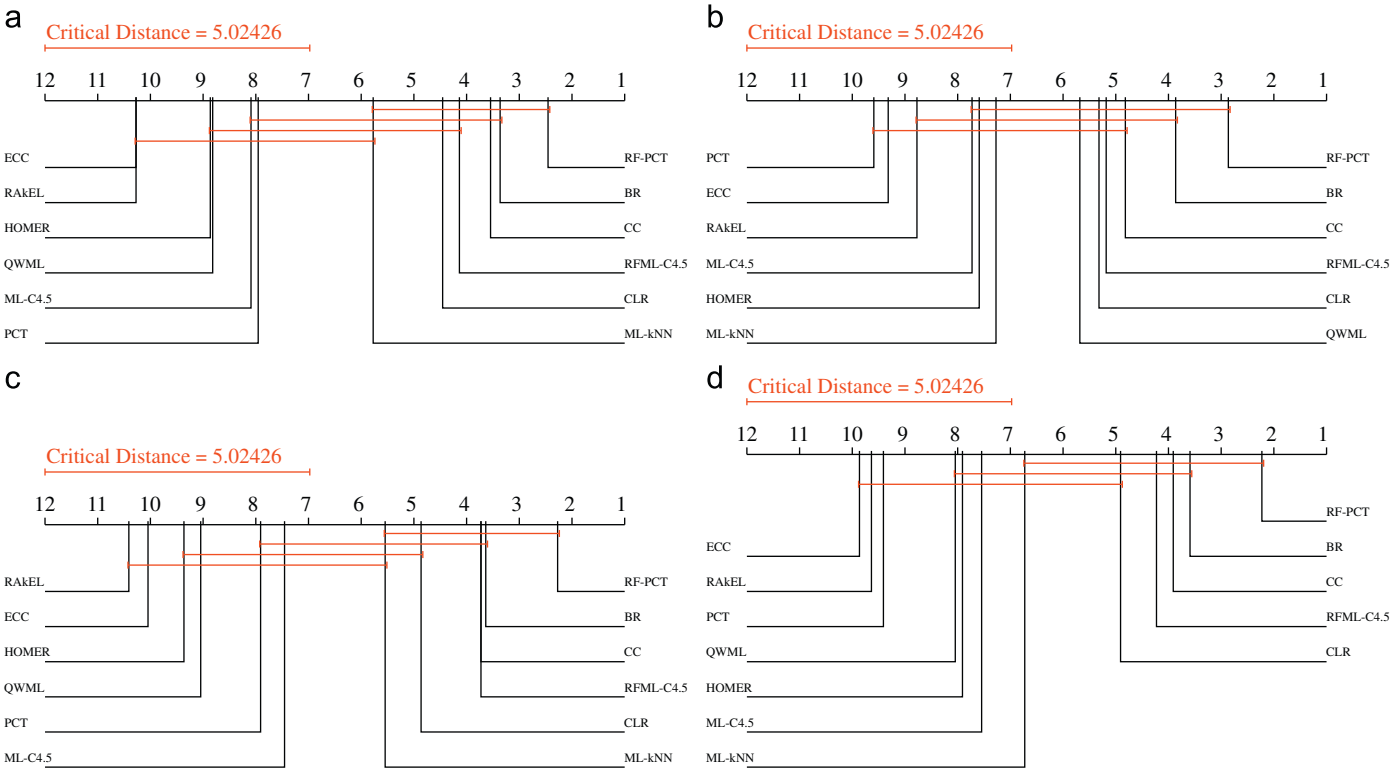

Fig. B3. The critical diagrams for the ranking-based evaluation measures: the results from the Nemenyi post-hoc test at 0.05 significance level on all the datasets: (a) *ranking loss*; (b) *one-error*; (c) *coverage*; and (d) *average precision*.

Table B20
The performance of the multi-label learning approaches in terms of the *training time* measures in seconds. DNF (*Did Not Finish*) stands for the algorithms that did not construct a predictive model within one week under the available resources.

Dataset	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-*k*NN	RAkEL	ECC	RFML-C4.5	RF-PCT
emotions	4.0	6.0	10.0	10.0	4.0	0.3	**0.1**	0.4	5.0	4.9	1.2	2.9
scene	71.0	99.0	195.0	195.0	68.0	8.0	**2.0**	14.0	79.0	319.0	10.0	23.0
yeast	145.0	206.0	672.0	672.0	101.0	14.0	**1.5**	8.2	157.0	497.0	19.0	25.0
medical	18.0	28.0	40.0	40.0	16.0	3.0	**0.6**	1.0	82.0	103.0	7.0	27.0
enron	318.0	440.0	971.0	971.0	158.0	15.0	**1.1**	6.0	493.0	1467.0	25.0	47.0
corel5k	926.0	1225.0	2388.0	2388.0	771.0	369.0	**30.0**	389.0	3380.0	20 073.0	385.0	902.0
tmc2007	42 645.0	46 704.0	52 427.0	52 427.0	31 300.0	469.0	**11.5**	737.0	102 394.0	92 169.0	460.0	557.0
mediamill	85 468.0	10 0435.0	260 156.0	260 156.0	78 195.0	2030.0	**440.0**	1094.0	33 554.0	188 957.0	4056.0	8360.0
bibtex	11 013.0	12 434.0	13 424.0	13 424.0	2896.0	566.0	**16.4**	124.0	DNF	29 578.0	645.0	1550.0
delicious	57 053.0	84 903.0	DNF	DNF	21218.0	2738.0	**70.0**	236.0	DNF	DNF	21 776.0	5376.0
bookmarks	DNF	DNF	DNF	DNF	DNF	4039.0	**965.0**	15 990.0	DNF	DNF	5602.0	28 900.0

Table B21
The performance of the multi-label learning approaches in terms of the *testing time* measures in seconds. DNF (*Did Not Finish*) stands for the algorithms that did not construct a predictive model within one week under the available resources).

Dataset	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-*k*NN	RAkEL	ECC	RFML-C4.5	RF-PCT
emotions	1.0	1.0	3.0	2.0	1.0	**0.0**	**0.0**	0.4	2.0	6.6	0.1	0.3
scene	25.0	25.0	87.0	40.0	21.0	1.0	**0.0**	14.0	72.0	168.0	2.0	1.0
yeast	23.0	25.0	153.0	64.0	17.0	0.1	**0.0**	5.0	70.0	158.0	0.5	0.2
medical	4.0	6.0	90.0	25.0	1.5	0.1	**0.0**	0.2	24.0	46.0	0.5	0.5
enron	50.0	53.0	634.0	174.0	22.0	0.2	**0.0**	3.0	153.0	696.0	1.0	1.0
corel5k	25.0	31.0	2161.0	119.0	14.0	**1.0**	**1.0**	45.0	3613.0	2077.0	1.8	2.5
tmc2007	927.0	891.0	3282.0	1543.0	730.0	1.7	**0.0**	230.0	10 985.0	10 865.0	3.4	2.8
mediamill	6152.0	6125.0	76 385.0	20 317.0	6079.0	**1.0**	**1.0**	477.0	39 001.0	50 183.0	8.0	4.0
bibtex	654.0	661.0	16 733.0	4710.0	155.0	6.5	**0.0**	64.0	DNF	10 756.0	12.0	18.0
delicious	2045.0	1872.0	DNF	DNF	816.0	19.0	**10.0**	55.0	DNF	DNF	32.0	48.0
bookmarks	DNF	DNF	DNF	DNF	DNF	21.0	**15.0**	4084.0	DNF	DNF	28.0	58.0

Table B22
The *p*-values of the assessment of performance of the multi-label learning approaches by the Friedman test using the efficiency measures. *Subset* shows the calculated *p*-values for the datasets on which all algorithms finished. *All* shows the calculated *p*-values for all datasets including those which did not finished.

Evaluation measure	All	Subset
Training time	1×10^{-18}	1×10^{-18}
Testing time	1×10^{-18}	1×10^{-18}

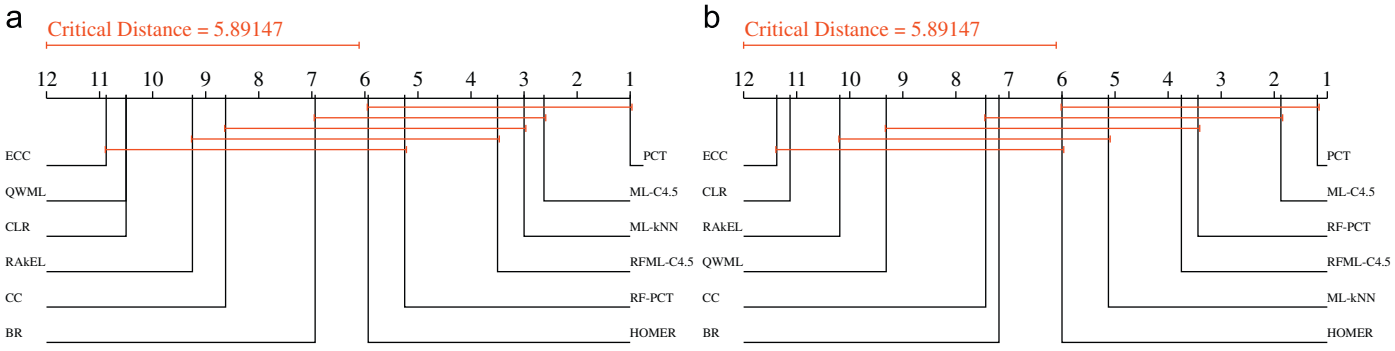

Fig. B4. The critical diagrams for the efficiency measures: the results from the Nemenyi post-hoc test at 0.05 significance level on all the datasets: (a) *training time* and (b) *testing time*.

References

[1] K. Brinker, J. Fürnkranz, E. Hüllermeier, A unified model for multilabel classification and ranking, in: Proceedings of the 17th European Conference on Artificial Intelligence, 2006, pp. 489–493.

[2] G. Tsoumakas, I. Katakis, Multi label classification: an overview, International Journal of Data Warehouse and Mining 3 (3) (2007) 1–13.

[3] M.L. Zhang, Z.H. Zhou, ML-kNN: a lazy learning approach to multi-label learning, Pattern Recognition 40 (7) (2007) 2038–2048.

[4] A. Wieczorkowska, P. Synak, Z. Ras, Multi-label classification of emotions in music, in: Intelligent Information Processing and Web Mining, Springer, Berlin/Heidelberg, 2006, pp. 307–315.

[5] E. Spyromitros, G. Tsoumakas, I. Vlahavas, An empirical study of lazy multilabel classification algorithms, in: Proceedings of the 5th Hellenic conference on Artificial Intelligence: Theories, Models and Applications, 2008, pp. 401–406.

[6] K. Crammer, Y. Singer, A family of additive online algorithms for category ranking, Journal of Machine Learning Research 3 (2003) 1025–1058.

[7] M.L. Zhang, Z.H. Zhou, Multi-label neural networks with applications to functional genomics and text categorization, IEEE Transactions on Knowledge and Data Engineering 18 (10) (2006) 1338–1351.
[8] R.E. Schapire, Y. Singer, Boostexter: a boosting-based system for text categorization, Machine Learning 39 (2000) 135–168.
[9] F. De Comité, R. Gilleron, M. Tommasi, 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, 2003, pp. 35–49.
[10] F.A. Thabtah, P. Cowling, Y. Peng, MMAC: a new multi-class, multi-label associative classification approach, in: Proceedings of the 4th IEEE International Conference on Data Mining, 2004, pp. 217–224.
[11] A. Clare, R.D. King, Knowledge discovery in multi-label phenotype data, in: Proceedings of the 5th European Conference on PKDD, 2001, pp. 42–53.
[12] H. Blockeel, L.D. Raedt, J. Ramon, Top-down induction of clustering trees, in: Proceedings of the 15th International Conference on Machine Learning, 1998, pp. 55–63.
[13] J. Fürnkranz, Round robin classification, Journal of Machine Learning Research 2 (2002) 721–747.
[14] T.-F. Wu, C.-J. Lin, R.C. Weng, Probability estimates for multi-class classification by pairwise coupling, Journal of Machine Learning Research 5 (2004) 975–1005.
[15] G. Tsoumakas, I. Vlahavas, Random k-labelsets: an ensemble method for multilabel classification, in: Proceedings of the 18th European conference on Machine Learning, 2007, pp. 406–417.
[16] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, in: Proceedings of the 20th European Conference on Machine Learning, 2009, pp. 254–269.
[17] D. Kocev, C. Vens, J. Struyf, S. Džeroski, Ensembles of multi-objective decision trees, in: Proceedings of the 18th European conference on Machine Learning, 2007, pp. 624–631.
[18] D. Kocev, Ensembles for predicting structured outputs, Ph.D. thesis, IPS Jožef Stefan, Ljubljana, Slovenia, 2011.
[19] W. Cheng, E. Hullermeier, Combining instance-based learning and logistic regression for multilabel classification, Machine Learning 76 (2009) 211–225.
[20] A. Elisseeff, J. Weston, A Kernel method for multi-labelled classification, in: Proceedings of the Annual ACM Conference on Research and Development in Information Retrieval, 2005, pp. 274–281.
[21] S. Godbole, S. Sarawagi, Discriminative methods for multi-labeled classification, in: Advances in Knowledge Discovery and Data Mining, Springer, Berlin/Heidelberg, 2004, pp. 22–30.
[22] J. Read, B. Pfahringer, G. Holmes, Multi-label classification using ensembles of pruned sets, in: Proceedings of the 8th IEEE International Conference on Data Mining, 2008, pp. 995–1000.
[23] J. Read, A pruned problem transformation method for multi-label classification, in: Proceedings of the New Zealand Computer Science Research Student Conference, 2008, pp. 143–150.
[24] G. Tsoumakas, I. Katakis, I. Vlahavas, Effective and efficient multilabel classification in domains with large number of labels, in: Proceedings of the ECML/PKDD Workshop on Mining Multidimensional Data, 2008, pp. 30–44.
[25] S.-H. Park, J. Fürnkranz, Efficient pairwise classification, in: Proceedings of the 18th European Conference on Machine Learning, 2007, pp. 658–665.
[26] E.L. Mencía, S.-H. Park, J. Fürnkranz, Efficient voting prediction for pairwise multilabel classification, Neurocomputing 73 (2010) 1164–1176.
[27] J. Read, B. Pfahringer, G. Holmes, Multi-label classification using ensembles of pruned sets, in: Proceedings of the 8th IEEE International Conference on Data Mining, 2008, pp. 995–1000.
[28] L. Breiman, J. Friedman, R. Olshen, C.J. Stone, Classification and Regression Trees, Chapman & Hall/CRC, 1984.
[29] L. Breiman, Random forests, Machine Learning 45 (2001) 5–32.
[30] G. Tsoumakas, I. Katakis, I. Vlahavas, Mining multi-label data, in: Data Mining and Knowledge Discovery Handbook, Springer, Berlin/Heidelberg, 2010, pp. 667–685.
[31] K. Trohidis, G. Tsoumakas, G. Kalliris, I. Vlahavas, Multilabel classification of music into emotions, in: Proceedings of the 9th International Conference on Music Information Retrieval, 2008, pp. 320–330.
[32] M.R. Boutell, J. Luo, X. Shen, C.M. Brown, Learning multi-label scene classification, Pattern Recognition 37 (9) (2004) 1757–1771.
[33] B. Klimt, Y. Yang, The enron corpus: a new dataset for email classification research, in: Proceedings of the 15th European conference on Machine Learning, 2004, pp. 217–226.
[34] P. Duygulu, K. Barnard, J. de Freitas, D. Forsyth, Object recognition as machine translation: learning a lexicon for a fixed image vocabulary, in: Proceedings of the 7th European Conference on Computer Vision, 2002, pp. 349–354.
[35] A. Srivastava, B. Zane-Ulman, Discovering recurring anomalies in text reports regarding complex space systems, in: Proceedings of the IEEE Aerospace Conference, 2005, pp. 55–63.
[36] C.G.M. Snoek, M. Worring, J.C. van Gemert, J.-M. Geusebroek, A.W.M. Smeulders, The challenge problem for automated detection of 101 semantic concepts in multimedia, in: Proceedings of the 14th Annual ACM International Conference on Multimedia, 2006, pp. 421–430.
[37] I. Katakis, G. Tsoumakas, I. Vlahavas, Multilabel text classification for automated tag suggestion, in: Proceedings of the ECML/PKDD Discovery Challenge, 2008.
[38] CDC/National Center for Health Statistics, International Classification of Diseases, Ninth Revision, Clinical Modification (ICD-9-CM), 〈http://www.cdc.gov/nchs/icd/icd9cm.htm〉 (2011).
[39] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, SIGKDD Explorations 11 (2009) 10–18.
[40] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, software available at 〈http://www.csie.ntu.edu.tw/cjlin/libsvm〉 (2001).
[41] E. Bauer, R. Kohavi, An empirical comparison of voting classification algorithms: bagging, boosting, and variants, Machine Learning 36 (1) (1999) 105–139.
[42] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, Annals of Mathematical Statistics 11 (1940) 86–92.
[43] P.B. Nemenyi, Distribution-free multiple comparisons, Ph.D. thesis, Princeton University, 1963.
[44] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.
[45] R.L. Iman, J.M. Davenport, Approximations of the critical region of the Friedman statistic, Communications in Statistics (1980) 571–595.
[46] E.S. Pearson, H.O. Hartley, Biometrika Tables for Statisticians, vol. I, Cambridge University Press, 1966.
[47] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, H. Blockeel, Decision trees for hierarchical multi-label classification, Machine Learning 73 (2) (2008) 185–214.

Gjorgji Madjarov received his bachelor and master degrees in computer science, automation and electrical engineering in 2007 and 2009, respectively, from the Faculty of Electrical Engineering and Information Technology, University "Ss. Cyril and Methodius" in Skopje, Republic of Macedonia. Now he is working on his Ph.D. thesis in the area of multi-label and hierarchical classification. At present he is a teaching and research assistant at the Faculty of Computer Science and Engineering in Skopje and a visiting researcher at the Jozef Stefan Institute, Slovenia. His fields of interest include artificial intelligence, supervised learning, unsupervised learning, computer vision and pattern recognition.

Dragi Kocev received his Ph.D. degree in Computer Science from the IPS Jozef Stefan in 2011. He is currently a post-doctoral researcher at the Department of Knowledge Technologies, Jozef Stefan Institute, Ljubljana, Slovenia. His research interests include machine learning and data mining, prediction of structured outputs and their applications in environmental and life sciences.

Dejan Gjorgjevikj received his B.Sc., in electrical engineering, and his M.Sc. and Ph.D. in computer science and engineering from the Faculty of Electrical Engineering, University "Ss. Cyril and Methodius" – Skopje, in 1992, 1997 and 2004, respectively. He is currently a professor at the Faculty of Computer Science and Engineering, University "Ss. Cyril and Methodius" in Skopje, Macedonia. His research interests include artificial intelligence, machine learning, computer vision, pattern recognition and software engineering. He is a member of IEEE and ACM.

Sašo Džeroski received his Ph.D. degree in Computer Science from the University of Ljubljana in 1995. He is currently a scientific councilor at the Department of Knowledge Technologies, Jozef Stefan Institute, and the Centre of Excellence for Integrated Approaches in Chemistry and Biology of Proteins, both in Ljubljana, Slovenia. He is also an associate professor at the Jozef Stefan International Postgraduate School, also in Ljubljana. His research interests include data mining and machine learning and their applications in environmental sciences (ecology) and life sciences (biomedicine). He is an ECCAI fellow, member of the executive board of SLAIS, member of ACM SIGKDD and IEEE.